On the usage of the Gram matrix for multivariate functional principal components analysis

Steven Golovkine[*]   Edward Gunning[†]   Andrew J. Simpkin[‡]

Norma Bargary[§]

May 19, 2023

Abstract

Dimension reduction is crucial in functional data analysis (FDA). The key tool to reduce the dimension of the data is functional principal component analysis. Existing approaches for functional principal component analysis usually involve the diagonalisation of the covariance operator. With the increasing size and complexity of functional datasets, estimating the covariance operator becomes more challenging. Therefore, there is a growing need for efficient methodologies to estimate the eigencomponents. Using the duality of the space of observations and the space of functional components, we propose to use the inner-product between the curves to estimate the eigenelements of multivariate and multidimensional functional datasets. The relationship between the eigenelements of the covariance operator and those of the inner-product matrix is established. We explore the application of these methodologies in several FDA settings and provide general guidance on their usability.

Keywords— Dimension Reduction; Functional Data Analysis; Functional Principal Components; Multivariate Functional Data

1   Introduction

Functional data analysis (FDA) is a statistical methodology for analyzing data that can be represented as functions. These functions could represent measurements taken over time or space, such as temperature readings over a yearly period or spatial patterns of disease occurrence. The goal of FDA is to extract meaningful information from these functions and to model their behavior. See, e.g., Ramsay and Silverman (2005); Horvàth and Kokoszka (2012); Wang et al. (2016); Kokoszka et al. (2017) for some references on functional data analysis.

Functional principal component analysis (FPCA) is an extension of principal component analysis (PCA), which is a commonly used tool for dimension reduction in multivariate data, to functional data. PCA was introduced by Karhunen (1947) and Loève (1945) and developed by Dauxois et al. (1982). Since then, FPCA has become a prevalent tool in FDA due to its ability to convert infinite-dimensional functional data into a finite-dimensional vector of random scores. These scores are a countable sequence of uncorrelated random variables that can be truncated to a finite vector in practical applications. By applying multivariate data analysis tools to these random scores, FPCA can achieve the goal of dimension reduction while assuming mild assumptions about the underlying stochastic process. FPCA is usually used as a preprocessing step to feed, e.g., regression and classification models. Recently, FPCA has been extended to multivariate functional data, which are data that consist of multiple functions that are observed simultaneously. This extension is referred to as multivariate functional principal component analysis (MFPCA). As for FPCA, a key benefit of MFPCA is that it allows one to identify and visualize the main sources of variation in the multivariate functional data. This can be useful in many different applications, such as identifying patterns of movements in sport biomechanics (Warmenhoven et al., 2019), analyzing changes in brain activity in neuroscience (Song and Kim, 2022), or comparing countries' competitiveness in economics (Krzyśko et al., 2022).

In MFPCA, we seek to decompose the covariance structure of the multivariate functional data into a set of orthogonal basis functions, named the principal components, which capture the main

---

[*]MACSI, Department of Mathematics and Statistics, University of Limerick, Ireland steven.golovkine@ul.ie

[†]MACSI, Department of Mathematics and Statistics, University of Limerick, Ireland edward.gunning@ul.ie

[‡]School of Mathematical and Statistical Sciences, University of Galway, Ireland andrew.simpkin@nuigalway.ie

[§]MACSI, Department of Mathematics and Statistics, University of Limerick, Ireland norma.bargary@ul.ie

sources of variation in the data. There are multiple approaches to estimate the principal components of a multivariate functional dataset. Ramsay and Silverman (2005) combine the multivariate curves into one big curve and then perform a usual FPCA via an eigendecomposition of the covariance structure. This methodology can only be run for data that are defined on the same unidimensional domain, that exhibit similar amounts of variability and measured in the same units. Jacques and Preda (2014) propose to expand each univariate component into a basis of functions separately. This results in a different set of coefficients for each univariate curve. The eigendecomposition is then run on the matrix of the stacked coefficients. To consider the normalization issue of Ramsay and Silverman (2005), Jacques and Preda (2014) and Chiou et al. (2014) propose to normalize the data by the standard deviation of the curves at each of the sampling points. Happ and Greven (2018) extend the estimation of principal components to functional data defined on different dimensional domains. Their estimation procedure is based on carrying out FPCA on each univariate component, and then using a weighted combination of the resulting principal components to obtain the multivariate eigencomponents. Finally, Berrendero et al. (2011) develop a different method to estimate the eigencomponents as they perform a principal components analysis for each sampling time point.

The key motivation of this paper is to investigate the duality between rows and columns of a data matrix to estimate the eigencomponents of a multivariate functional dataset. The duality between rows and columns of a data matrix is a fundamental concept in classical statistics (Escofier, 1979; Saporta, 1990). A data matrix typically represents a set of observations of multiple features, each row corresponds to an individual observation and each column correspond to an individual feature. The duality between rows and columns refers to the fact that many statistical methodologies can be conducted either on the rows or the columns of the data matrix, and the results will be related to each other. For example, the principal components obtained from a PCA run on the rows of the data matrix can be used to compute PCA on the columns of the matrix. The choice of method to use, based on criteria such as computational time or data storage needs, is thus left to the statistician. This concept has been widely studied for multivariate statistics (see, e.g., Pagès (2014); Härdle and Simar (2019)). In the context of functional data, this principle has received limited attention despite being mentioned in the seminal paper of FDA (Ramsay, 1982). Ramsay and Silverman (2005) briefly commented on it in a concluding remark of Chapter 8, while Kneip and Utikal (2001) and Benko et al. (2009) utilized it to compute principal components for dense univariate functional data. Chen et al. (2017) also employ it to gain computational advantage when univariate functional data are sampled on a very dense grid. To the best of our knowledge, however, there is no available literature on its application to multivariate functional data that are observed on different dimensional domains. Our aim is therefore to investigate this duality for multivariate functional data observed on different dimensional domains and provide guidelines to statisticians on which method to use in different cases.

The remainder of the paper is organized as follows. In Section 2, we define multivariate functional data, with components that are observed on possibly different domains. In Section 3, we develop the duality between the observations' space and the functional components' space. The relationship between the eigencomponents of the covariance operator of the multivariate functional datasets and the eigencomponents of the inner-product matrix between the observations is derived in Section 4. Extensive simulations are given in Section 5. We also provide guidelines on which method to use with respect to different data characteristics. The paper concludes with a discussion and an outlook in Section 6.

2 Model

The structure of the data we consider, referred to as *multivariate functional data*, is similar to that presented in Happ and Greven (2018). The data consist of independent trajectories of a vector-valued stochastic process $X = (X^{(1)}, \ldots, X^{(P)})^\top$, $P \geq 1$. (Here and in the following, for any matrix A, $A^\top$ denotes its transpose.) For each $1 \leq p \leq P$, let $\mathcal{T}_p$ be a rectangle in some Euclidean space $\mathbb{R}^{d_p}$ with $d_p \geq 1$, e.g., $\mathcal{T}_p = [0,1]^{d_p}$. Each coordinate $X^{(p)} : \mathcal{T}_p \to \mathbb{R}$ is assumed to belong to $\mathcal{L}^2(\mathcal{T}_p)$, the Hilbert space of square-integrable real-valued functions defined on $\mathcal{T}_p$, having the usual inner product that we denote by $\langle \cdot, \cdot \rangle$, and $\|\cdot\|$ the associated norm. Thus X is a stochastic process indexed by $\mathbf{t} = (t_1, \ldots, t_P)$ belonging to the $P-$fold Cartesian product $\mathcal{T} := \mathcal{T}_1 \times \cdots \times \mathcal{T}_P$ and taking values in the $P-$fold Cartesian product space $\mathcal{H} := \mathcal{L}^2(\mathcal{T}_1) \times \cdots \times \mathcal{L}^2(\mathcal{T}_P)$.

Figure 1: Functional data matrix, adapted from Berrendero et al. (2011).

We consider the function $\langle\!\langle \cdot, \cdot \rangle\!\rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$,

$$\langle\!\langle f, g \rangle\!\rangle := \sum_{p=1}^{P} \langle f^{(p)}, g^{(p)} \rangle = \sum_{p=1}^{P} \int_{\mathcal{T}_p} f^{(p)}(t_p) g^{(p)}(t_p) \mathrm{d}t_p, \quad f, g \in \mathcal{H}.$$

$\mathcal{H}$ is a Hilbert space with respect to the inner product $\langle\!\langle \cdot, \cdot \rangle\!\rangle$ (Happ and Greven, 2018). We denote by $\|\cdot\|$, the norm induced by $\langle\!\langle \cdot, \cdot \rangle\!\rangle$. Let $\mu : \mathcal{T} \to \mathcal{H}$ denote the mean function of the process X, $\mu(\mathbf{t}) := \mathbb{E}(X(\mathbf{t})), \mathbf{t} \in \mathcal{T}$. Let C denote the $P \times P$ matrix-valued covariance function which, for $\mathbf{s}, \mathbf{t} \in \mathcal{T}$, is defined as

$$C(\mathbf{s}, \mathbf{t}) := \mathbb{E}\left(\{X(\mathbf{s}) - \mu(\mathbf{s})\}\{X(\mathbf{t}) - \mu(\mathbf{t})\}^{\top}\right), \quad \mathbf{s}, \mathbf{t} \in \mathcal{T}.$$

More precisely, for $1 \leq p, q \leq P$, the (p, q)th entry of the matrix $C(\mathbf{s}, \mathbf{t})$ is the covariance function between the pth and the qth features of the process X:

$$C_{p,q}(s_p, t_q) := \mathbb{E}\left(\{X^{(p)}(s_p) - \mu^{(p)}(s_p)\}\{X^{(q)}(t_q) - \mu^{(q)}(t_q)\}\right), \quad s_p \in \mathcal{T}_p, t_q \in \mathcal{T}_q.$$

Let $\Gamma : \mathcal{H} \to \mathcal{H}$ denote the covariance operator of X, defined as an integral operator with kernel C. That is, for $f \in \mathcal{H}$ and $\mathbf{t} \in \mathcal{T}$, the pth feature of $\Gamma f(\mathbf{t})$ is given by

$$(\Gamma f)^{(p)}(t_p) := \langle\!\langle C_{p,\cdot}(t_p, \cdot), f(\cdot) \rangle\!\rangle = \langle\!\langle C_{\cdot,p}(\cdot, t_p), f(\cdot) \rangle\!\rangle, \quad t_p \in \mathcal{T}_p.$$

Let us consider a set of N curves $\mathcal{X} = \{X_1, \ldots, X_n, \ldots, X_N\}$ generated as a random sample of the P-dimensional stochastic process X with continuous trajectories. Unless otherwise stated, the data are assumed to be observed without error. The data can be viewed as a table with N rows and P columns where each entry is a curve, potentially on a multidimensional domain (see Figure 1). Each row of this matrix represents an observation; while each column represents a functional feature. At the intersection of row n and column p, we thus have $X_n^{(p)}$ which is the curve that concerns the (functional) feature p for the individual n.

