Clustering multivariate functional data in group-specific functional subspaces

Amandine Schmutz, Julien Jacques, Charles Bouveyron, Laurence Cheze, Pauline Martin

Computational Statistics manuscript No.
(will be inserted by the editor)

Clustering multivariate functional data in group-specific functional subspaces

Amandine Schmutz · Julien Jacques · Charles Bouveyron · Laurence Chèze · Pauline Martin

Received: date / Accepted: date

Abstract With the emergence of numerical sensors in many aspects of everyday life, there is an increasing need in analyzing multivariate functional data. This work focuses on the clustering of such functional data, in order to ease their modeling and understanding. To this end, a novel clustering technique for multivariate functional data is presented. This method is based on a functional latent mixture model which fits the data into group-specific functional subspaces through a multivariate functional principal component analysis. A family of parsimonious models is obtained by constraining model parameters within and between groups. An Expectation Maximization algorithm is proposed for model inference and the choice of hyper-parameters is addressed through model selection. Numerical experiments on simulated datasets highlight the good performance of the proposed methodology compared to existing works. This algorithm is then applied to the analysis of the pollution in French cities for one year.

Keywords Multivariate functional curves · multivariate functional principal component analysis · model-based clustering · EM algorithm

We would like to thank the LabCom 'CWD-VetLab' for its financial support. The LabCom 'CWD-VetLab' is financially supported by the Agence Nationale de la Recherche (contract ANR 16-LCV2-0002-01)

Amandine Schmutz · Pauline Martin
Lim France, Chemin Fontaine de Fanny, Nontron, France & CWD-VetLab, Ecole Nationale Vétérinaire d'Alfort, Maisons-Alfort, F-94700, France. Tel.: +336-85861287,
Orcid: 0000-0003-2523-0411 / 0000-0002-3571-4244, E-mail: aschmutz@lim-group.com

Julien Jacques
Université de Lyon, Lyon 2, ERIC EA3083, Lyon, France. Orcid: 0000-0003-4808-2781

Charles Bouveyron
Université Côte d'Azur, Inria, CNRS, LJAD, Maasai team, France. Orcid: 0000-0002-6956-4491

Laurence Chèze
Université de Lyon, Lyon 1, LBMC UMR T9406, Lyon, France. Orcid: 0000-0003-2265-9781

1 Introduction

The modern technologies ease the collection of high frequency data which is of interest to model and understand the studied phenomenon for further analyses. For example in sports, athletes wear devices that collect data during their training to improve their performances and follow their physical constants in order to prevent injuries. This kind of data can be classified as functional data: a quantitative entity evolving along time. For instance in the univariate case, a functional data X is represented by a single curve, $X(t) \in \mathbb{R}$, $\forall t \in [0, T]$. With the growth of smart device market, more and more data are collected for the same individual, such as runner heartbeat and the altitude of his travel. An individual is then represented by several curves. The corresponding multivariate functional data can be written: $\mathbf{X} = \mathbf{X}(t)_{t\in[0,T]}$ with $\mathbf{X}(t) = (X^1(t), ..., X^p(t))' \in \mathbb{R}^p, p \geq 2$. We refer to Ramsay and Silverman (2005) for univariate and bivariate examples.
Because of this amount of collected data, there is an increasing need for methods able to identify homogeneous subgroups of data, to make better individualized predictions for example. The clustering of functional data can be addressed with different methods, that can be split into 4 categories according to Jacques and Preda (2014a): the raw data methods that consists of clustering directly the curves on their finite set of points ; the filtering methods that need a first step of smoothing curves into a basis of functions and a second step of clustering the obtained expansion coefficients ; the adaptive methods where clustering and expression of the curves into a finite dimensional space are performed simultaneously ; and the distance-based methods where usual clustering algorithms are applied with specific distances for functional data. Among these categories, there exist numerous works for the clustering of univariate functional data as for instance James and Sugar (2003), Tarpey and Kinateder (2003), Chiou and Li (2007), Bouveyron and Jacques (2011), Jacques and Preda (2013), Bouveyron et al. (2015) and Bongiorno and Goia (2016).

Conversely, only a few exists for clustering multivariate functional data. Singhal and Seborg (2005) and Ieva et al. (2013) use a k-means algorithm based on specific distances between multivariate functional data. Kayano et al. (2010) consider Self-Organizing Maps based on the coefficients of multivariate curves into an orthonormalized Gaussian basis expansions. Tokushige et al. (2007) extend crisp and fuzzy k-means algorithms for multivariate functional data by considering a specific distance between functions. Those methods cluster data by considering that they lie in the same subspace. A new method has been recently published based on a hypothesis testing k-means (Dias et al., 2018). At each step of the k-means algorithm, the curve belonging decision is based on the combination of two hypothesis test statistics. The performance of their algorithm is compared to distance-based methods and some dimension reduction based methods. Those dimension reduction techniques main principle is to obtain a low-dimensional representation of functions. For example, Ieva and Paganoni (2013) present a generalized functional linear regression

model that cluster individuals in two categories. The first step consist of a multivariate functional principal component analysis applied on the variance-covariance matrix on the functional data and their first derivatives. Then, the obtained scores are used as covariates in a generalized linear model to predict the outcome.Yamamoto and Hwang (2017) propose a clustering method that combines a subspace separation technique with functional subspace clustering, named FGRC, that is less sensible to data variance than functional principal component k-means developed by Yamamoto (2012) and functional factorial k-means (Yamamoto and Terada (2014)). Finally, Jacques and Preda (2014b) present a Gaussian model-based clustering method based on a principal component analysis for multivariate functional data (MFPCA). One of the benefits of this method is that the dependency between functional variables is managed thanks to the MFPCA. More recently, new methods based on a mix between dimension reduction and nonparametric approaches appear. Indeed, Traore et al. (2019) propose a clustering technique for nuclear safety experiment where one individual curve is decomposed into two new curves that are used in the decision making process. The first step consists in doing a dimension reduction technique on the first curves and applying a hierarchical clustering on those obtained values. Then, a semi-metric is build to compare the second curves, and the clusters are refining thanks to this comparison. But, even if this method is developped to deal with two curves for a same individual, at first the functional data are univariate.

In Jacques and Preda (2014b), MFPCA scores are considered as random variables whose probability distributions are cluster specific. Although this model is far more flexible than other methods due to its probabilistic modeling, it suffers nevertheless from some limitations. Indeed, using an approximation of the notion of density distribution for functional data, the authors modeled only a given proportion of principal components and thus a significant part of the available information is ignored. In this paper, we propose a model which extends Jacques and Preda (2014b) work by modeling all principal components whose estimated variance are non-null. All available information is therefore taken into account. This is a significant advantage because it will give a finner modeling and, consequently, a better clustering in most cases. Moreover, our model allows to use an Expectation Maximisation (EM) algorithm for its inference, with the theoretical guaranties it implies, whereas Jacques and Preda (2014b) use an heuristic pseudo-EM algorithm with no theoretical guaranties. The resulting model can be also viewed as an extension of Bouveyron and Jacques (2011) method to the multivariate case, that is why we will refer to it as the funHDDC model in the following.

The paper is organized as follows. A quick reminder of function data analysis is done in Section 2 . Section 3 presents principal component analysis for multivariate functional data, as introduced in Jacques and Preda (2014b). Section 4 introduces the mixture model allowing the clustering of multivariate functional data. Section 5 discusses parameters estimation via an EM algorithm, proposes criteria for the selection of number of clusters and computational details. Comparisons between the proposed method and existing ones

on simulated and real datasets are presented in Sections 6 and 7. A discussion concludes the paper in Section 8.