For $n \in \{1, \ldots, N\}$, each observation n is attributed the weight π_n such that $\sum_n \pi_n = 1$, e.g., $\pi_n = 1/N$. For a given $p \in \{1, \ldots, P\}$, the mean curve of the pth feature along the N observations is denoted by $\mu^{(p)}$. This quantity can be computed using

$$\mu^{(p)}(t_p) = \sum_{n=1}^{N} \pi_n X_n^{(p)}(t_p), \quad t_p \in \mathcal{T}_p, \quad p \in \{1, \ldots, P\}.$$

The covariance function of the pth feature along the N observations can be computed using

$$C_{p,p}(s_p, t_p) = \sum_{n=1}^{N} \pi_n X_n^{(p)}(s_p) X_n^{(p)}(t_p) - \mu^{(p)}(s_p) \mu^{(p)}(t_p), \quad s_p, t_p \in \mathcal{T}_p, \quad p \in \{1, \ldots, P\}. \quad (1)$$

For the set $\mathcal{X}$, the inner-product matrix, also called the Gram matrix, M is defined as a matrix of size $N \times N$ with entries

$$M_{nn'} = \langle\!\langle X_n - \mu, X_{n'} - \mu \rangle\!\rangle, \quad n, n' = 1, \ldots, N. \quad (2)$$

2.1 Basis decomposition

In many practical situations, data are noisy and only observed at specific time points. To extract the underlying functional features of the data, smoothing and interpolation techniques are commonly employed. These techniques involve approximating the true underlying function generating the data by a finite-dimensional set of basis functions. Assume that for each feature $p = 1, \ldots, P$, there exists a set of basis of functions $\Psi^{(p)} = \{\psi_k^{(p)}\}_{1 \leq k \leq K_p}$ such that each feature of each curve $n = 1, \ldots, N$ can be expanded into using basis:

$$X_n^{(p)}(t_p) = \sum_{k=1}^{K_p} c_{nk}^{(p)} \psi_k^{(p)}(t_p), \quad t_p \in \mathcal{T}_p, \tag{3}$$

where $\{c_{nk}^{(p)}\}_{1 \leq k \leq K_p}$ is a set of coefficients the feature p of curve n. We denote by $\bar{c}_k^{(p)} = \sum_{n=1}^N \pi_n c_{nk}^{(p)}$ the mean coefficient of feature p corresponding to the kth basis function. The pth feature of the mean function can be then expanded in the same basis as:

$$\mu^{(p)}(t_p) = \sum_{k=1}^{K_p} \bar{c}_k^{(p)} \psi_k^{(p)}(t_p), \quad t_p \in \mathcal{T}_p.$$

In a similar way, the covariance function of the pth feature is given by:

$$C_{p,p}(s_p, t_p) = \sum_{k=1}^{K_p} \sum_{l=1}^{K_p} \left(\sum_{n=1}^N \pi_n c_{nk}^{(p)} c_{nl}^{(p)} - \bar{c}_k^{(p)} \bar{c}_l^{(p)} \right) \psi_k^{(p)}(s_p) \psi_l^{(p)}(t_p), \quad s_p, t_p \in \mathcal{T}_p.$$

These formula can be written in matrix form as follows. For $\mathbf{t} \in \mathcal{T}$, we have that $X(\mathbf{t}) = \mathbf{C}\Psi(\mathbf{t})$ where $X(\mathbf{t})$ is a matrix with N rows and P columns with entries $X_n^{(p)}(t_p)$, $t_p \in \mathcal{T}_p$, $1 \leq p \leq P$, $1 \leq n \leq N$,

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}^{(1)} & \cdots & \mathbf{C}^{(P)} \end{pmatrix}, \quad \text{and} \quad \Psi(\mathbf{t}) = \text{diag}\{\Psi^{(1)}(t_1), \ldots, \Psi^{(P)}(t_P)\},$$

where

$$\mathbf{C}^{(p)} = \begin{pmatrix} c_{11}^{(p)} & \cdots & c_{1K_p}^{(p)} \\ \vdots & \ddots & \vdots \\ c_{N1}^{(p)} & \cdots & c_{NK_p}^{(p)} \end{pmatrix} \quad \text{and} \quad \Psi^{(p)}(t_p) = \begin{pmatrix} \psi_1^{(p)}(t_p) \\ \vdots \\ \psi_{K_p}^{(p)}(t_p) \end{pmatrix}.$$

Using the basis expansion and denoting $\Pi^\top = (\pi_1, \ldots, \pi_N)$, the mean and covariance functions are given by

$$\mu(\mathbf{t}) = \Psi(\mathbf{t})^\top \mathbf{C}^\top \Pi \quad \text{and} \quad C(\mathbf{s}, \mathbf{t}) = \Psi(\mathbf{s})^\top \mathbf{C}^\top \left(\text{diag}\{\pi_1, \ldots, \pi_N\} - \Pi\Pi^\top \right) \mathbf{C}\Psi(\mathbf{t}).$$

Finally, we denote by $\mathbf{W}$ the matrix of inner products of the functions in the basis Ψ. The matrix $\mathbf{W}$ is a block-diagonal matrix such that $\mathbf{W} = \text{diag}\{\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(P)}\}$ where each entry is given by

$$\mathbf{W}_{k,l}^{(p)} = \langle \psi_k^{(p)}, \psi_l^{(p)} \rangle, \quad 1 \leq k, l \leq K_p, \quad 1 \leq p \leq P.$$

We remark that, if the basis Ψ is an orthonormal basis, the matrix $\mathbf{W}$ is equal to the identity matrix of size $\sum_{p=1}^P K_p$. Using the expansion of the data into the basis of function Ψ, the inner-product matrix $\mathbf{M}$ is written

$$\mathbf{M} = \left(\mathbf{I}_N - \mathbf{1}_N \Pi^\top \right) \mathbf{C}\mathbf{W}\mathbf{C}^\top \left(\mathbf{I}_N - \Pi\mathbf{1}_N^\top \right)$$

$$= \left[\left(\mathbf{I}_N - \mathbf{1}_N \Pi^\top \right) \mathbf{C}\mathbf{W}^{1/2} \right] \left[\left(\mathbf{I}_N - \mathbf{1}_N \Pi^\top \right) \mathbf{C}\mathbf{W}^{1/2} \right]^\top,$$

where $\mathbf{I}_N$ is the identity matrix of size N and $\mathbf{1}_N$ is the vector of 1 of length N.

3 On the geometry of multivariate functional data

3.1 Duality diagram

The distinction between the space of rows of a matrix as a sample from a population and the space of columns as the fixed variables on which the observations were measured has been explained in Holmes (2008); De la Cruz and Holmes (2011) for multivariate data analysis. We propose to define
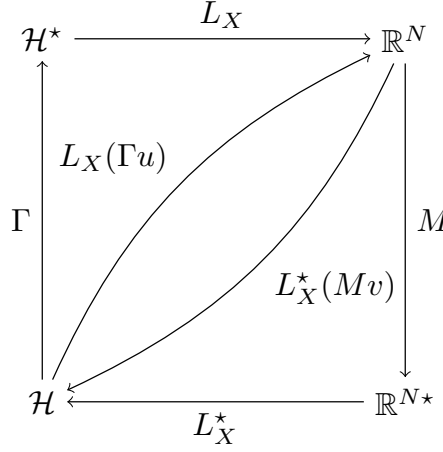
Figure 2: Duality diagram between the spaces $\mathcal{H}$ and $\mathbb{R}^N$. The operator L_X and its adjoint $L_X^\star$ are linear operators. The covariance operator Γ and the matrix M define geometries in $\mathcal{H}$ and $\mathbb{R}^N$ respectively. The space $\mathcal{H}^\star$ (resp. $\mathbb{R}^{N\star}$) is the dual space of $\mathcal{H}$ (resp. $\mathbb{R}^N$).

a duality diagram in the context of multivariate functional data. Consider the data matrix defined by the set $\mathcal{X}$. We define an operator $L_X : \mathcal{H} \to \mathbb{R}^N$ by

$$
L_X : u \mapsto \begin{pmatrix} \langle\!\langle X_1 - \mu, u \rangle\!\rangle \\ \vdots \\ \langle\!\langle X_N - \mu, u \rangle\!\rangle \end{pmatrix}.
$$

Using the linearity of the inner-product $\langle\!\langle \cdot, \cdot \rangle\!\rangle$ and vectors, the operator L_X is linear. The adjoint operator of L_X is given by

$$
L_X^\star : v \mapsto \begin{pmatrix} \sum_{n=1}^N v_n \{X_n^{(1)}(t_1) - \mu^{(1)}(t_1)\} \\ \vdots \\ \sum_{n=1}^N v_n \{X_n^{(P)}(t_P) - \mu^{(P)}(t_P)\} \end{pmatrix}.
$$

As $L_X^\star$ is the adjoint of the linear operator L_X, it is a linear operator. We consider the function $\langle\!\langle f, g \rangle\!\rangle_\Gamma = \langle\!\langle f, \Gamma g \rangle\!\rangle$, $f, g \in \mathcal{H}$ and $\langle u, v \rangle_{\mathbf{M}} = u^\top \mathbf{M} v$, $u, v \in \mathbb{R}^N$. These relationship can be expressed as a duality diagram, see Figure 2. The triplet $(X, \Gamma, \mathbf{M})$ defines a (multivariate) functional data analysis setup. The estimation of the eigencomponents can be performed using Γ or $\mathbf{M}$ and the relationship between the two are derived in Section 4.

3.2 Cloud of individuals

Given $n \in \{1, \dots, N\}$, let $\{X_n^{(p)}(t_p), t_p \in \mathcal{T}_p, p = 1, \dots, P\}$ be the features set for a particular observation n. We identify this set as the point M_n in the space $\mathcal{H}$. The space $\mathcal{H}$ is referred to as the observations' space. The cloud of points that represents the set of observations in $\mathcal{H}$ is denoted by $\mathcal{C}_N$. Let G_N be the centre of gravity of the cloud $\mathcal{C}_N$. In the space $\mathcal{H}$, its coordinates are given by $\{\mu^{(p)}(t_p), t_p \in \mathcal{T}_p, p = 1, \dots, P\}$. If the features are centered, the origin $\mathrm{O}_\mathcal{H}$ of the axes in $\mathcal{H}$ coincides with G_N.

Let f and g be two elements in $\mathcal{H}$ and denote by M_f and M_g their associated points in $\mathcal{C}_N$ (see Figure 3). The distance between these observations is defined as

$$
d^2(\mathrm{M}_f, \mathrm{M}_g) = \|f - g\|^2 = \sum_{p=1}^P \int_{\mathcal{T}_p} \left\{ f^{(p)}(t_p) - g^{(p)}(t_p) \right\}^2 \mathrm{d}t_p.
$$

This distance measures how different the observations are, and thus characterizes the shape of the cloud $\mathcal{C}_N$. Another description of this shape is to consider the distance between each observation and G_N, the center of the cloud. Let f be an element of $\mathcal{H}$, associated to the point M_f, and μ the element of $\mathcal{H}$ related to G_N, the distance between f and μ is given by

$$
d^2(\mathrm{M}_f, \mathrm{G}_N) = \|f - \mu\|^2 = \sum_{p=1}^P \int_{\mathcal{T}_p} \left\{ f^{(p)}(t_p) - \mu^{(p)}(t_p) \right\}^2 \mathrm{d}t_p.
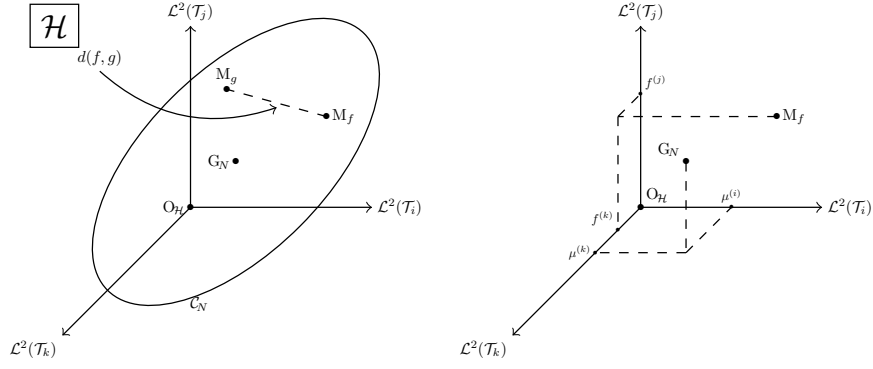$$

5

Figure 3: Left: Cloud of observations. Right: Projection of the points onto the elements of $\mathcal{H}$. The observation f (resp. g) is identified by the point M_f (resp. M_g) in the cloud $\mathcal{C}_N$. The point G_N is the center of gravity of $\mathcal{C}_N$ and the point $\mathrm{O}_{\mathcal{H}}$ is the origin of the space $\mathcal{H}$.

Given the set $\mathcal{X}$, the total inertia of $\mathcal{C}_N$, with respect to G_N, is given by

$$\sum_{n=1}^{N} \pi_n d^2(\mathrm{M}_n, \mathrm{G}_N) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \pi_i \pi_j d^2(\mathrm{M}_i, \mathrm{M}_j) = \sum_{p=1}^{P} \int_{\mathcal{T}_p} \mathrm{Var}\, X^{(p)}(t_p)\mathrm{d}t_p. \tag{4}$$

The derivation of these equalities are given in Appendix A.

Remark 1. *These results have the same interpretation as for multivariate scalar data. This is also the multivariate analogue of the relation between variance and sum of squared differences known for univariate functional data. If the features are reduced beforehand, the total inertia of the cloud $\mathcal{C}_N$ is equal to the number of components P. We are, in general, not interested by the total inertia but mostly how this variance is spread among the features.*

3.3 Cloud of features

Given an element $f \in \mathcal{H}$, let $\{\langle\!\langle X_n - \mu, f\rangle\!\rangle, n = 1, \ldots, N\}$ be the set of projections of f on the observations. We identify this set as the point M_f in the space $\mathbb{R}^N$. The space $\mathbb{R}^N$ is referred to as the features' space. The cloud of points that represents the set of observations in $\mathbb{R}^N$ is denoted by $\mathcal{C}_P$. The origin $\mathrm{O}_{\mathbb{R}}$ of the axes in $\mathbb{R}^N$ coincides with the centre of gravity of the cloud $\mathcal{C}_P$. We consider the usual inner-product in $\mathbb{R}^N$. Let f and g be two elements in $\mathcal{H}$ and denote by M_f and M_g their associated points in $\mathcal{C}_P$ (see Figure 4). The distance between M_f and M_g is thus defined as

$$d^2(\mathrm{M}_f, \mathrm{M}_g) = \sum_{n=1}^{N} \pi_n \langle\!\langle X_n - \mu, f - g\rangle\!\rangle.$$

Let f be an element of $\mathcal{H}$, associated to the point M_f, the distance between M_f and $\mathrm{O}_{\mathbb{R}}$ is given by

$$d^2(\mathrm{M}_f, \mathrm{O}_{\mathbb{R}}) = \sum_{n=1}^{N} \pi_n \langle\!\langle X_n - \mu, f\rangle\!\rangle.$$

$$\cos(\theta_{fg}) = \frac{\sum_{n=1}^{N} \pi_n \langle\!\langle X_n - \mu, f\rangle\!\rangle \langle\!\langle X_n - \mu, g\rangle\!\rangle}{\left(\sum_{n=1}^{N} \pi_n \langle\!\langle X_n - \mu, f\rangle\!\rangle^2\right)^{1/2} \left(\sum_{n=1}^{N} \pi_n \langle\!\langle X_n - \mu, g\rangle\!\rangle^2\right)^{1/2}}$$
$$= \frac{\langle\!\langle f, \Gamma g\rangle\!\rangle}{\langle\!\langle f, \Gamma f\rangle\!\rangle^{1/2} \langle\!\langle g, \Gamma g\rangle\!\rangle^{1/2}}$$
$$= \frac{\langle\!\langle \Gamma f, g\rangle\!\rangle}{\langle\!\langle f, \Gamma f\rangle\!\rangle^{1/2} \langle\!\langle g, \Gamma g\rangle\!\rangle^{1/2}}.$$

Remark 2. *Although each axis of the space does not directly represent the features, but rather the projection of an element of $\mathcal{H}$ onto the elements of the set $\mathcal{X}$, we refer to this space as the features' space. We use this terminology to highlight the similarity between multivariate functional data analysis and traditional multivariate data analysis, as well as to emphasize the dimensionality of this space.*
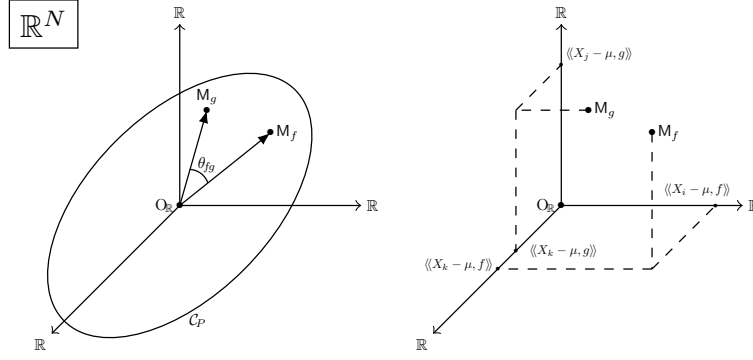
6

Figure 4: Left: Cloud of features. Right: Projection of the points on the elements of $\mathbb{R}^N$. The observation f (resp. g) is identified by the point M_f (resp. M_g) in the cloud $\mathcal{C}_P$. The point $\mathsf{O}_\mathbb{R}$ is the origin of the space $\mathbb{R}^N$ (and coincide with the center of gravity of $\mathcal{C}_P$).

3.4 On centering and reducing

For conducting an MFPCA, the features are usually assumed centred (Happ and Greven, 2018). Prothero et al. (2021) give a complete overview of centering in the context of FDA. Here, we comment on the geometric interpretation of centering in this context and compare with the multivariate scalar case. We focus on the usual centering in FDA, namely $X_n^{(p)}(t_p) - \mu^{(p)}(t_p)$, $t_p \in \mathcal{T}_p$ (refered as *object centering* in Prothero et al. (2021)). The geometric interpretation of the centering differs if we refer to the observations' space $\mathcal{H}$ or the features' space $\mathbb{R}^N$. Within the space $\mathcal{H}$, centering is interpreted as translating the centre of gravity of the curves G_N to the origin point $\mathsf{O}_\mathcal{H}$ of $\mathcal{H}$. This transformation, being a translation, does not change the shape of the cloud $\mathcal{C}_N$. The interpretation is the same as for the multivariate scalar data. Within the space $\mathbb{R}^N$, the centering is harder to interpret and does not have the same meaning as in the multivariate case. Actually, in the multivariate scalar case, the centering of the data can be geometrically interpreted as the projection of the data on the subspace orthogonal to the constant vector. However, what happens if we project the multivariate curves onto the vector (of length P) of constant functions?

In the features' space, the inner product is the usual Euclidean inner-product.

$$\langle\!\langle f, g \rangle\!\rangle = \sum_{i=1}^N \int_{\mathcal{T}_k} f_i^{(k)}(t) g_i^{(k)}(t) dt, \quad f, g \in \mathcal{G}.$$

Let $\mathbf{1}$ be the vector of constant function in $\mathcal{G}$ and f an element of $\mathcal{G}$. The projection of f onto $\mathbf{1}$ is then given by

$$P_\mathbf{1} f = \frac{\langle\!\langle f, \mathbf{1} \rangle\!\rangle}{\|\mathbf{1}\|} \mathbf{1} = \frac{1}{N|\mathcal{T}_k|} \sum_{n=1}^N \int_{\mathcal{T}_k} f_n^{(k)}(t) dt \, \mathbf{1}$$

In practice, this is equivalent to computing the mean value of the mean curve for each component, referred as *grand mean centering* in Prothero et al. (2021).

Concerning the standardization of the data, there are mainly two proposal in the literature. Happ and Greven (2018) propose to weight each component p by

$$w_p = \left(\int_{\mathcal{T}_p} \operatorname{Var} X^{(p)}(t_p) dt_p \right)^{-1}.$$

This standardization is coherent with the derivation of the total inertia of the observations' space. Using these weights, the total inertia of $\mathcal{C}_N$ is equal to the number of components P. Chiou et al. (2014) propose to standardize each component p of the data using the function

$$w_p(t_p) = \left(\operatorname{Var} X^{(p)}(t_p) \right)^{-1/2}, \quad t_p \in \mathcal{T}_p.$$

This corresponds to a standardization of the curves by the standard deviation of the component at each sampling point. The standard deviation curve is estimated as the square root of the diagonal of the covariance function estimates, obtained using a local linear smoother of the pooled data.

4 Multivariate functional principal components analysis

Assuming that the covariance operator Γ is a compact positive operator on $\mathcal{H}$ and using the results in Happ and Greven (2018), and the theory of Hilbert-Schmidt operators, e.g., Reed and Simon (1980), there exists a complete orthonormal basis $\Phi = \{\phi_k\}_{k\geq 1} \subset \mathcal{H}$ associated to a set of real numbers $\{\lambda_k\}_{k\geq 1}$ such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ that satisfy

$$\Gamma\phi_k = \lambda_k\phi_k, \quad \text{and} \quad \lambda_k \longrightarrow 0 \quad \text{as} \quad k \longrightarrow \infty. \tag{5}$$

The set $\{\lambda_k\}_{k\geq 1}$ contains the eigenvalues of the covariance operator Γ and Φ contains the associated eigenfunctions. Using the multivariate Karhunen-Loève theorem (Happ and Greven, 2018), we obtain the decomposition

$$X(\mathbf{t}) = \mu(\mathbf{t}) + \sum_{k=1}^{\infty} \mathfrak{c}_k\phi_k(\mathbf{t}), \quad \mathbf{t} \in \mathcal{T} \tag{6}$$

where $\mathfrak{c}_k = \langle\!\langle X - \mu, \phi_k \rangle\!\rangle$ are the projections of the centered curve onto the eigenfunctions. We have that $\mathbb{E}(\mathfrak{c}_k) = 0$, $\mathbb{E}(\mathfrak{c}_k^2) = \lambda_k$ and $\mathbb{E}(\mathfrak{c}_k\mathfrak{c}_{k'}) = 0$ for $k \neq k'$. Note that the coefficients $\mathfrak{c}_k$ are scalar random variables while the multivariate functions ϕ_k are vectors of functions. Let us call Φ the multivariate functional principal component analysis basis. In practice, we use a truncated version of the Karhunen-Loève expansion (6) as the eigenvalues λ_k, and hence the contribution of $\mathfrak{c}_k$ to (6), becomes negligible as k goes to infinity. Let

$$X_{\lceil K \rceil}(\mathbf{t}) = \mu(\mathbf{t}) + \sum_{k=1}^{K} \mathfrak{c}_k\phi_k(\mathbf{t}), \quad \mathbf{t} \in \mathcal{T}, \quad K \geq 1, \tag{7}$$

be the truncated Karhunen-Loève expansion of the process X and

$$X_{\lceil K_p \rceil}^{(p)}(t_p) = \mu^{(p)}(t_p) + \sum_{k=1}^{K_p} \mathfrak{c}_k^{(p)}\phi_k^{(p)}(t_p), \quad t_p \in \mathcal{T}_p, \quad K_p \geq 1, \quad 1 \leq p \leq P, \tag{8}$$

be the truncated Karhunen-Loève expansion of the features of the process X.

4.1 Diagonalization of the covariance operator

The estimation of the eigencomponents of the covariance Γ by its diagonalization is derived in Happ and Greven (2018) for a general class of multivariate functional data defined on different dimensional domains. They give a direct relationship between the truncated representation (8) of the single elements $X^{(p)}$ and the truncated representation (7) of the multivariate functional data X (see Happ and Greven, 2018, Prop. 5).