2 Functional data analysis

Let us first assume that the observed curves $\mathbf{X}_1, ..., \mathbf{X}_n$ are independent realizations of a L_2-continuous multivariate stochastic process $\mathbf{X} = \{\mathbf{X}(t), t \in [0,T]\} = \{(X^1(t), \ldots, X^p(t))\}_{t\in[0,T]}$ for which the sample paths, i.e. the observed curves $\mathbf{X}_i = (X_i^1, ..., X_i^p)$, belong to $L_2[0,T]$. Without loss of generality, let assume that $E(\boldsymbol{X}) = 0$.

In practice, the functional expressions of the observed curves are not known and it is only possible to have access to discrete observations at a finite set of times $X_i^j(t_1), \ldots, X_i^j(t_s)$ with $0 \leq t_1 \leq \ldots, \leq t_s \leq 1$ for every $1 \leq i \leq n$ and $1 \leq j \leq p$. The first task, when working with functional data, is therefore to convert these discretely observed values to a function $X_i^j(t)$, computable for any desired argument value $t \in [0,T]$. One way to do that is interpolation, which is used if the observed values are assumed to be errorless. However, if there is some noise that needs to be removed, a common way to reconstruct the functional form is to assume that the curves $X_i^j(t)$ can be decomposed into a finite dimensional space, spanned by a basis of functions (Ramsay and Silverman, 2005):

$$X_i^j(t) = \sum_{r=1}^{R_j} c_{ir}^j (X_i^j) \phi_r^j(t) \tag{1}$$

where $(\phi_r^j(t))_{1\leq r\leq R_j}$ is the basis of functions for the j-th component of the multivariate curve and R_j the number of basis functions. In order to ease the description of the model, let us introduce the following notations. The coefficients c_{ir}^j can be gathered in the matrix $\boldsymbol{C}$:

$$\mathbf{C} = \begin{pmatrix} c_{11}^1 \ ... \ c_{1R_1}^1 & c_{11}^2 \ ... \ c_{1R_2}^2 & ... & c_{11}^p \ ... \ c_{1R_p}^p \\ & ... & & \\ c_{n1}^1 \ ... \ c_{nR_1}^1 & c_{n1}^2 \ ... \ c_{nR_2}^2 & ... & c_{n1}^p \ ... \ c_{nR_p}^p \end{pmatrix}.$$

Let also introduce the matrix $\boldsymbol{\phi(t)}$, gathering the basis functions:

$$\boldsymbol{\phi(t)} = \begin{pmatrix} \phi_1^1(t) \ ... \ \phi_{R_1}^1(t) & 0 \ ... \ 0 & ... & 0 \ ... \ 0 \\ 0 \ ... \ 0 & \phi_1^2(t) \ ... \ \phi_{R_2}^2(t) & ... & 0 \ ... \ 0 \\ & ... & & \\ 0 \ ... \ 0 & 0 \ ... \ 0 & ... & \phi_1^p(t) \ ... \ \phi_{R_p}^p(t) \end{pmatrix}.$$

With these notations, Equation (1) can be rewritten as follows:

$$\boldsymbol{X}(t) = \boldsymbol{C}\boldsymbol{\phi}'(t).$$

The estimation of $\boldsymbol{C}$ is usually done through least square smoothing (see Ramsay and Silverman (2005)). The choice of the basis functions, contained in $\boldsymbol{\phi}$, has to be made by the user. There is no straight rules about how to choose the

appropriate ones (Jacques and Preda (2014a)). We can nevertheless recommend the use of a Fourier basis in the case of data with a repetitive pattern, and B-spline functions in most other cases.

3 Multivariate functional principal component analysis

Principal component analysis for multivariate functional data has already been suggested by various authors. Ramsay and Silverman (2005) propose to concatenate observations of the functions measured on a fine grid of points into a single vector and then to perform a standard principal component analysis (PCA) on these concatenated vectors. They also propose to express observations into a known basis of functions and apply PCA on the vector of concatenated coefficients. Both approaches may be problematic when the functions correspond to different observed phenomena. Moreover, the interpretation of multivariate scores for one individual is usually difficult. In Berrendero et al. (2011), the authors propose instead to summarize the curves with functional principal components. For this purpose, they carry out classical PCA for each value of the domain on which the functions are observed and suggest an interpolation method to build their principal functional components. In a different approach, Jacques and Preda (2014b) suggest the MFPCA method, with a normalization step if the units of measurement differ between functional variables. Their method relies on the multidimensional version of the Karhunen-Loève expansion (Saporta, 1981). Chiou et al. (2014) also present a normalized multivariate functional principal component analysis which takes into account the differences in degrees of variability and units of measurement among the components of the multivariate random functions. As in Jacques and Preda (2014b), it leads to a single set of scores for each individual. Chen and Jiang (2016) present a multi-dimensional functional principal component analysis and Happ and Greven (2015) a multivariate functional principal component analysis that both can handle data observed on more than one-dimensional domain. Happ and Greven (2015) method can be applied to sparse functional data and includes the MFPCA proposed by Jacques and Preda (2014b) when the interval is $[0, T]$ and steady.

Because our data are collected on the one-dimensional interval [0,T] and with a regular sampling scheme, the MFPCA proposed by Jacques and Preda (2014b) is used in combination with a fine probabilistic modeling of the group-specific densities. The MFPCA method is therefore summarized hereafter. MFPCA aims at finding the eigenvalues and eigenfunctions that solve the spectral decomposition of the covariance operator ν:

$$\nu \boldsymbol{f}_l = \lambda_l \boldsymbol{f}_l, \forall l \geq 1, \tag{2}$$

with λ_l a set of positive eigenvalues and $\boldsymbol{f}_l$ the set of associated multivariate eigenfunctions. The estimator of the covariance operator can be written as:

$$\widehat{\nu}(s,t) = \frac{1}{n-1}\boldsymbol{X}'(s)\boldsymbol{X}(t) = \frac{1}{n-1}\boldsymbol{\phi}(s)\boldsymbol{C}'\boldsymbol{C}\boldsymbol{\phi}'(t). \tag{3}$$

Let suppose that each principal factor $\boldsymbol{f}_l$ belongs to the linear space spanned by the matrix $\boldsymbol{\phi}$:

$$\boldsymbol{f}_l(t) = \boldsymbol{\phi}(t)\boldsymbol{b}_l' \tag{4}$$

with $\boldsymbol{b}_l = (b_{l11}, ..., b_{l1R_1}, b_{l21}, ..., b_{l2R_2}, ..., b_{lp1}, ..., b_{lpR_p})$. Using Equation (3), the eigen problem (2) becomes:

$$\frac{1}{n-1}\boldsymbol{\phi}(s)\boldsymbol{C}'\boldsymbol{C}\boldsymbol{W}\boldsymbol{b}_l' = \lambda_l\boldsymbol{\phi}(s)\boldsymbol{b}_l' \tag{5}$$

where $\boldsymbol{W} = \int_0^T \boldsymbol{\phi}'(t)\boldsymbol{\phi}(t)$ is a $R \times R$ matrix, where $R = \sum_{j=1}^p R_j$, which contains the inner products between the basis functions. The MFPCA then reduces to eigendecomposition of the matrix $\frac{1}{\sqrt{n-1}}\boldsymbol{C}\boldsymbol{W}^{1/2}$. Thus, each multivariate curve $\boldsymbol{X}_i$ is identified by its score $\boldsymbol{\delta}_i = (\delta_{il})_{l\geq 1}$ into the basis of multivariate eigenfunctions $(\boldsymbol{f}_l)_{l\geq 1}$. Scores are obtained from $\delta_{il} = \boldsymbol{C}_i\boldsymbol{W}\boldsymbol{b'}_l$ where $\boldsymbol{C}_i$ is the i-th row of matrix $\boldsymbol{C}$.

In practice, due to the fact that each component X_i^j of $\boldsymbol{X}_i$ is approximated into a finite basis of functions of size R_j, the maximum number of scores which can be computed is $R = \sum_{j=1}^p R_j$.