We recall here, how to estimate the eigencomponents. Following Happ and Greven (2018, Prop. 5), the multivariate components for X are estimated by a weighted combination of the univariate components computed from each $X^{(p)}$. These estimations are done as follows. First, we perform a univariate FPCA on each of the features of X separately. For a feature $X^{(p)}$, the eigenfunctions and eigenvectors are computed as a matrix decomposition of the estimated covariance $C_{p,p}$ from Equation (1). This results in a set of eigenfunctions $\{\varphi_k^{(p)}\}_{1\leq k \leq K_p}$ associated with a set of eigenvalues $\{\lambda_k^{(p)}\}_{1\leq k \leq K_p}$ for a given truncation integer K_p. Then, the univariate scores for a realization $X_n^{(p)}$ of $X^{(p)}$ are given by $\mathbf{c}_{nk}^{(p)} = \langle X_n^{(p)}, \varphi_k^{(p)} \rangle$, $1 \leq k \leq K_p$. These scores might be estimated by numerical integration for example. Considering $K_+ \coloneqq \sum_{p=1}^{P} K_p$, we then define the matrix $\mathcal{Z} \in \mathbb{R}^{N \times K_+}$, where on each row we concatenate the scores obtained for the P features of the nth observation: $(\mathbf{c}_{n1}^{(1)}, \ldots, \mathbf{c}_{nK_1}^{(1)}, \ldots, \mathbf{c}_{n1}^{(P)}, \ldots, \mathbf{c}_{nK_P}^{(P)})$. An estimation of the covariance of the matrix $\mathcal{Z}$ is given by $\mathbf{Z} = (N-1)^{-1}\mathcal{Z}^\top\mathcal{Z}$. An eigenanalysis of the matrix $\mathbf{Z}$ is carried out to estimate the eigenvectors $\boldsymbol{v}_k$ and eigenvalues λ_k. And finally, the multivariate eigenfunctions are estimated with

$$\phi_k^{(p)}(t_p) = \sum_{l=1}^{K_p} [\boldsymbol{v}_k]_l^{(p)}\varphi_l^{(p)}(t_p), \quad t_p \in \mathcal{T}_p, \quad 1 \leq k \leq K_+, \quad 1 \leq p \leq P,$$

where $[\boldsymbol{v}_k]_l^{(p)}$ denotes the lth entry of the pth block of the vector $\boldsymbol{v}_k$. The multivariate scores are estimated as

$$\mathfrak{c}_{nk} = \mathcal{Z}_{n,\cdot}\boldsymbol{v}_k, \quad 1 \leq n \leq N, \quad 1 \leq k \leq K_+.$$

We let the reader refer to Happ and Greven (2018) for the derivation of the eigencomponents of the covariance operator if the curves are expanded in a general basis of functions.

4.2 Diagonalization of the inner product matrix

We can use the duality relation between row and column spaces of a data matrix to estimate the eigencomponents of the covariance operator. Consider the inner-product matrix M, with entries defined in (2). Let $\{l_k\}_{1 \leq k \leq N}$ such that $l_1 \geq \cdots \geq l_N \geq 0$ be the set of eigenvalues and $\{v_k\}_{1 \leq k \leq N}$ be the set of eigenvectors of the matrix M. The relationship between all nonzero eigenvalues of the covariance operator Γ and the eigenvalues of M is given by

$$\lambda_k = \frac{l_k}{N}, \quad k = 1, 2, \ldots, N,$$

while the relationship between the multivariate eigenfunctions of the covariance operator Γ and the orthonormal eigenvectors of M is given by

$$\phi_k(\mathbf{t}) = \sum_{n=1}^{N} [v_k]_n \left\{ X_n(\mathbf{t}) - \mu(\mathbf{t}) \right\}, \quad \mathbf{t} \in \mathcal{T}, \quad k = 1, 2, \ldots, N,$$

where $[v_k]_n$ is the nth entry of the vector v_k. The scores are then computed as the inner-product between the multivariate curves and the multivariate eigenfunctions and are given by

$$\mathfrak{c}_{nk} = \sqrt{l_k}[v_k]_n, \quad n = 1, 2, \ldots, N, \quad k = 1, 2, \ldots, N.$$

These results can be extended in a natural way if all the curves are expanded in a general basis of functions, defined in Equation (3). The derivation of these equalities, as well as the derivation of the eigencomponents using the expansion of the curves in a general basis of function and the Gram matrix, are given in Appendix B.

4.3 Computational complexity

We describe the time complexity for the computation of the MFPCA algorithm using the covariance operator and the inner product matrix. Considering the observation of N curves with P features, we assume that all observations of the feature p are sampled on a common grid of M_p points. For $a \in \mathbb{N}$, let $M^a = \sum_p M_p^a$. Let K be the number of multivariate eigenfunctions to estimate. For the estimation of the eigencomponents using the covariance operator, we have $K \leq K_+$. While K has the same interpretation for both the eigendecomposition of the covariance operator and the eigendecomposition of the inner product matrix, in the latter case, it is not computed as the summation over the univariate elements, but rather as the number of components needed to achieve a certain amount of variance explained. Here, we also assume that the curves are perfectly observed, and thus no smoothing step is included in the expression of the time complexity. Note that, the smoothing step will often have the same impact on complexity between the approaches. To estimate the time complexity of an algorithm, we count the number of elementary operations performed, considering a fixed execution time for each. Worst-case time complexity is usually considered; and this is what we do in the following. We first give the time complexity for the estimation of the eigencomponents using the covariance operator by explaining the time complexity of each individual step (see Happ and Greven (2018) and Section 4.1). For each feature p, the time complexity of the estimation of the covariance matrix is $\mathcal{O}(NM_p^2)$, of the eigendecomposition of the matrix is $\mathcal{O}(M_p^3)$ and of the univariate score is $\mathcal{O}(NM_pK_p)$. Therefore, the total time complexity is the sum over the p univariate time complexities. The covariance matrix $\mathbf{Z}$ of the stacked univariate scores $\mathcal{Z}$ is then computed with a time complexity of $\mathcal{O}(NK_+^2)$, because the dimension of the matrix $\mathcal{Z}$ is $N \times K_+$. The eigendecomposition of the matrix $\mathbf{Z}$ has a time complexity of $\mathcal{O}(K_+^3)$. The final step is to compute the multivariate eigenfunctions and scores. For the estimation of the $K \leq K_+$ multivariate eigenfunctions, the time complexity is $\mathcal{O}(K \sum_p M_p K_p)$ and for the estimation of the scores, the time complexity is $\mathcal{O}(NK^2)$. Gathering all the results, the final complexity of the estimation of the eigencomponents using the eigendecomposition of the covariance operator is

$$\mathcal{O}\left(NM^2 + M^3 + N \sum_{p=1}^{P} M_p K_p + NK_+^2 + K_+^3 + K \sum_{p=1}^{P} M_p K_p + NK^2 \right).$$

We now consider the time complexity of the estimation of the eigencomponents using the eigendecomposition of the inner product matrix (see Section 4.2). The inner product between two curves can be estimated in $\mathcal{O}(M^1)$. Since there are N^2 terms in the matrix, the time complexity for the computation of the inner product matrix is then $\mathcal{O}(N^2M^1)$. The eigendecomposition of this matrix has a time complexity of $\mathcal{O}(N^3)$. For the multivariate eigenfunctions, the time complexity is

$\mathcal{O}(KNP)$ and is $\mathcal{O}(KN)$ for the multivariate scores. Gathering all the results, the final complexity of the estimation of eigencomponents using the eigendecomposition of the inner product matrix is

$$\mathcal{O}\left(N^2 M^1 + N^3 + KNP + KN\right).$$

The number of components K to estimate is usually small compared to the number of curves N or to the total number of sampling points M^1. Both time complexities can then be reduced to $\mathcal{O}(NM^2 + M^3)$ for the diagonalization of the covariance operator and to $\mathcal{O}(N^2 M^1 + N^3)$ using the Gram matrix. If the number of observations is large compared to the total number of sampling points, it thus seems preferable to use the covariance operator to estimate the eigencomponents, while if the total number of sampling points is large compare to the number of observations, the use of the Gram matrix seems better. Note that the number of features P does not have much impact on the computational complexity. These results are confirmed in the simulation (see Section 5.2).

Remark 3. *We can use singular values decomposition (SVD) in both cases to make the algorithm faster as it allows to compute only the first K eigenfunctions.*

5   Empirical analysis

Using simulated data, we compare the behavior of the estimation of the eigencomponents using the diagonalization of the covariance operator and the Gram matrix. The diagonalization of the covariance operator is performed using the methodology of Happ and Greven (2018). As this methodology is based on the expansion of each univariate feature into univariate principal components, we used univariate FPCA, if the curves are unidimensional, and the Functional Canonical Polyadic-Tensor Power Algorithm (FCP-TPA) for regularized tensor decomposition (Allen, 2013), if the curves are two-dimensional. We choose the FCP-TPA as it is the one used by Happ and Greven (2018) in their algorithm and implemented in their software (Happ-Kurz, 2020). Note that we could also use a two-dimensional basis expansion such as penalized tensor splines or discrete cosine transform, but we do not investigate this expansion here, as we do not want to prespecify a basis of functions.

The results of the simulation are compared using computation times (CT), the integrated squared error (ISE) risk function for the multivariate eigenfunctions, the log-absolute error $(\log-\mathrm{AE})$ risk function for the eigenvalues and the mean integreated squared error (MISE) risk function for the reconstructed data. Let ϕ_k be the true eigenfunction and $\widehat{\phi}_k$ the estimated eigenfunction defined on $\mathcal{T}$. We then define the ISE as

$$\mathrm{ISE}(\phi_k, \widehat{\phi}_k) = \left\|\phi_k - \widehat{\phi}_k\right\|^2 = \sum_{p=1}^{P} \int_{\mathcal{T}_p} \{\phi_k^{(p)}(t_p) - \widehat{\phi}_k^{(p)}(t_p)\}^2 \mathrm{d}t_p, \quad k = 1, \dots, K. \tag{9}$$

Let $\lambda = (\lambda_1, \dots, \lambda_K)$ be the vector of true eigenvalues and $\widehat{\lambda} = (\widehat{\lambda}_1, \dots, \widehat{\lambda}_K)$ be the vector of estimated eigenvalues. We then define the $\log-\mathrm{AE}$ as

$$\log-\mathrm{AE}(\lambda_k, \widehat{\lambda}_k) = \log(|\lambda_k - \widehat{\lambda}_k|), \quad k = 1, \dots, K. \tag{10}$$

Let $\mathcal{X}$ be the set of true data and $\widehat{\mathcal{X}}$ be the set of reconstructed data. We define the MISE of the reconstructed data as

$$\mathrm{MISE}(\mathcal{X}, \widehat{\mathcal{X}}) = \frac{1}{N} \sum_{n=1}^{N} \left\|X_n - \widehat{X}_n\right\|^2 = \frac{1}{N} \sum_{n=1}^{N} \sum_{p=1}^{P} \int_{\mathcal{T}_p} \left\{X_n^{(p)}(t_p) - \widehat{X}_n^{(p)}(t_p)\right\}^2 \mathrm{d}t_p. \tag{11}$$

Each integral is approximated by the trapezoidal rule with an equidistant grid. Note $\widehat{\phi}, \widehat{\lambda}$ and $\widehat{\mathcal{X}}$ the different estimators using the Gram matrix and $\widetilde{\phi}, \widetilde{\lambda}$ and $\widetilde{\mathcal{X}}$ the different estimators using the covariance operator. For each simulation, we compute the ratios

$$\frac{\mathrm{ISE}(\phi_k, \widehat{\phi}_k)}{\mathrm{ISE}(\phi_k, \widetilde{\phi}_k)}, \quad \frac{\log-\mathrm{AE}(\lambda_k, \widehat{\lambda}_k)}{\log-\mathrm{AE}(\lambda_k, \widetilde{\lambda}_k)}, \quad k = 1, \dots, K, \quad \text{and} \quad \frac{\mathrm{MISE}(\mathcal{X}, \widehat{\mathcal{X}})}{\mathrm{MISE}(\mathcal{X}, \widetilde{\mathcal{X}})},$$

and compare them to 1.

5.1 Simulation experiments

We consider two simulation scenarios. One consists of multivariate functional data with univariate features defined as unidimensional curves and the other consists of univariate functional data of two-dimensional surfaces. Each experiment is repeated 500 times.

Scenario 1. The simulation setting is based on the simulation in Happ and Greven (2018). The data generating process is based on a truncated version of the Karhunen-Loève decomposition. First, we generate a large orthonormal basis $\{\psi_k\}_{1 \leq k \leq K}$ of $\mathcal{L}^2(\mathcal{T})$ on an interval $\mathcal{T} = [0, T] \subset \mathbb{R}$. We fix $T_1 = 0$ and $T_{P+1} = T$ and we generate $P - 1$ cutting points $T_2, \ldots, T_P$ uniformly in $\mathcal{T}$ such that $0 = T_1 < \cdots < T_P < T_{P+1} = T$. Let $s_1, \ldots, s_P \in \{-1, 1\}$ be coefficients that randomly flip the eigenfunctions. The univariate components of the eigenfunctions are then defined as

$$\phi_k^{(p)}(t_p) = s_p \psi_k\big|_{[T_p, T_{p+1}]}\left(\frac{t_p - T_p}{T_{p+1} - T_p}\right), \quad k = 1, \ldots, K, \quad p = 1, \ldots, P.$$

The notation $\phi_k\big|_{[T_p, T_{p+1}]}$ is the restriction of the function ϕ_k to the set $[T_p, T_{p+1}]$. The set of multivariate functions $\{\psi_k\}_{1 \leq k \leq K}$ is an orthonormal system in $\mathcal{H} := \mathcal{L}^2(\mathcal{T}_1) \times \cdots \times \mathcal{L}^2(\mathcal{T}_P)$ with $\mathcal{T}_p = [0, 1]$. Each curve is then simulated using the truncated multivariate Karhunen-Loève expansion (7):

$$X(\mathbf{t}) = \sum_{k=1}^{K} \mathfrak{c}_k \phi_k(\mathbf{t}), \quad \mathbf{t} \in \mathcal{T},$$

where the scores $\mathfrak{c}_k$ are sampled as random normal variables with mean 0 and variance λ_k. The eigenvalues λ_k are defined with an exponential decrease, $\lambda_k = \exp(-(k+1)/2)$. We simulate, for each replication of the simulation, $N = 25, 50, 75$ and 100 observations. Similarly, each component is sampled on a regular grid of $M = 25, 50, 75$ and 100 sampling points. We compare the methods for $P = 2, 10, 20$ and 50 features and we set $K = 10$.

Scenario 2. The data generating process is again based on a truncated version of the Karhunen-Loève decomposition. First, we generate an orthonormal basis $\{\phi_k\}_{1 \leq k \leq K}$ of $\mathcal{L}^2(\mathcal{T})$ on an interval $\mathcal{T} = [0, 1] \times [0, 1]$ as the tensor product of the first Fourier basis functions:

$$\phi_k(s, t) = \psi_l(s) \otimes \psi_m(t), \quad s, t \in [0, 1], \quad k = 1, \ldots, K,$$

where ψ_l and ψ_m are elements of the Fourier basis. Each curve is then simulated using the truncated multivariate Karhunen-Loève expansion (7):

$$X(s, t) = \sum_{k=1}^{K} \mathfrak{c}_k \phi_k(s, t), \quad s, t \in [0, 1],$$

where the scores $\mathfrak{c}_k$ are defined as for the Scenario 1. We simulate, for each replication of the simulations, $N = 25, 50, 75$ and 100 observations. Similarly, each component is sampled on a regular grid of $M = 25 \times 25, 50 \times 50, 75 \times 75$ and 100×100 sampling points. We set $K = 10$.

5.2 Simulation results

We compared the MFPCA using the diagonalisation of the covariance operator and the diagonalisation of the Gram matrix on their computational time, estimation of eigenvalues, estimation of eigenfunctions, and reconstruction of curves. The results are presented below.

Computation time. To compare the computational time of the diagonalisation of the covariance operator and the diagaonlisation of the Gram matrix, we measured the time it took for each method to complete the MFPCA for each simulated dataset. For the Scenario 1, we found that the diagonalisation of the covariance has a shorter CT compared to the diagonalisation of the Gram matrix across all sample sizes and number of functions. Figure 5 shows the kernel density estimation of the ratio of CT for each method across all sample sizes, number of sampling points and number of features. Note that the number of features P does not change the ratio of CT between the methods. For the Scenario 2, we found that the diagonalisation of the Gram matrix has a shorter CT compared to the diagonalisation of the Gram matrix across all sample sizes and number of sampling points. Figure 6 shows the kernel density estimation of the ratio of CT for each method across all sample sizes and number of sampling points. The shorter computational time of the diagonalisation of the Gram matrix for the Scenario 2 makes it a more efficient option
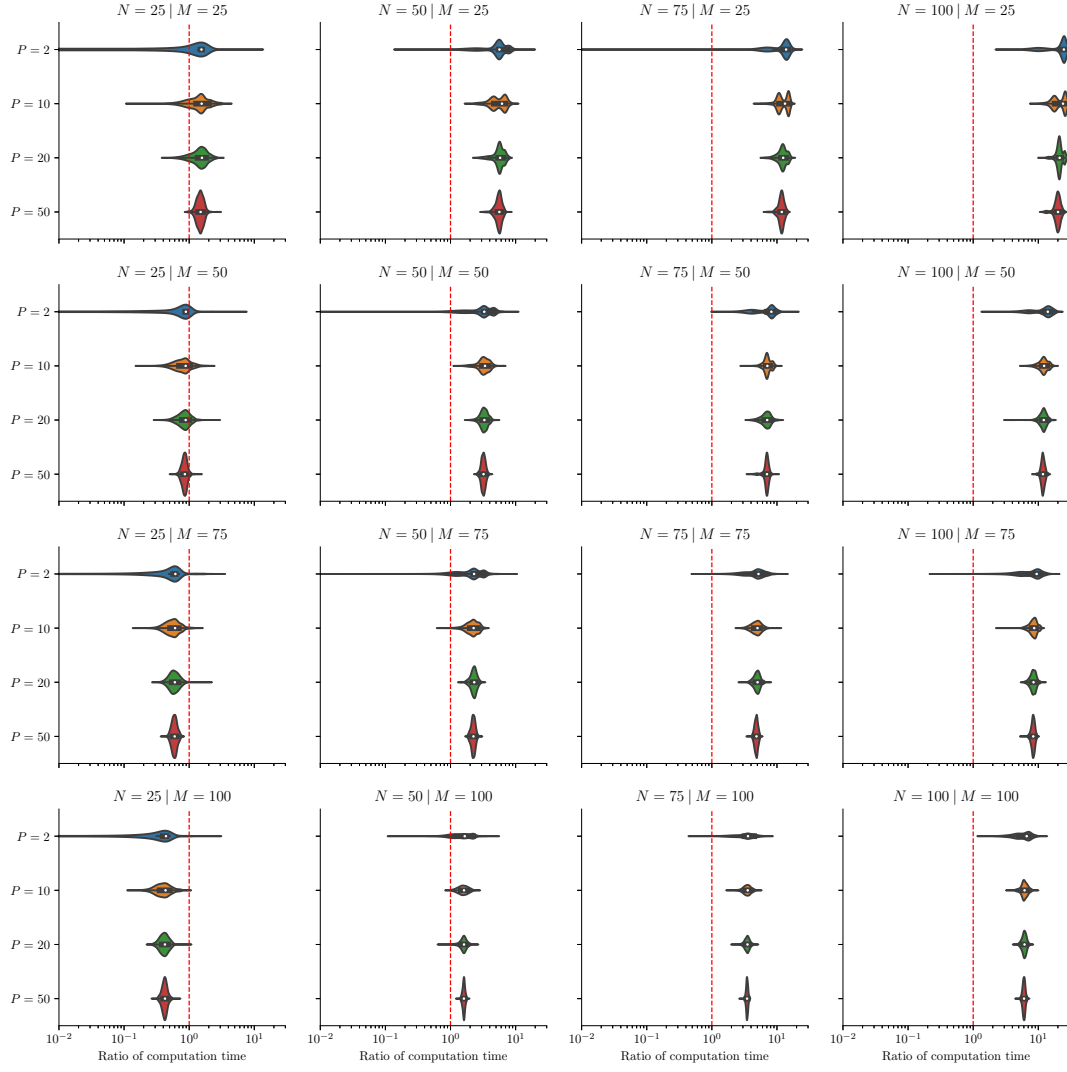
Figure 5: Ratio of computation time for Scenario 1 between the Gram matrix method and the covariance operator method. Each univariate component is defined on a one-dimensional domain. N is the number of observation, M is the number of sampling points per curve and P is the number of features.
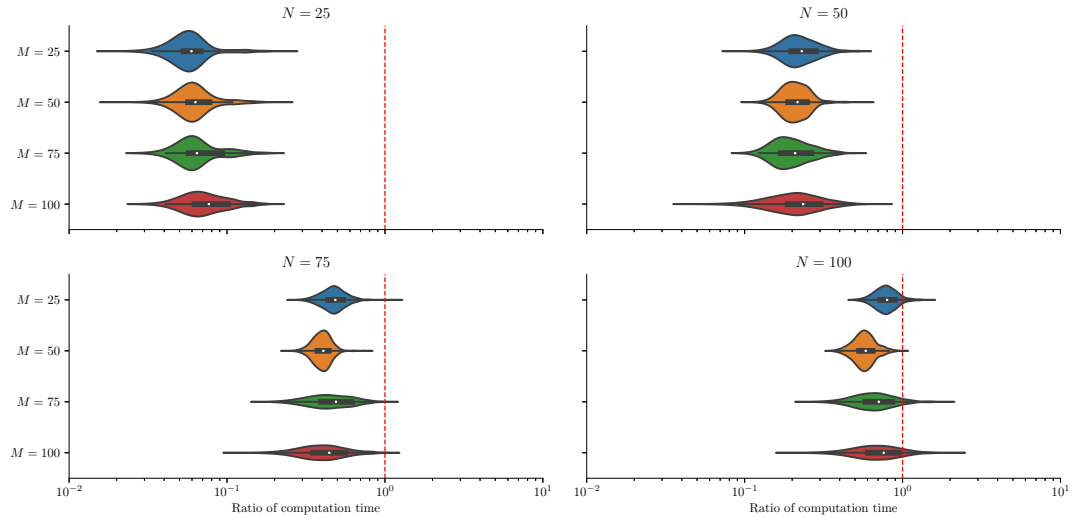

Figure 6: Ratio of computation time for Scenario 2 between the Gram matrix method and the covariance operator method. N is the number of observation and $M \times M$ is the number of sampling points per images.
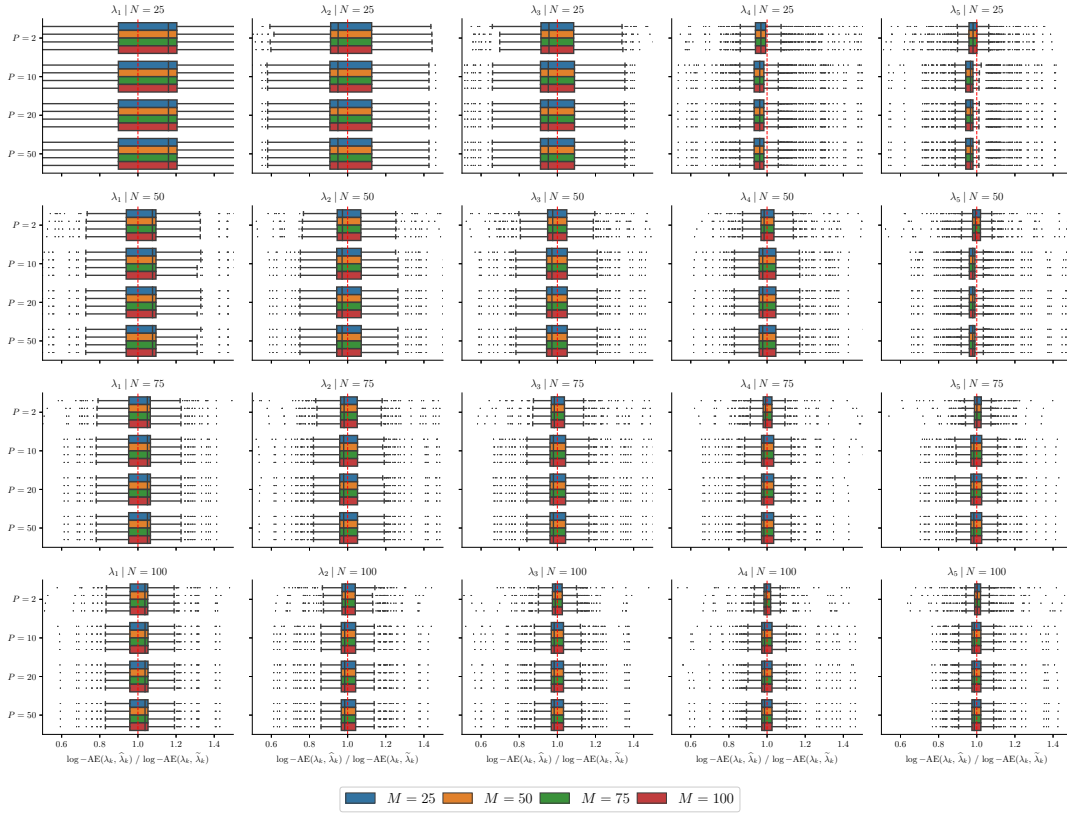
Figure 7: Ratio of $\log{-\mathrm{AE}}$ for Scenario 1 between the Gram matrix method and the covariance operator method. Each univariate component is defined on a one-dimensional domain. N is the number of observation, M is the number of sampling points per curve and P is the number of features.

for analyzing two and higher-dimensional functional datasets. It is however worth noting that the computational time can still vary depending on the specific implementation of each method, the computational resources available, and the complexity of the dataset (number of observations, number of sampling points, etc.).