4 A generative model for multivariate functional data clustering

Our goal is to group the observed multivariate curves $\boldsymbol{X}_1, ..., \boldsymbol{X}_n$ into K homogeneous clusters. At this stage, K is fixed a priori and an estimation procedure for this parameter will be suggested in Section 5.3. Let Z_{ik} be the latent variable such that $Z_{ik} = 1$ if $\boldsymbol{X}_i$ belongs to cluster k and 0 otherwise. In order to ease the presentation of the modeling, let us assume at first that the values z_{ik} of Z_{ik} are known for all $1 \leq i \leq n$ and $1 \leq k \leq K$ (our goal is in practice to recover them from the data). Let $n_k = \sum_{i=1}^n z_{ik}$ be the number of curves within cluster k.

Let suppose that the curves of each cluster can be described into a low-dimensional functional latent subspace specific to each cluster, with intrinsic dimensions $d_k < R$, $k = 1, ..., K$. Curves can be expressed into a group-specific basis $\{\varphi_r^k\}_{1\leq r\leq d_k}$, which is determined thanks to the model, and is obtained from $\{\phi_r^j\}_{1\leq j\leq p, 1\leq r\leq R}$ through a linear transformation:

$$\varphi_r^k(t) = \sum_{\ell=1}^{R} q_{kr\ell}\phi_\ell(t), 1 \leq r \leq R$$

where $Q_k = (q_{kr\ell})_{1\leq r,\ell\leq R}$ is the orthogonal $R \times R$ matrix containing the basis expansion coefficients of the eigenfunctions. Q_k is split for later use into two parts: $Q_k = [U_k, V_k]$ with U_k of size $R \times d_k$ and V_k of size $R \times (R - d_k)$, $U_k'U_k = I_{d_k}$, $V_k'V_k = I_{R-d_k}$ and $U_k'V_k = 0$.

Let $(\boldsymbol{\delta}_i^k)_{1\leq i\leq n_k}$ be the MFPCA scores of the n_k curves of cluster k. These scores are assumed to follow a Gaussian distribution

$$\boldsymbol{\delta}_i^k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Delta}_k)$$

with $\boldsymbol{\mu}_k \in \mathbb{R}^R$ the mean function and $\boldsymbol{\Delta}_k$ the covariance matrix with the following form:

$$\boldsymbol{\Delta}_k = \begin{pmatrix} \boxed{\begin{matrix} a_{k1} & & 0 \\ & \ddots & \\ 0 & & a_{kd_k} \end{matrix}} & \mathbf{0} \\ \mathbf{0} & \boxed{\begin{matrix} b_k & & 0 \\ & \ddots & \\ & & b_k \end{matrix}} \end{pmatrix} \begin{matrix} \} \, d_k \\ \} \, R - d_k \end{matrix}$$

The assumption on $\boldsymbol{\Delta}_k$ allows to finely model the variance of the first d_k principal components, while the remaining ones are considered as noise components and modeled by a unique parameter b_k. This model will be referred to as $[a_{kj}b_kQ_kd_k]$ hereafter. The model of Jacques and Preda (2014b) is similar but with the constraint $b_k = 0$, for all $k = 1, ..., K$. The latter leads to ignore information contained in the last eigenfunctions, whereas we propose to model it in a parsimonious way.

In addition, different sub-models can be defined depending on the constraints we apply on model parameters, within or between groups, leading to more parsimonious models. This possibility allows to fit into various situations. The following 5 sub-models can be derived from the most general one:

- $[a_kb_kQ_kd_k]$: this model is used if the first d_k eigenvalues are assumed to be common within each group. In this case, there is only 2 eigenvalues in $\boldsymbol{\Delta}_k$, a_k and b_k.
- $[a_{kj}bQ_kd_k]$: the parameters b_k are fixed to be common between groups. It assumes that the variance outside the group-specific subspaces is common, a usual hypothesis when data are obtained in a common acquisition process.
- $[a_kbQ_kd_k]$: the parameters a_k are fixed to be common within each group and b_k are fixed to be common between groups.
- $[ab_kQ_kd_k]$: the parameters a_{kj} are fixed to be common between and within groups.
- $[abQ_kd_k]$: the parameters a_{kj} and b_k are fixed to be common between and within groups.

In practice, the z_{ik}'s are not known and our goal is to predict them. That is why an EM algorithm is proposed below in order to estimate model parameters and then to predict the z_{ik}'s.

5 Model inference and choice of the number of clusters

5.1 Model inference through an EM algorithm

In model-based clustering, the estimation of model parameters is traditionally done by maximizing the likelihood through the EM algorithm (Dempster et al.,

1977). The EM algorithm alternates between two steps: the expectation (E) and maximization (M) steps. The E step aims at computing the conditional expectation of the complete log-likelihood using the current estimate of parameters. Then, the M step computes parameter estimates maximizing the expected complete log-likelihood found in the E step.

This section presents the update formulae of the EM algorithm in the case of the $[a_{kj}b_kQ_kd_k]$ model. Update formulae can be easily derived in the same manner for other models. The following proposition provides the expression of the complete log-likelihood associated with the model described above. Proof of this result is provided in Appendix A.1.

Proposition 1 *The complete log-likelihood of the observed curves under the* $[a_{kj}b_kQ_kd_k]$ *model can be written as:*

$$\begin{aligned}\ell_c(\theta) = -\frac{1}{2}\sum_{k=1}^{K} n_k \Bigg[&\sum_{j=1}^{d_k}\left(\log(a_{kj}) + \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{a_{kj}}\right) \\ &+ \sum_{j=d_k+1}^{R}\left(\log(b_k) + \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{b_k}\right) - 2\log(\pi_k)\Bigg] \\ &+ \frac{nR}{2} log(2\pi), \end{aligned} \tag{6}$$

where $\theta = (\pi_k, \boldsymbol{\mu}_k, a_{kj}, b_k, q_{kj})_{kj}$ *for* $1 \leq k \leq K$ *and* $1 \leq j \leq d_k$, q_{kj} *is the* j*-th column of* Q_k, $C_k = \frac{1}{n_k}\sum_{i=1}^{n} Z_{ik}(c_i - \boldsymbol{\mu}_k)^t(c_i - \boldsymbol{\mu}_k)$ *and* $c_i = (c_{ir}^1, ..., c_{ir}^p)$ *is a vector of coefficients.*

As the group memberships Z_{ik} are unknown, the EM algorithm starts by computing their conditional expectation (*E step*) before maximizing the expected complete likelihood (*M step*).

E step This step aims at computing the conditional expectation of the complete log-likelihood and reduces to the computing of the conditional expectation $E[Z_{ik}|c_i, \theta^{(q-1)}]$, which can be computed as follows.

Proposition 2 *For the model* $[a_{kj}b_kQ_kd_k]$*, the posterior probability that each curve belongs to the k-th cluster can be written:*

$$t_{ik}^{(q)} = E[Z_{ik}|c_i, \theta^{(q-1)}] = 1/\sum_{l=1}^{K} \exp[\frac{1}{2}(H_k^{(q-1)}(c_i) - H_l^{(q-1)}(c_i))], \tag{7}$$

where $H_k^{(q-1)}(c)$ *is the cost function defined for* $c \in \mathbb{R}^R$ *as:*

$$\begin{aligned} H_k^{(q-1)}(c) = ||\boldsymbol{\mu}_k^{(q-1)} - P_k(c)||_{D_k}^2 + \frac{1}{b_k^{(q-1)}}||c - P_k(c)||^2 \\ + \sum_{j=1}^{d_k}\log(a_{kj}^{(q-1)}) + (R - d_k)\log(b_k^{(q-1)}) - 2\log(\pi_k^{(q-1)}), \end{aligned} \tag{8}$$

where $||.||^2_{\mathcal{D}_k}$ *is a norm on the latent space* $\mathbb{E}_k$ *defined by* $||y||^2_{\mathcal{D}_k} = y^t\mathcal{D}_k y$, $\mathcal{D}_k = \tilde{Q}\boldsymbol{\Delta}_k^{-1}\tilde{Q}^t$ *and* $\tilde{Q}$ *is a matrix containing the* d_k *vectors of* U_k *completed by zeros such as* $\tilde{Q} = [U_k, 0_{R-d_k}]$, P_k *is the projection operator on the functional latent space* $\mathbb{E}_k$ *defined by* $P_k(c) = \boldsymbol{W}U_kU_k^t\boldsymbol{W}^t(c - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_k$.