Eigenvalues estimation. To compare the estimation of the eigenvalues between the diagonalisation of the covariance operator and the diagaonlisation of the Gram matrix, we calculated the ratio of the $\log{-\mathrm{AE}}$ (10) between the estimated eigenvalues and the true eigenvalues for each simulated dataset and for the first five eigenvalues. For the Scenario 1, we found that the two methods behave similarly for all considered settings. Figure 7 shows the boxplots of the $\log{-\mathrm{AE}}$ for each method across all sample sizes, number of sampling points and number of features. For the Scenario 2, we found that the FCP-TPA gives slighty better estimation of the eigenvalues. Figure 8 shows the boxplots of the $\log{-\mathrm{AE}}$ for each method across all sample sizes and number of sampling points.

Eigenfunctions estimation. To compare the estimation of the eigenfunctions between the diagonalisation of the covariance operator and the diagaonlisation of the Gram matrix, we calculated the ratio of the ISE (9) between the estimated eigenfunctions and the true eigenfunctions for each simulated dataset and for the first five eigenfunctions. For the Scenario 1, we found that the two methods behave similarly for all considered settings. For $P = 10, 20$ and 50, the results are identical. Figure 9 shows the boxplots of the ISE for each method across all sample sizes, number of sampling points and number of features. For the Scenario 2, we found that the decomposition of the Gram matrix gives better estimation of the eigenfunctions compared to the FCP-TPA, especially the quality of the estimation increase when the number of observations increases. Figure 10 shows the boxplots of the ISE for each method across all sample sizes and number of sampling points.

Curves reconstruction. To compare the quality of the reconstruction of the curves between the diagonalisation of the covariance operator and the diagaonlisation of the Gram matrix, we calculated the ratio of the MISE (11) between the reconstruction of the curves and the true curves
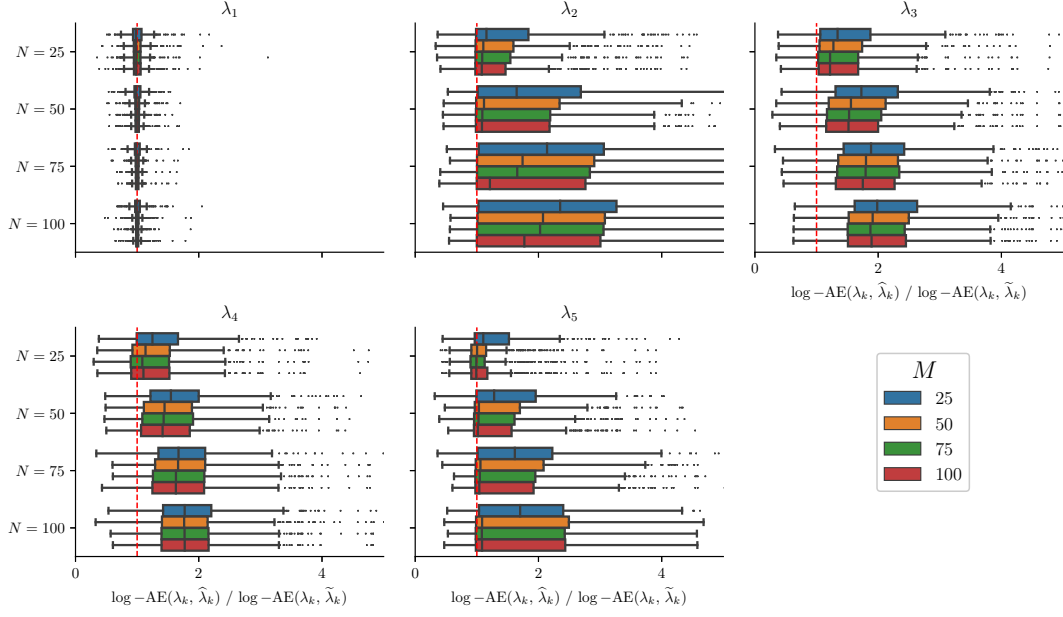
13

Figure 8: Ratio of $\log-\mathrm{AE}$ for Scenario 2 between the Gram matrix method and the covariance operator method. N is the number of observation and $M \times M$ is the number of sampling points per images.
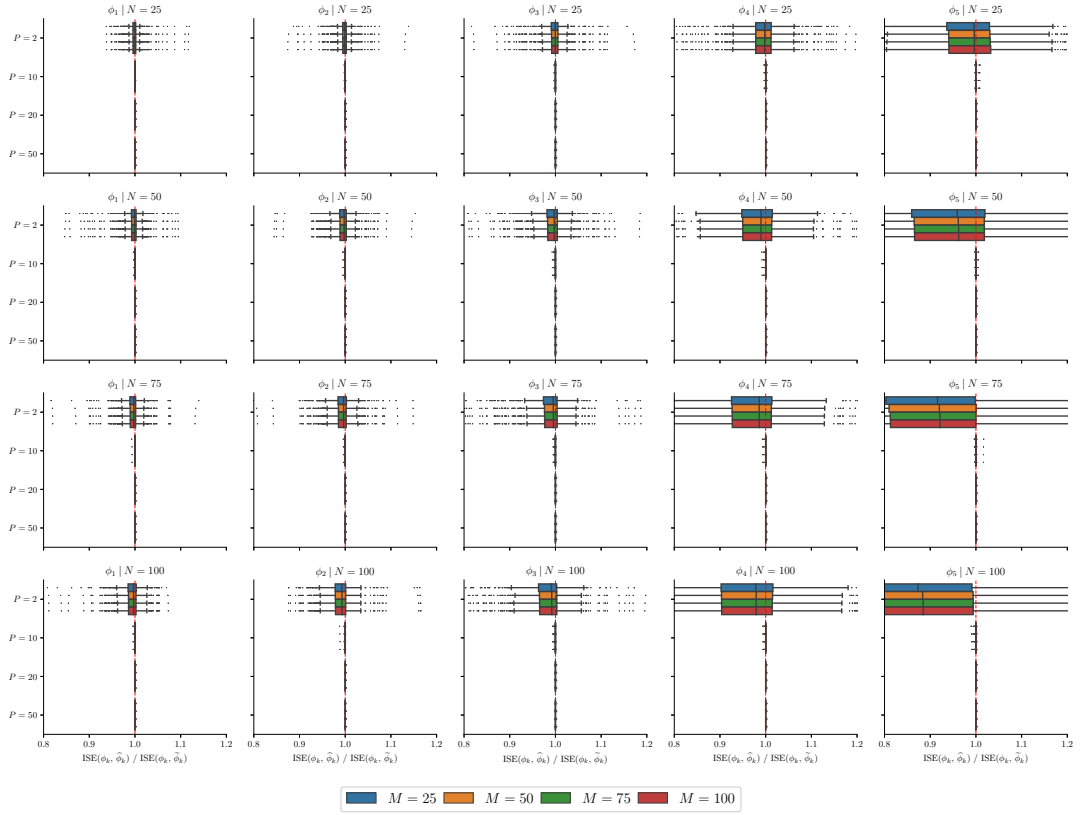

Figure 9: Ratio of ISE for Scenario 1 between the Gram matrix method and the covariance operator method. Each univariate component is defined on a one-dimensional domain. N is the number of observation, M is the number of sampling points per curve and P is the number of features.
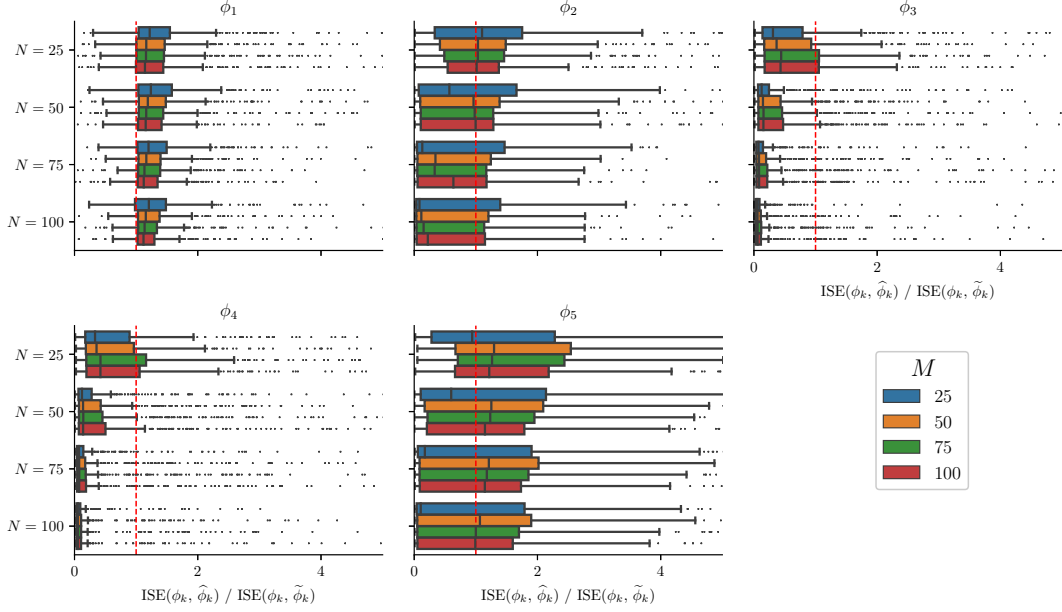
Figure 10: Ratio of ISE for Scenario 2 between the Gram matrix method and the covariance operator method. N is the number of observation and $M \times M$ is the number of sampling points per images.

for each simulated dataset. For ~~the~~ Scenario 1, we found that the diagonalization of the Gram matrix gives sli~~ghlt~~y better results for all considered settings. Figure 11 shows the boxplots of the ISE for each method across all sample sizes, number of sampling points and number of features. Similarly, for ~~the~~ Scenario 2, we found that the decomposition of the Gram matrix gives better estimation of the ~~eigenfunctions~~ compared to the FCP-TPA, especially ~~the quality of the estimation increase~~ when the number of observations increases. Figure 12 shows the boxplots of the ISE for each method across all sample sizes and number of sampling points.

Overall, our simulations showed that the diagonalisation of the covariance operator or of the Gram matrix gave similar results in terms of the estimation of the eigenvalues, the eigenfunctions and reconstruction of the curves for one-dimensional multivariate functional data, while the performance of the diagonalisation of the Gram matrix outperformed the FCP-TPA for higher-dimensional functional datasets. Regarding the computation time, the use of the covariance operator is faster in most ~~of the~~ case for multivariate functional data defined on unidimensional domains. The only situation where the use of the Gram matrix is quicker for $M \gg N$. For data defined on higher-dimensional domains, diagonalisation of the Gram matrix is faster. In conclusion, we recommend to use the covariance operator for multivariate functional data with all features defined on a one-dimensional domeins (curves) and for a number of observations larger or comparable to the number of sampling points regardless of the number of features. If the data are defined on multi-dimensional domains (images) or the number of sampling points is much higher than the number of observations, we ~~can~~ advise ~~to use~~ the Gram matrix.