Proof of this result is provided in Appendix A.2.

M step This step estimates the model parameters by maximizing the expectation of the complete log-likelihood conditionally on the posterior probabilities $t_{ik}^{(q)}$ computed in the previous step. The following proposition provides update formulae for mixture parameters. Proof of these results are provided in Appendix A.3.

Proposition 3 *For the model* $[a_{kj}b_kQ_kd_k]$, *the maximization of the conditional expected complete log-likelihood leads to the following update:*

- $\pi_k^{(q)} = \frac{\eta_k^{(q)}}{n}$, $\mu_k^{(q)} = \frac{1}{\eta_k^{(q)}}\sum_{i=1}^n t_{ik}^{(q)}c_i$,
- *the* d_k *first columns of the orientation matrix* Q_k *are updated by the eigenfunctions coefficients associated with the largest eigenvalues of* $\boldsymbol{W}^{1/2}C_k^{(q)}\boldsymbol{W}^{1/2}$,
- *the variance parameters* a_{kj}, $j = 1, ..., d_k$, *are updated by the* d_k *largest eigenvalues of* $\boldsymbol{W}^{1/2}C_k^{(q)}\boldsymbol{W}^{1/2}$,
- *the variance parameters* b_k *are updated by* $b_k^{(q)} = \frac{1}{R-d_k}[tr(\boldsymbol{W}^{1/2}C_k^{(q)}\boldsymbol{W}^{1/2}) - \sum_{j=1}^{d_k}\widehat{a}_{kj}^{(q)}]$.

where $\eta_k^{(q)} = \sum_{i=1}^n t_{ik}^{(q)}$ *and* $C_k^{(q)} = \frac{1}{\eta_k^{(q)}}\sum_{i=1}^n t_{ik}^{(q)}(c_i - \mu_k^{(q)})^t(c_i - \mu_k^{(q)})$ *is the sample covariance matrix of group* k.

To summarize, the algorithm introduced above, named hereafter funHDDC, clusters multivariate functional data through their projection into low dimensional subspaces. Those projections are obtained by performing a MFPCA per cluster thank to an iterative algorithm.

5.2 Estimation of intrinsic dimensions

In order to choose the intrinsic dimensions d_k of each cluster, the Cattell's scree-test (Cattell, 1966) is used. This test looks for a drop in the eigenvalues scree. The selected dimension is the one for which the subsequent eigenvalues differences are smaller than a threshold provided by the user or selected using Bayesian information criterion, Akaike information criterion, integrated completed likelihood or slope heuristic (described below).

This estimation of the number of intrinsic dimensions is done within the M step of EM algorithm. It may allow the estimated intrinsic dimensions to vary along the iterations in order to fit well data.

5.3 Choice of the number of clusters

We now focus on the choice of the hyper-parameter K, the number of clusters. The choice of this hyper-parameter is here viewed as model selection problem. Classical model selection tools include the Akaike information criterion (AIC, Akaike (1974)), the Bayesian information criterion (BIC, Schwarz (1978)) and the integrated completed likelihood criterion (ICL, Biernacki et al. (2000)). In the context of mixture model, BIC is certainly the most popular. The BIC criterion can be computed as follows:

$$BIC = l(\hat{\theta}) - \tfrac{m}{2} \times log(n),$$

with $l(\hat{\theta})$ the maximum log-likelihood value, m the number of model parameters and n the number of individuals. The criterion penalizes the log-likelihood through model complexity. The model maximizing the criterion is chosen.

Another criterion, that has proved its usefulness, is the slope heuristic (SH, Birge and Massart (2007)). This data-driven criterion penalty has a multiplicative factor provided by the linear part of the log-likelihood:

$$SH = l(\hat{\theta}) - 2\, s\, m,$$

where s is the slope of the linear part of the maximum log-likelihood value $l(\hat{\theta})$ when plotted against the model complexity. It has to be noticed that this method requires to test a large number of clusters number, or a large number of models, so that there is enough points in the log-likelihood versus model complexity plot (bottom left plot of Fig. 4) to detect a plateau in the log-likelihood.

For either of those criteria, different values for K need to be tested. Then, the one that maximizes the chosen criterion's value is the best one that has to be kept.

5.4 Computational details

As explained in section 4.1, funHDDC algorithm relies on an EM algorithm. The EM algorithm needs to be initialized, by setting initial values for the partitions. To this end, two initialization strategies are considered: random and kmeans initializations. In the case of random initialization, the partitions are randomly sampled using a multinomial distribution with uniform probabilities. The kmeans strategy consists in initializing the partitions with those obtained by a kmeans algorithm applied directly on the whole set of discretized observations. With kmeans initialization, the EM algorithm usually converges quicker than with random initialization. For both initialization, it is highly recommended, in order to prevent the convergence to a local maximum, to perform multiple initializations of the algorithm and keep the solution maximizing the log-likelihood. The number of initializations is a parameter of funHDDC algorithm that can be tuned by the user.

The funHDDC algorithm is stopped when the difference between two consecutive log-likelihood values is lower than a given threshold ϵ or after a maximal number of iterations.

Running times for different sizes of datasets will be presented later, in Section 6.4.

6 Numerical experimentation on simulated data

This section presents numerical experiments to illustrate the behavior of the proposed methodology and confront it to competitors from the literature. Firstly, the quality of the model inference algorithm is illustrated on simulated data. Secondly, the sensitivity of the proposed approach to sample size is investigated in term of correct classification rate as well as in term of computational time. Thirdly, BIC and SH are compared for selecting the number of clusters. Finally, funHDDC is confronted to competitors on several datasets. The R code (R Core Team, 2017) for our multivariate functional clustering algorithm is available on CRAN in the funHDDC package.

6.1 Simulation setup

In order to ease the reproducibility of the results, we consider 3 simulation scenarios designed as follows.

Scenario A: For this first scenario, a sample of 1,000 bivariate curves are simulated based on $[a_k b_k Q_k D_k]$ model. To do so, scores are simulated according to a Gaussian model with mean μ and diagonal variance Δ. Curves coefficients can be rebuild based on $(\delta_{il})_{l\geq 1} = \boldsymbol{CWb'}_l$ as shown in Section 2. The number of clusters is fixed to $K = 3$ and mixing proportions are equal. Scores are generated from a multivariate normal distribution with the following parameters:

$$\begin{aligned}
&\text{Group } 1 : d = 5, a = 150, b = 5, \mu = (1, 0, 50, 100, 0, \ldots, 0),\\
&\text{Group } 2 : d = 20, a = 15, b = 8, \mu = (0, 0, 80, 0, 40, 2, 0, \ldots, 0),\\
&\text{Group } 3 : d = 10, a = 30, b = 10, \mu = (0, \ldots, 0, 20, 0, 80, 0, 0, 100),
\end{aligned}$$

where d is the intrinsic dimension of subgroups, μ is the mean vector of size 70, a is the value of the d-first diagonal elements of Δ and b the value of the (70-d)-last ones. Curves are smoothed using a basis of 35 Fourier functions (cf. top panel of Figure 1).

Scenario B: The second simulation setting is inspired by the data simulation process of Ferraty and Vieu (2003); Preda (2007); Bouveyron et al. (2015), and therefore will not favor our approach in the comparison. For this simulation,

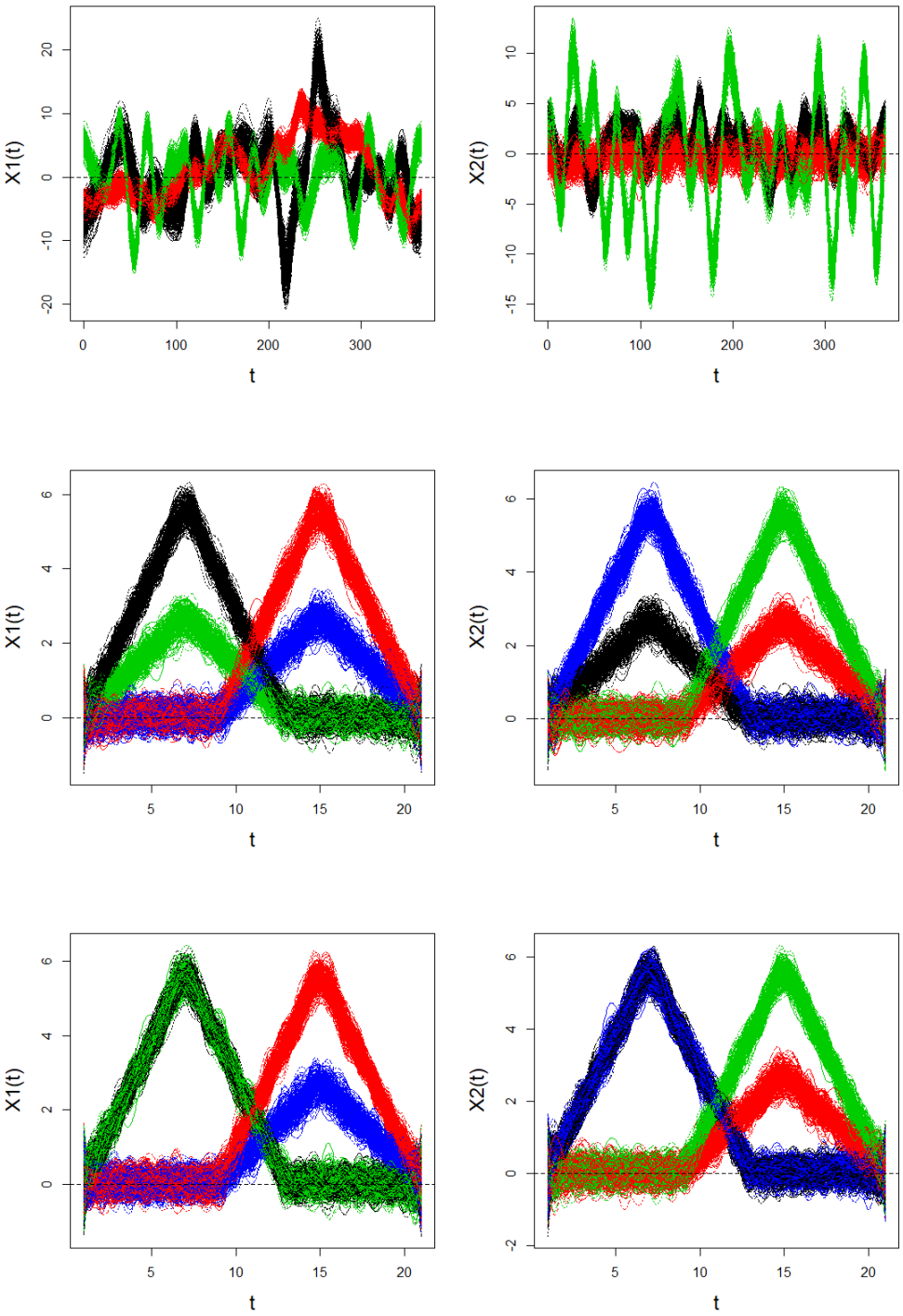


Fig. 1 Smooth data simulated for variable 1 (left) and variable 2 (right) for scenario A (top), scenario B (middle) and scenario C (bottom) colored by group for one simulation

the number of clusters is fixed to $K = 4$. A sample of 1000 bivariate curves is simulated according to the following model for $t \in [1, 21]$:

$$\begin{aligned}
\text{Group 1}: \quad & X_1(t) = U + (1-U)h_1(t) + \epsilon(t), \\
& X_2(t) = U + (0.5-U)h_1(t) + \epsilon(t), \\
\text{Group 2}: \quad & X_1(t) = U + (1-U)h_2(t) + \epsilon(t), \\
& X_2(t) = U + (0.5-U)h_2(t) + \epsilon(t), \\
\text{Group 3}: \quad & X_1(t) = U + (0.5-U)h_1(t) + \epsilon(t), \\
& X_2(t) = V + (1-V)h_2(t) + \epsilon(t), \\
\text{Group 4}: \quad & X_1(t) = U + (0.5-U)h_2(t) + \epsilon(t), \\
& X_2(t) = U + (1-U)h_1(t) + \epsilon(t),
\end{aligned}$$

where $U \sim \mathcal{U}(0, 0.1)$ and $\epsilon(t)$ is a white noise independent of U and such that $\mathbb{V}ar(\epsilon(t)) = 0.25$. The functions h_1 and h_2 are defined, for $t \in [1, 21]$, by $h_1(t) = (6 - |t-7|)_+$ and $h_2(t) = (6 - |t-15|)_+$ where $(\cdot)_+$ means the positive part. The mixing proportions are equal, and the curves are observed in 101 equidistant points. The functional form of the data is reconstructed using a cubic B-spline basis smoothing with 25 basis functions (cf. middle panel of Figure 1).

Scenario C: For this third simulation scenario, the number of clusters is fixed to $K = 4$. A sample of 1000 bivariate curves is simulated according to the following model for $t \in [1, 21]$:

$$\begin{aligned}
\text{Group 1}: \quad & X_1(t) = U + (1-U)h_1(t) + \epsilon(t), \\
& X_2(t) = U + (0.5-U)h_1(t) + \epsilon(t), \\
\text{Group 2}: \quad & X_1(t) = U + (1-U)h_2(t) + \epsilon(t), \\
& X_2(t) = U + (0.5-U)h_2(t) + \epsilon(t), \\
\text{Group 3}: \quad & X_1(t) = U + (1-U)h_1(t) + \epsilon(t), \\
& X_2(t) = U + (1-U)h_2(t) + \epsilon(t), \\
\text{Group 4}: \quad & X_1(t) = U + (0.5-U)h_2(t) + \epsilon(t), \\
& X_2(t) = U + (0.5-U)h_1(t) + \epsilon(t),
\end{aligned}$$

where $U, \epsilon(t), h_1$ and h_2 are defined as before. The mixing proportions are equal, and the curves are observed in 101 equidistant points. The functional form of the data is reconstructed using a cubic B-splines basis smoothing with 25 basis functions. As shown in Figure 1 (bottom), the 4 groups cannot be distinguished with one variable only: indeed group 3 (green) is similar to group 1 (black) for variable $X_1(t)$ and similarly group 4 (blue) is similar to group 1 (black) for variable $X_2(t)$. Consequently, any univariate functional clustering methods applied either on variable $X_1(t)$ or $X_2(t)$ should fail.

For each scenario, the estimated partitions are compared to the true partition with the adjusted Rand index (ARI, Rand (1971)). This criterion value is less than or equal to 1, with 1 representing a perfect agreement between the true partition and the one estimated by the algorithm, and 0 a random agreement. The algorithm settings used for all simulations are the following: the threshold of the Cattell's scree-test for the selection of intrinsic dimensions d_k is fixed to 0.2 (the optimal threshold value should be chosen using BIC or slope heuristic), the stopping criterion for the EM algorithm is a growth of the log-likelihood lower than $\epsilon = 10^{-3}$ or a maximal number of iterations of 200, the initialization of the algorithm is done through a *random* partition in the introductory example, and with the *kmeans* strategy the model selection and benchmark experiments, in order to speed up the convergence.