6 Discussion and conclusion

MFPCA is a fundamental statistical tool utilized for the analysis of multivariate functional data, which enables to capture the variability in observations defined by multiple curves. In this paper, we have described the duality between rows and columns of a data matrix within the context of multivariate functional data. We have proposed to use this duality to estimate the eigencomponents of the covariance operator in multivariate functional datasets. By comparing the results of the two methods, we have provided the researcher with guidelines for determining the most appropriate method for application within a range of functional data frameworks. In summary, if the number of sampling points is significantly greater than the number of observations, or if the data at hand are multidimensional (e.g., surfaces), it is preferable to estimate the eigencomponents utilizing the Gram matrix. Conversely, if the data are unidimensional with a large number of observations, the preferred method is the direct decomposition of the covariance operator, regarless of the number of features.
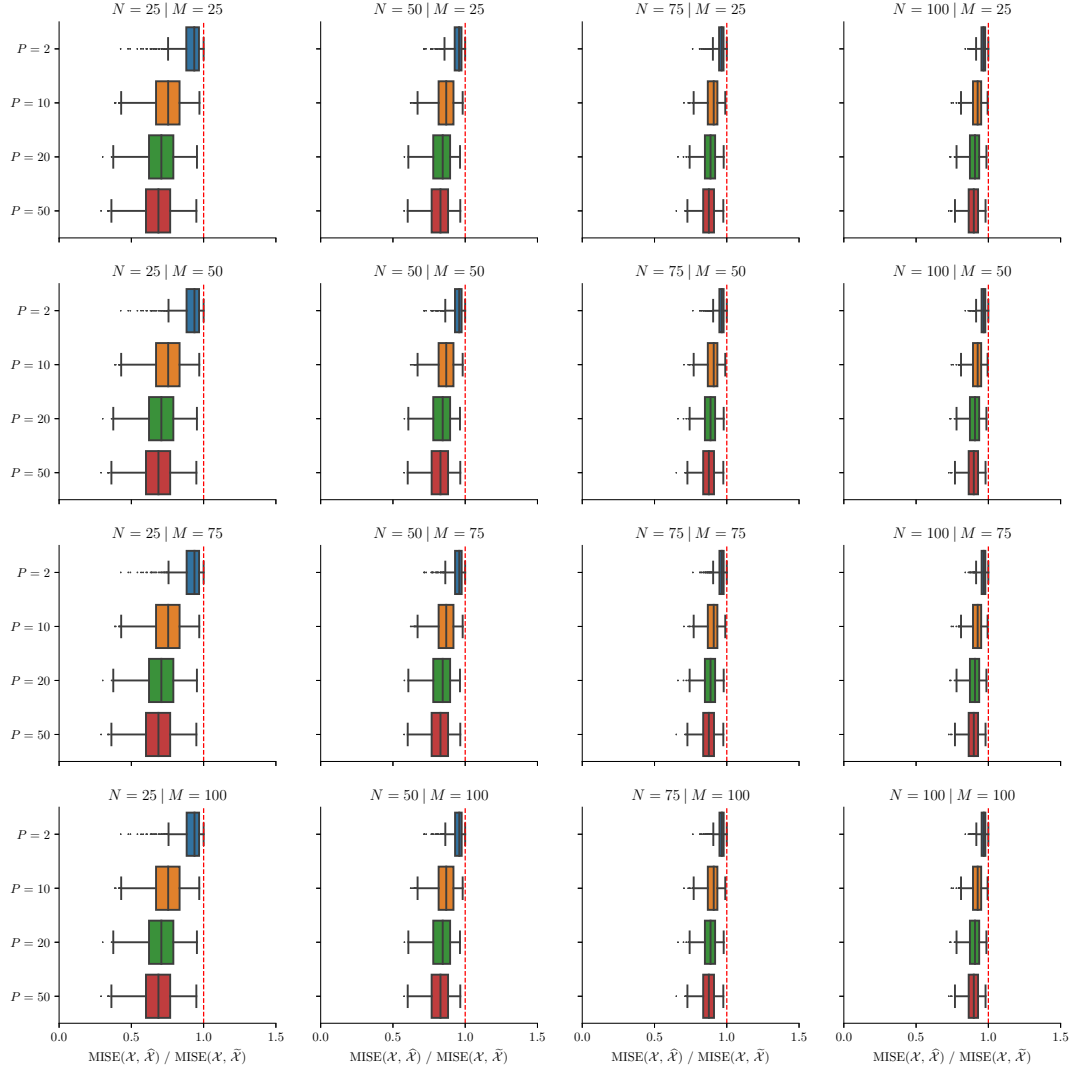
Figure 11: Ratio of MISE for Scenario 1 between the Gram matrix method and the covariance operator method. Each univariate component is defined on a one-dimensional domain. N is the number of observation, M is the number of sampling points per curve and P is the number of features.
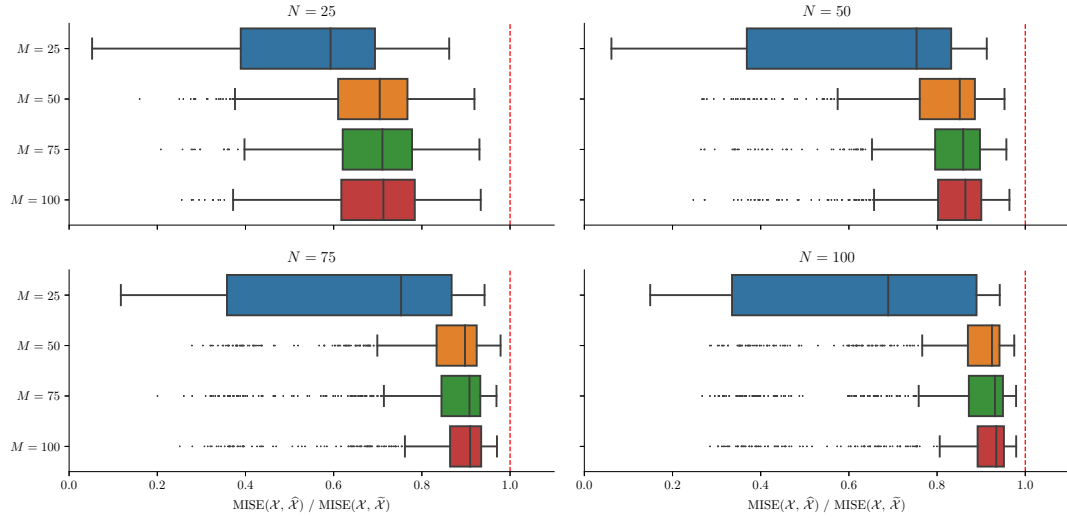

Figure 12: Ratio of MISE for Scenario 2 between the Gram matrix method and the covariance operator method. N is the number of observation and $M \times M$ is the number of sampling points per images.

Utilizing the Gram matrix enables the estimation of the number of components retained via the percentage of variance accounted for by the multivariate functional data, whereas the decomposition of the covariance operator necessitates the specification of the percentage of variance accounted for by each individual univariate feature. The necessity of specifying the percentage of variance explained for each features ~~can~~ not guarantees that we recover the ~~right~~ percentage of variance explained for the multivariate data.

In practice, observations of (multivariate) functional data are often subject to noise. As we recommend the use of the Gram matrix solely for densely sampled functional datasets, individual curve smoothing should suffice to approximate the Gram matrix in such cases. The estimation of the Gram matrix in the context of sparsely sampled functional data is however deemed irrelevant, given our findings that the utilization of the covariance operator for the estimation of the eigencomponents yields comparable results, while typically requiring less computational time.

The open-source implementation can be accessed at `https://github.com/StevenGolovkine/FDApy`, while scripts to reproduce the simulation are at `https://github.com/FAST-ULxNUIG/geom_mfpca`.

A   Derivation of the inertia of the clouds

In this Section, we derive the inertia of the cloud $\mathcal{C}_N$. Recall that

$$\text{Var}\{X^{(p)}(t_p)\} = \sum_{n=1}^{N} \pi_n \{X_n^{(p)}(t_p)\}^2 - \{\mu^{(p)}(t_p)\}^2 \quad \text{where} \quad \mu^{(p)}(t_p) = \sum_{n=1}^{N} \pi_n X_n^{(p)}(t_p), \quad t_p \in \mathcal{T}_p.$$

$$
\begin{aligned}
\sum_{n=1}^{N} \pi_n d^2(X_n, \mu) &= \sum_{n=1}^{N} \pi_n \sum_{p=1}^{P} \left\| X_n^{(p)} - \mu^{(p)} \right\|^2 \\
&= \sum_{p=1}^{P} \left(\sum_{n=1}^{N} \pi_n \left\| X_n^{(p)} \right\|^2 - \left\| \mu^{(p)} \right\|^2 \right) \\
&= \sum_{p=1}^{P} \int_{\mathcal{T}_p} \text{Var}\, X^{(p)}(t_p) \mathrm{d}t_p,
\end{aligned}
$$

$$
\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{N} \pi_i \pi_j d^2(X_i, X_j) &= \sum_{i=1}^{N}\sum_{j=1}^{N} \pi_i \pi_j \sum_{p=1}^{P} \left\| X_i^{(p)} - X_j^{(p)} \right\|^2 \\
&= \sum_{p=1}^{P} \left(2\sum_{i=1}^{N} \pi_i \left\| X_i^{(p)} \right\|^2 - 2\sum_{i=1}^{N}\sum_{j=1}^{N} \langle X_i^{(p)}, X_j^{(p)} \rangle \right) \\
&= \sum_{p=1}^{P} \left(2\sum_{i=1}^{N} \pi_i \left\| X_i^{(p)} \right\|^2 - 2\left\| \mu^{(p)} \right\|^2 - 2\sum_{i=1}^{N}\sum_{j=1}^{N} \langle X_i^{(p)}, X_j^{(p)} \rangle + 2\left\| \mu^{(p)} \right\|^2 \right) \\
&= 2\sum_{p=1}^{P} \int_{\mathcal{T}_p} \text{Var}\, X^{(p)}(t_p) \mathrm{d}t_p.
\end{aligned}
$$

The equalities in Equation (4) are shown.

B   Derivation of the eigencomponents

Using the Hilbert-Schmidt theorem, there exists a complete orthonormal basis of eigenvectors $\{v_k\}_{1 \le k \le N}$ of the inner-product matrix $\mathbf{M}$ such that

$$\mathbf{M}v_k = l_k v_k. \tag{12}$$

Let $X = (X_1 - \mu, \ldots, X_N - \mu)^\top$. Recall that, in the case of P-dimensional process, the realisations of the process X_n, $n = 1, \cdots, N$ and μ are vectors of functions of length P, and thus X is a matrix of functions of size $N \times P$. By left multiplying Equation (12) by $X^\top$, we obtain

$$X^\top \mathbf{M} v_k = l_k X^\top v_k. \tag{13}$$

Expanding Equation (13), for each component $p = 1, \ldots, P$, we have,

$$\sum_{i=1}^{N} \sum_{j=1}^{N} [v_k]_j \left\{ X_i^{(p)}(\cdot) - \mu^{(p)}(\cdot) \right\} \langle\!\langle X_i - \mu, X_j - \mu \rangle\!\rangle = l_k \sum_{n=1}^{N} [v_k]_n \left\{ X_n^{(p)}(\cdot) - \mu^{(p)}(\cdot) \right\}. \tag{14}$$

Here and in the following, we note $[a]_p$ the pth entry of the vector a. Starting from the left side of Equation (14), we get

$$\begin{aligned} [X^{\top} \mathbf{M} v_k]_p &= \sum_{i=1}^{N} \sum_{j=1}^{N} [v_k]_j \left\{ X_i^{(p)}(\cdot) - \mu^{(p)}(\cdot) \right\} \langle\!\langle X_i - \mu, X_j - \mu \rangle\!\rangle \tag{15} \\ &= \sum_{q=1}^{P} \int_{\mathcal{T}_q} \sum_{i=1}^{N} \left\{ X_i^{(p)}(\cdot) - \mu^{(p)}(\cdot) \right\} \left\{ X_i^{(q)}(s_q) - \mu^{(q)}(s_q) \right\} \\ &\qquad \sum_{j=1}^{N} [v_k]_j \left\{ X_j^{(q)}(s_q) - \mu^{(q)}(s_q) \right\} \mathrm{d}s_q \\ &= \sum_{q=1}^{P} \int_{\mathcal{T}_q} N C_{pq}(\cdot, s_q) \sum_{j=1}^{N} [v_k]_j \left\{ X_j^{(q)}(s_q) - \mu^{(q)}(s_q) \right\} \mathrm{d}s_q \\ &= N \sum_{j=1}^{N} \langle\!\langle C_{p\cdot}(\cdot, \cdot), [v_k]_j \{ X_j - \mu \} \rangle\!\rangle \\ &= N \Gamma \left(\sum_{j=1}^{N} [v_k]_j \{ X_j - \mu \} \right)^{(p)}(\cdot) \end{aligned}$$

and, starting from the right side of Equation (14),

$$[l_k X^{\top} v_k]_p = l_k \sum_{n=1}^{N} [v_k]_n \left\{ X_n^{(p)}(\cdot) - \mu^{(p)}(\cdot) \right\}. \tag{16}$$

From Equation (15) and Equation (16), we obtain

$$\Gamma \left(\sum_{j=1}^{N} [v_k]_j \{ X_j - \mu \} \right)^{(p)}(\cdot) = \frac{l_k}{N} \sum_{n=1}^{N} [v_k]_n \left\{ X_n^{(p)}(\cdot) - \mu^{(p)}(\cdot) \right\}, \quad \text{for all } p = 1, \ldots, P.$$

By identification in Equation (5), we find that, for each components p,

$$\lambda_k = \frac{l_k}{N} \quad \text{and} \quad \phi_k^{(p)}(\cdot) = \sum_{n=1}^{N} [v_k]_n \left\{ X_n^{(p)}(\cdot) - \mu^{(p)}(\cdot) \right\}, \quad k \geq 1. \tag{17}$$

For $k \geq 1$, the norm of the eigenfunction is computed as the following:

$$\begin{aligned} \|\phi_k\|^2 &= \sum_{i=1}^{N} \sum_{j=1}^{N} [v_k]_i [v_k]_j \langle\!\langle X_i - \mu, X_j - \mu \rangle\!\rangle = \sum_{i=1}^{N} [v_k]_i \sum_{j=1}^{N} \mathbf{M}_{ij} [v_k]_j \\ &= \sum_{i=1}^{N} [v_k]_i l_k [v_k]_i = l_k \|v_k\|^2 = l_k. \end{aligned}$$

Therefore, in order to have an orthonormal basis of eigenfunctions, we normalise the eigenfunctions ϕ_k from Equation (17) by $1/\sqrt{l_k}$. Concerning the estimation of the scores, for $n = 1, \ldots, N$, for $k \geq 1$, we have

$$\begin{aligned} \widehat{\mathfrak{c}}_{nk} &= \langle\!\langle X_n - \mu, \phi_k \rangle\!\rangle = \frac{1}{\sqrt{l_k}} \sum_{j=1}^{N} [v_k]_j \langle\!\langle X_n - \mu, X_j - \mu \rangle\!\rangle \\ &= \frac{1}{\sqrt{l_k}} \sum_{j=1}^{N} [v_k]_j \mathbf{M}_{nj} = \sqrt{l_k} [v_k]_n. \end{aligned}$$

Rewrite this with centering. Concerning the expansion of the data into the basis of function Ψ, we write

$$M = \left(CW^{1/2}\right)\left(CW^{1/2}\right)^{\top}.$$

We also assume that $\widehat{\phi}_1, \widehat{\phi}_2, \ldots$ the eigenfunctions of $\widehat{\Gamma}_N$ have a decomposition into the basis Ψ

$$\widehat{\phi}_k(\cdot) = \begin{pmatrix} \widehat{\phi}_k^{(1)}(\cdot) \\ \vdots \\ \widehat{\phi}_k^{(P)}(\cdot) \end{pmatrix} = \begin{pmatrix} \psi^{(1)\top}(\cdot)b_{1k} \\ \vdots \\ \psi^{(P)\top}(\cdot)b_{Pk} \end{pmatrix}, \quad \text{where} \quad b_{pk} = \left(b_{pk1}, \ldots, b_{pkK_p}\right)^{\top}.$$

We have, for $p = 1, \ldots, P$,

$$\left(\widehat{\Gamma}_N \phi_k\right)^{(p)}(\cdot) = \sum_{q=1}^{P} \int_{\mathcal{T}_q} \widehat{C}_{pq}(\cdot, s_q)\phi_k^{(q)}(s_q)\mathrm{d}s_q$$

$$= \frac{1}{N}\sum_{q=1}^{P} \int_{\mathcal{T}_q} \Psi(\cdot)^{(p)\top}C^{(p)\top}C^{(q)}\Psi^{(q)}(s_q)\Psi^{(q)}(s_q)^{\top}b_{qk}\mathrm{d}s_q$$

$$= \frac{1}{N}\Psi(\cdot)^{(p)\top}C^{(p)\top}\sum_{q=1}^{P}C^{(q)}\int_{\mathcal{T}_q}\Psi^{(q)}(s_q)\Psi^{(q)\top}(s_q)^{(q)\top}\mathrm{d}s_q b_{qk}$$

$$= \frac{1}{N}\Psi(\cdot)^{(p)\top}C^{(p)\top}\sum_{q=1}^{P}C^{(q)}W^{(q)}b_{qk}.$$

This equation is true for all $p = 1, \cdots, P$, this can be rewritten with matrices as

$$\widehat{\Gamma}_N\phi_k(\cdot) = \frac{1}{N}\Psi^{\top}(\cdot)C^{\top}CWb_k.$$

From the eigenequation, we have that

$$\widehat{\Gamma}_N\phi_k(\cdot) = \lambda_k\phi_k(\cdot) \iff \frac{1}{N}\Psi(\cdot)^{\top}C^{\top}CWb_k = \lambda_k\Psi(\cdot)^{\top}b_k.$$

Since this equation must be true for all $t_p \in \mathcal{T}_p$, this imply the equation

$$C^{\top}CWb_k = N\lambda_k b_k. \tag{18}$$

As the eigenfunctions are assumed to be normalized, $\|\phi_k\|^2 = 1$. And so, $b_k^{\top}Wb_k = 1$. Let $u_k = W^{1/2}b_k$. Then, from Equation (18), we obtain

$$W^{1/2}C^{\top}CW^{1/2}u_k = N\lambda_k u_k \iff \left(CW^{1/2}\right)^{\top}\left(CW^{1/2}\right)u_k = N\lambda_k u_k. \tag{19}$$

From the eigendecomposition of the matrix M, we get

$$Mv_k = l_k v_k \iff \left(CW^{1/2}\right)\left(CW^{1/2}\right)^{\top}v_k = l_k v_k. \tag{20}$$

The equations (19) and (20) are eigenequations in the classical PCA case, with the duality $X^{\top}X$ and $XX^{\top}$. Following Pagès (2014); Härdle and Simar (2019), we find that, for $1 \le k \le K$,

$$\lambda_k = \frac{l_k}{N}, \quad v_k = \frac{1}{\sqrt{l_k}}CW^{1/2}u_k \quad \text{and} \quad u_k = \frac{1}{\sqrt{l_k}}W^{1/2}C^{\top}v_k.$$

And finally, to get the coefficient of the eigenfunctions, for $1 \le k \le K$,

$$b_k = W^{-1/2}u_k = \frac{1}{\sqrt{l_k}}C^{\top}v_k.$$

Acknowledgment

References

Allen, G. I. (2013). Multi-way functional principal components analysis. In *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 220–223.

Benko, M., Härdle, W., and Kneip, A. (2009). Common functional principal components. *The Annals of Statistics*, 37(1):1–34.

Berrendero, J., Justel, A., and Svarc, M. (2011). Principal components for multivariate functional data. *Computational Statistics & Data Analysis*, 55:2619–2634.

Chen, K., Zhang, X., Petersen, A., and Müller, H.-G. (2017). Quantifying Infinite-Dimensional Data: Functional Data Analysis in Action. *Statistics in Biosciences*, 9(2):582–604.

Chiou, J.-M., Chen, Y.-T., and Yang, Y.-F. (2014). Multivariate Functional Principal Component Analysis: A Normalization Approach. *Statistica Sinica*, 24(4):1571–1596.

Dauxois, J., Pousse, A., and Romain, Y. (1982). Asymptotic theory for the principal component analysis of a vector random function: Some applications to statistical inference. *Journal of Multivariate Analysis*, 12(1):136–154.

De la Cruz, O. and Holmes, S. (2011). The Duality Diagram in Data Analysis: Examples of Modern Applications. *The annals of applied statistics*, 5(4):2266–2277.

Escofier, B. (1979). Traitement simultané de variables qualitatives et quantitatives en analyse factorielle. *Cahiers de l'analyse des données*, 4(2):137–146.

Happ, C. and Greven, S. (2018). Multivariate Functional Principal Component Analysis for Data Observed on Different (Dimensional) Domains. *Journal of the American Statistical Association*, 113(522):649–659.

Happ-Kurz, C. (2020). Object-Oriented Software for Functional Data. *Journal of Statistical Software*, 93(1):1–38.

Härdle, W. K. and Simar, L. (2019). *Applied Multivariate Statistical Analysis*. Springer Nature.

Holmes, S. (2008). Multivariate data analysis: The French way. *Probability and Statistics: Essays in Honor of David A. Freedman*, 2:219–234.

Horvàth, L. and Kokoszka, P. (2012). *Inference for Functional Data with Applications*. Springer Series in Statistics.

Jacques, J. and Preda, C. (2014). Model-based clustering for multivariate functional data. *Computational Statistics & Data Analysis*, 71:92–106.

Karhunen, K. (1947). *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. PhD thesis, (Sana), Helsinki.

Kneip, A. and Utikal, K. J. (2001). Inference for Density Families Using Functional Principal Component Analysis. *Journal of the American Statistical Association*, 96(454):519–532.

Kokoszka, P., Oja, H., Park, B., and Sangalli, L. (2017). Special issue on functional data analysis. *Econometrics and Statistics*, 1:99–100.

Krzyśko, M., Nijkamp, P., Ratajczak, W., and Wołyński, W. (2022). Multidimensional economic indicators and multivariate functional principal component analysis (MFPCA) in a comparative study of countries' competitiveness. *Journal of Geographical Systems*, 24(1):49–65.

Loève, M. (1945). Sur les fonctions aléatoires stationnaires de second ordre. *La Revue Scientifique, 5. Série*, 83:297–303.

Pagès, J. (2014). *Multiple Factor Analysis by Example Using R*. CRC Press.

Prothero, J. B., Hannig, J., and Marron, J. S. (2021). New Perspectives on Centering.

Ramsay, J. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer Science & Business Media.

Ramsay, J. O. (1982). When the data are functions. *Psychometrika*, 47(4):379–396.

Reed, M. and Simon, B. (1980). *Methods of Modern Mathematical Physics: Functional Analysis.* Academic Press.

Saporta, G. (1990). Simultaneous Analysis of Qualitative and Quantitative Data. *Atti 35° Riunione Scientifica della Societa Italiana di Statistica*, pages pp. 63–72.

Song, J. and Kim, K. (2022). Sparse multivariate functional principal component analysis. *Stat*, 11(1):e435.

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

Warmenhoven, J., Cobley, S., Draper, C., Harrison, A., Bargary, N., and Smith, R. (2019). Bivariate functional principal components analysis: Considerations for use with multivariate movement signatures in sports biomechanics. *Sports Biomechanics*, 18(1):10–27.