6.2 An introductory example

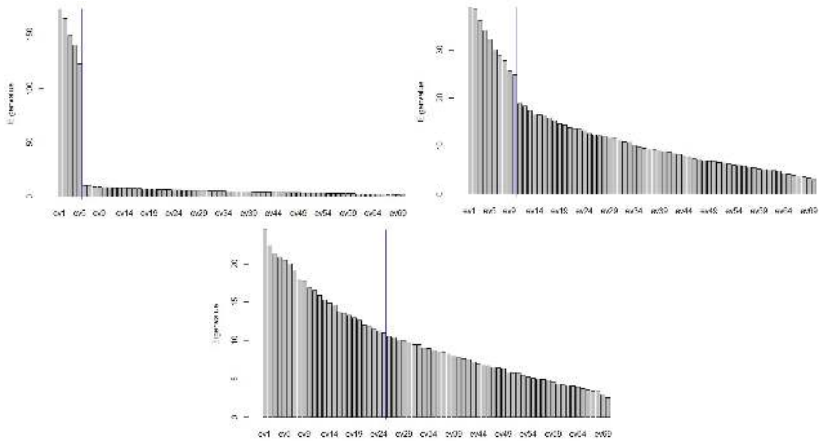
Table 1 Mean (and s.d.) of ARI for 50 simulations

funHDDC model	Mean (SD)
$[a_{kj}b_kQ_kD_k]$	0.99 (0.08)
$[a_{kj}bQ_kD_k]$	0.85 (0.26)
$[a_kb_kQ_kD_k]$	**1 (0)**
$[a_kbQ_kD_k]$	0.88 (0.26)
$[ab_kQ_kD_k]$	0.95 (0.16)
$[abQ_kD_k]$	0.49 (0.36)

In order to illustrate the good behavior of the inference algorithm, we first consider a single simulation according to *Scenario A*, which is overall a difficult situation. The algorithm is run for $K = 3$ groups with the model $[a_kb_kQ_kD_k]$ which has been used to generate the data and the simulation setting is repeated 50 times. Figure 2 allows the comparison of the fitted values with the actual ones of model parameters. Parameter a turns out to be well estimated for all 3 clusters, whereas parameters b and d are only well estimated for 2 clusters out of 3. Indeed, this simulation scenario has one mixture component with a low signal-to-noise ratio which disturbs the estimation of d_k for this component, and therefore also perturb the estimation of the noise variance b_k. Nevertheless, the fact that our model actually models the variance within the whole space (and not only a part as in Jacques and Preda (2014b)), allows us to correctly recover the cluster partition even in difficult estimation conditions.

To assess the clustering quality, the funHDDC algorithm is now run for $K = 3$ groups with all 6 sub-models and the simulation setting is repeated 50 times. The quality of the estimated partitions is evaluated using the ARI and results are given in Table 1. As expected, the best result is obtained for the model $[a_kb_kQ_kD_k]$ which has been used to generate data and it shows that

the algorithm correctly recovers the cluster pattern. It is worth noticing that the other models also have satisfying performances.



Parameters	Group 1	Group 2	Group 3
a	150.8 (5.1)	12.6 (2.9)	31.4 (2.4)
b	4.9 (0.1)	4.3 (1.7)	9.3 (0.4)
d	5 (0)	50.2 (20.1)	11.3 (3.0)

Fig. 2 Scree-test of Cattell performed for each group with the threshold set to 0.2 (blue line) for one simulation and mean (sd) of parameters estimation for the 50 simulations with the $[a_k b_k Q_k D_k]$ model

6.3 Sample size influence

In order to evaluate the sensitivity of the proposed approach to the sample size, we now consider 50 simulations according to *Scenario B*, and with different sample size: 1000, 500, 200 and 30. Table 2 presents the corresponding results. The impact of the sample size is not really significant between 1000 to 200. For very small sample size, 30 observations for 3 clusters, the quality of the partition estimation significantly decreases, but such small sample size are seldom used in practice for clustering studies.

Table 2 Mean (and s.d.) of ARI for 50 simulations

	sample size n			
funHDDC model	1000	500	200	30
$[a_{kj}b_kQ_kD_k]$	0.98 (0.08)	0.99 (0.05)	0.98 (0.07)	0.84 (0.20)
$[a_{kj}bQ_kD_k]$	0.82 (0.19)	0.71 (0.15)	0.70 (0.12)	0.67 (0.09)
$[a_kb_kQ_kD_k]$	1 (0)	0.99 (0.07)	0.98 (0.08)	0.80 (0.23)
$[a_kbQ_kD_k]$	0.88 (0.18)	0.66 (0.10)	0.69 (0.11)	0.66 (0.08)
$[ab_kQ_kD_k]$	0.98 (0.09)	1 (0)	0.99 (0.05)	0.90 (0.16)
$[abQ_kD_k]$	0.86 (0.18)	0.71 (0.14)	0.68 (0.11)	0.66 (0.08)

6.4 Computational time and cost

When dealing with multivariate functional data, a big issue consists of scalability and computational effort in performing analyses. We will present here the impact of sample size and number of functional variables on running time.

Firstly, funHDDC algorithm is applied for $K = 4$ groups on *scenario B* bivariate functional data. Then, a second scenario with four functional variables is built based on *scenario B* with the two additional functional variables be cosinus and sinus functions. The computer used for the experiments has a Windows 10 operating system, Intel(R) Core(TM) i7-6700 CPU 3.40GHz processor and 8.00 Go of RAM memory. The associated running times, estimated with Sys.time R function, are shown in Table 3.

Table 3 Computational effort in performing analyses

Sample size	Number of functional variables	Running time (sec)
1000	2	0.24
10 000	2	1.16
100 000	2	10.71
1000	4	0.59
10 000	4	3.50
100 000	4	30.85

6.5 Model selection

In this section, the selection of the number of clusters is investigated. As previously mentioned, two criteria are used: BIC and the slope heuristic. Data are generated from *Scenario A*. This simulation setting has been repeated 50 times and the 6 sub-models have been estimated for a number of clusters from 2 to 10.

Figures 3 and 4 show for one simulation with the model $[a_kb_kQ_kD_k]$, the values of BIC and the slope heuristic in view of the number of clusters. For this simulation, both the slope heuristic and BIC succeed in selecting the right number of clusters. On Figure 4, the left plot corresponds to the log-likelihood

function with respect to the number of free model parameters. The red line is estimated using a robust linear regression and its slope coefficient is used to compute the penalized log-likelihood function shown on the right plot.

Table 4 summarized the results of the 50 simulations for the BIC and the slope heuristic. The BIC criterion has some difficulties to estimate the actual number of clusters K. Indeed, depending on the simulation, BIC selects between 2 to 3 clusters and succeed in 46% of simulations in the case of $[a_k b_k Q_k D_k]$ model. The slope heuristic is conversely more efficient to recover the actual number of groups, in about 66% of simulations in the case of $[a_k b_k Q_k D_k]$ model.

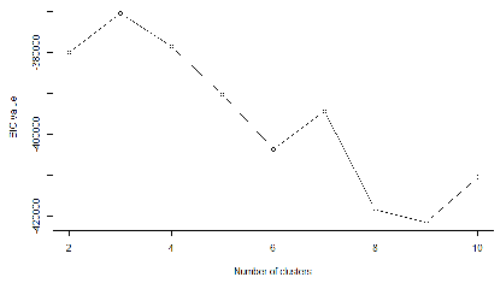


Fig. 3 BIC for one simulation for the model $[a_k b_k Q_k D_k]$

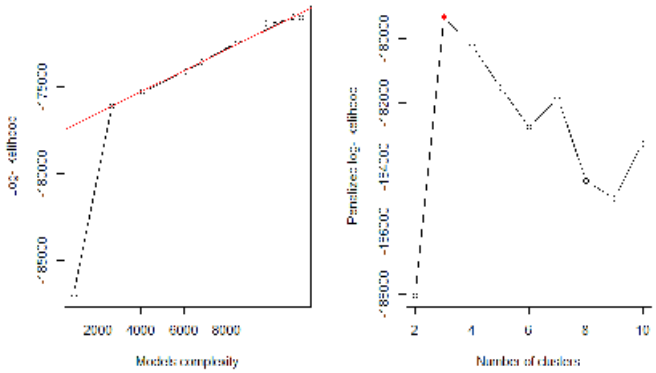


Fig. 4 Slope heuristic for one simulation for the model $[a_k b_k Q_k D_k]$

Table 4 Best model selected by BIC (top) and by the slope heuristic (SH, bottom) for 50 simulations as a percentage

BIC		Number K of clusters								
Method	Model	2	3	4	5	6	7	8	9	10
funHDDC	$[a_{kj}b_kQ_kD_k]$	36	**48**	10	6	-	-	-	-	-
funHDDC	$[a_{kj}bQ_kD_k]$	38	**54**	6	-	2	-	-	-	-
funHDDC	$[a_kb_kQ_kD_k]$	42	**46**	8	0	2	2	-	-	-
funHDDC	$[a_kbQ_kD_k]$	44	**48**	8	-	-	-	-	-	-
funHDDC	$[ab_kQ_kD_k]$	**46**	40	10	4	-	-	-	-	-
funHDDC	$[abQ_kD_k]$	**64**	24	10	2	-	-	-	-	-

SH		Number K of clusters								
Method	Model	2	3	4	5	6	7	8	9	10
funHDDC	$[a_{kj}b_kQ_kD_k]$	6	**60**	24	10	-	-	-	-	-
funHDDC	$[a_{kj}bQ_kD_k]$	10	**74**	12	4	-	-	-	-	-
funHDDC	$[a_kb_kQ_kD_k]$	18	**66**	14	2	-	-	-	-	-
funHDDC	$[a_kbQ_kD_k]$	26	**52**	14	8	2	-	-	-	-
funHDDC	$[ab_kQ_kD_k]$	34	**42**	16	6	2	-	-	-	-
funHDDC	$[abQ_kD_k]$	**38**	28	20	10	2	0	0	2	-

6.6 Benchmark with existing methods

In this section, the proposed clustering algorithm is compared to competitors of the literature: Funclust (from Funclustering package, Jacques and Preda (2014b)), *kmeans-d1* and *kmeans-d2* (our own implementation of Ieva et al. (2013)) and FGRC (provided at our request by the authors Yamamoto and Hwang (2017)). All algorithms are applied for $K = 3$ groups for *Scenario A* and $K = 4$ groups for *Scenario B* and *Scenario C*. These methods are compared on the basis of the 3 simulation settings and according to the adjusted Rand index (ARI).

Table 5 presents clustering accuracies for the 10 tested models and the best funHDDC model selected at each iteration by slope heuristic or BIC. These scenarios seem to be hard situations since only funHDDC performs well for the 3 of them. Those good results for funHDDC are due to the fact that the MFPCA are carried out cluster per cluster. FGRC performed well for 2 out of 3 scenarios, it is the second best method behind funHDDC. Let also remark that both *kmeans* methods have a high variance. The SH does not perform as well as in the previous example. SH seems to be a good criterion to select the number of clusters, but, with a number of clusters fixed, the BIC seems to be a better criterion for model selection. One can also wonder if this counter performance of the SH is not linked to the small number of tested models.

7 Case study: analysis of pollution in French cities

This section focuses on the analysis of pollution data in French cities. The monitoring and the analysis of such data is of course important in the sense

Table 5 Mean (and s.d) of ARI for all tested models on 50 simulations

Method	Model	Scenario A	Scenario B	Scenario C
funHDDC	$[a_{kj}b_kQ_kD_k]$	0.99 (0.08)	**0.98 (0.08)**	0.94 (0.14)
funHDDC	$[a_{kj}bQ_kD_k]$	0.85 (0.26)	0.82 (0.19)	0.76 (0.19)
funHDDC	$[a_kb_kQ_kD_k]$	**1 (0)**	0.96 (0.11)	0.94 (0.13)
funHDDC	$[a_kbQ_kD_k]$	0.88 (0.26)	0.88 (0.18)	0.81 (0.20)
funHDDC	$[ab_kQ_kD_k]$	0.95 (0.16)	**0.98 (0.09)**	**0.95 (0.13)**
funHDDC	$[abQ_kD_k]$	0.49 (0.36)	0.86 (0.18)	0.78 (0.23)
funHDDC	SH best model	0.48 (0.29)	0.76 (0.18)	0.70 (0.14)
funHDDC	BIC best model	0.97 (0.12)	0.86 (0.18)	0.79 (0.18)
Funclust	-	0.30 (0.27)	0.42 (0.25)	0.41 (0.24)
$kmeans - d_1$	-	0.57 (0.49)	0.18 (0.37)	0.30 (0.46)
$kmeans - d_2$	-	0.61 (0.48)	0.29 (0.43)	0.18 (0.37)
FGRC	-	0.87 (0.01)	0.65 (0.21)	0.81 (0.19)

that they could help cities in designing their policy against pollution. As a reminder, pollution kills at least nine million people and costs trillions of dollars every year, according to the most comprehensive global analyses to date [1].

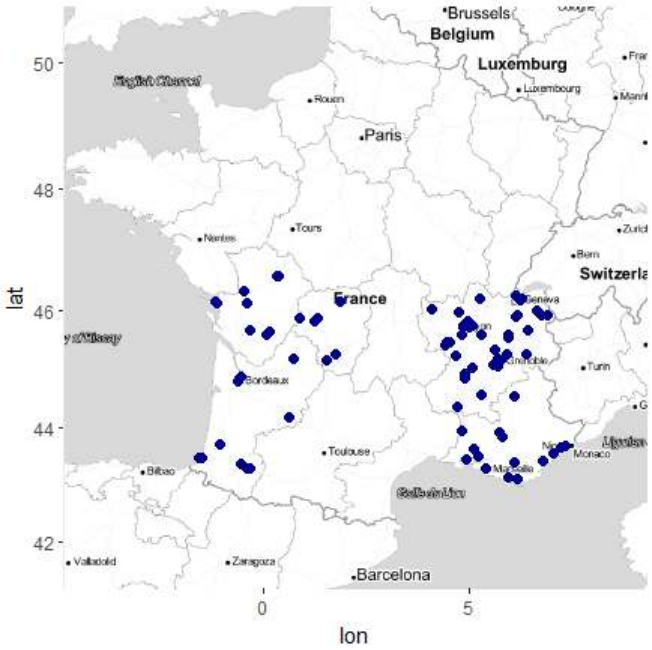


Fig. 5 Location of measured cities (dark blue)

[1] who.int/airpollution/en/

7.1 Data

This dataset deals with pollution in some French cities (available on 3 different websites [2]). It has been documented by Atmo France, a federation which monitor air quality in France. It gathered 5 categories of pollutants measured hourly since 1985. In this study we choose to work on Ozone value ($\mu g/m^3$) and PM10 particles ($\mu g/m^3$) measured in 84 France southern cities. The regions affected are Nouvelle Aquitaine, Auvergne Rhône Alpes and Provence Alpes Côte d'Azur (cf. Figure 5). A period of one year from 1/01/2017 to 31/12/2017 is considered. The goal of this study is to characterize the daily evolution of these polluants. In order to do this, data are split into daily curves, and the clustering algorithm has been carried out on all the daily curves for all the cities. Doing this, geographical and temporal dependencies between the daily curves are ignored in this preliminary study. Finally, we remove from the analysis the daily curves which have more than 4 missing values or for which there is missing values at the beginning or at the end of the period. The number of bivariate curves to analyse is thus 25,658.

The functional form of the data is reconstructed using a cubic B-spline smoothing with 10 basis functions. As we can see in Figure 6 (bottom), the presence of missing values does not disrupt smoothing. Data are collected through calibrated meteorological stations, we consider that data are obtained in a common acquisition process, then the noise is assumed to be the same for all stations. So, our algorithm has been applied with $[a_{kj}bQ_kD_k]$ model on smoothed data with a varying number of clusters, from 2 to 20. The BIC criteria is used to choose an appropriate number of clusters because there is here not enough models to use the slope heuristic criteria.

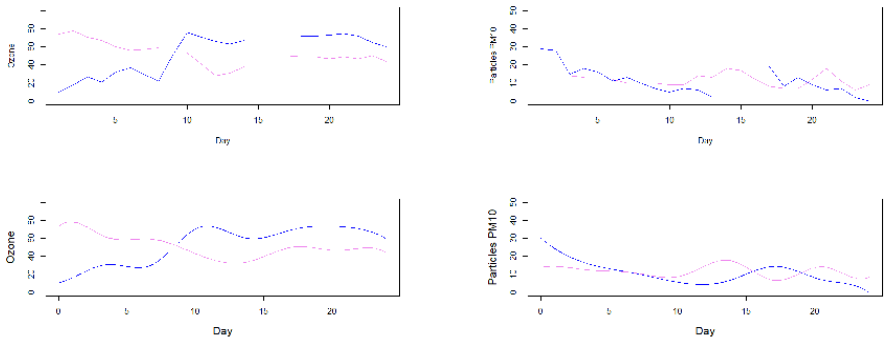


Fig. 6 Pollutants real curves (top) and smooth curves (bottom) for Avignon day 12 (blue) and La Rochelle day 177 (pink)

[2] https://www.airpaca.org/donnees/telecharger,
https://www.atmo-auvergnerhonealpes.fr/donnees/telecharger,
https://www.atmo-nouvelleaquitaine.org/donnees/telecharger

7.2 Results

Table 6 BIC values for the 10 first number of clusters, with $[a_{kj}bQ_kD_k]$

Number of clusters	Complexity	BIC
6	544	-4,756,123.71
9	849	-4,778,048.84
18	2,057	-4,819,115.13
17	1,929	-4,833,556.40
5	472	-4,939,371.01
14	1,545	-4,966,517.37
16	1,834	-4,969,406.42
15	1,735	-4,970,505.90
11	1,260	-4,972,458.71
2	101	-4,976,490.18

According to BIC, the best partition for $[a_{kj}bQ_kD_k]$ model is with 6 clusters (cf. Table 6).

The main sources of variation of Ozone and particles PM10 overall mean curves can be studied thanks to the MFPCA performed for each subgroup. The solid curve on Figure 7 and 8 represents the overall mean curve and the blue and red ones show the effect of adding (resp. subtracting) a multiple of the first eigenfunction and can be interpret as the first source of variation of the overall mean. According to this plot the first source is an amplitude variation for Ozone. For PM10 particles it is the peaks size that varies the most between groups.

Looking at the groups mean curves (cf. Figure 7 for Ozone and 8 for PM10 particles) we can see that Ozone mean curves are more variable than PM10 particles ones. We can also see a common pattern between groups. For the PM10 particles, the mean curves of each group have a wavy shape with a first summit at night and a second at mid-afternoon. There is two main patterns in the O3 curves. During a day, the Ozone concentration has a tendency to decrease from midnight to midday and to increase until reaching a plateau between 5 pm and 8 pm for the first pattern. For the second one, the Ozone concentration is stable from midnight to 2 pm and increases until reaching a maximum at 8 pm. But it is the level of concentration that varies the most from a group to another.

The dark blue group is characterized by the lower concentration of Ozone along the day. This group gathers winter days (cf. Figure 9) for cities mostly in urban area (cf. Table 7). Ozone is a product of photochemical reaction between various pollutants when there is a lot of sunlight. The low duration of sunshine during winter can explain those low values. Its Ozone mean curve is very close to turquoise group one, they differ from their PM10 particle concentration. Indeed, dark blue average curve is three times higher than turquoise group one. It gathers days the most contaminated by particles PM10 (with the highest concentration along the day). For that matter, the European Union recommends that PM10 concentrations should not to be higher than 50 $\mu g/m^3$ in daily mean more than 35 days per year and in this group the mean value is above this threshold at any time. Whereas turquoise group gathers fall and winter days (cf. Figure 9). The black group has the highest values of Ozone (cf. Figure 7 group 5). Its maximum is reached between 5pm and 10pm. This can be due to exhaust gas when people commute from their work to their home.

Table 7 Proportion of city type in each group

Type of city	Whole dataset	Dark blue Group	Pink Group	Turquoise Group
Urban	0.78	0.81	0.75	0.84
Suburban	0.17	0.16	0.19	0.14
Rural	0.05	0.03	0.06	0.03
Type of city	Grey Group	Black Group	Purple Group	
Urban	0.79	0.78	0.77	
Suburban	0.16	0.17	0.17	
Rural	0.06	0.05	0.05	

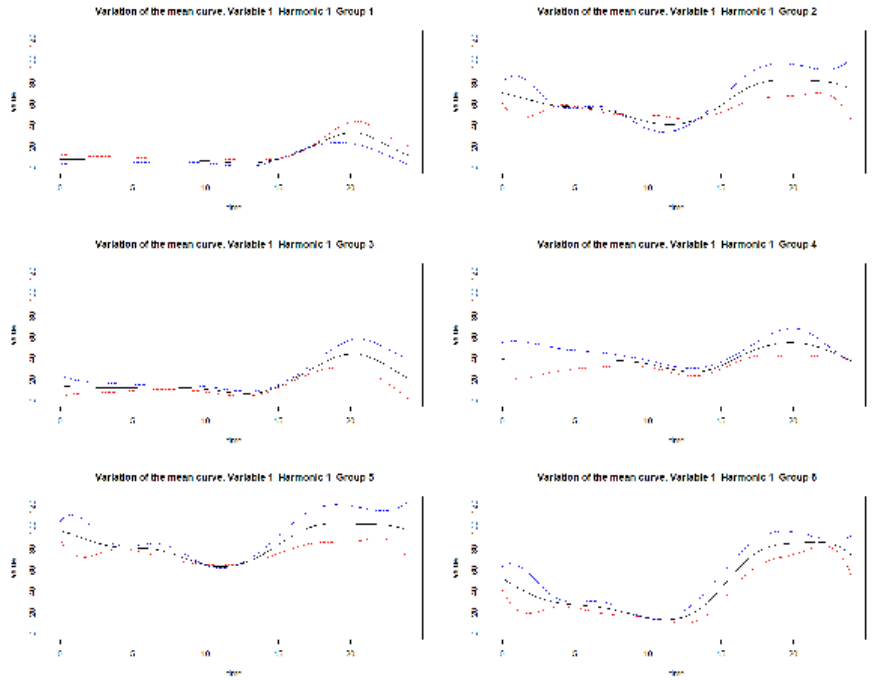


Fig. 7 O3 overall mean curve and its variation expressed in each group functional subspace. Blue (resp. red) curve shows the effect of adding (resp. subtracting) a multiple of the first eigenfunction and represents the first source of the overall mean variation

To conclude, the use of multiple variables to cluster cities allow the distinction between different pollution profiles. Those results enable local councils to have a look at the daily pollution of their towns along the year. Those results have especially highlighted critical days in particles PM10 pollution, that can lead to recommendations in order to try to lower these levels the next year. However we have to stay vigilant about the interpretation of those results. In fact, the measurement of contaminating elements are very localized, some sensors are located near companies and thus are not always representative of the pollution of the whole city in which they are located.

8 Discussion and conclusion

This work was motivated by the will to provide a new clustering method for multivariate functional data, called funHDDC, which takes into account the possibility that data live in subspaces of different dimensions. The method is based on a multivariate functional principal component analysis and a functional latent mixture model. Its efficiency has been demonstrated on simulated datasets and the proposed technique outperforms *state-of-the-art* methods for

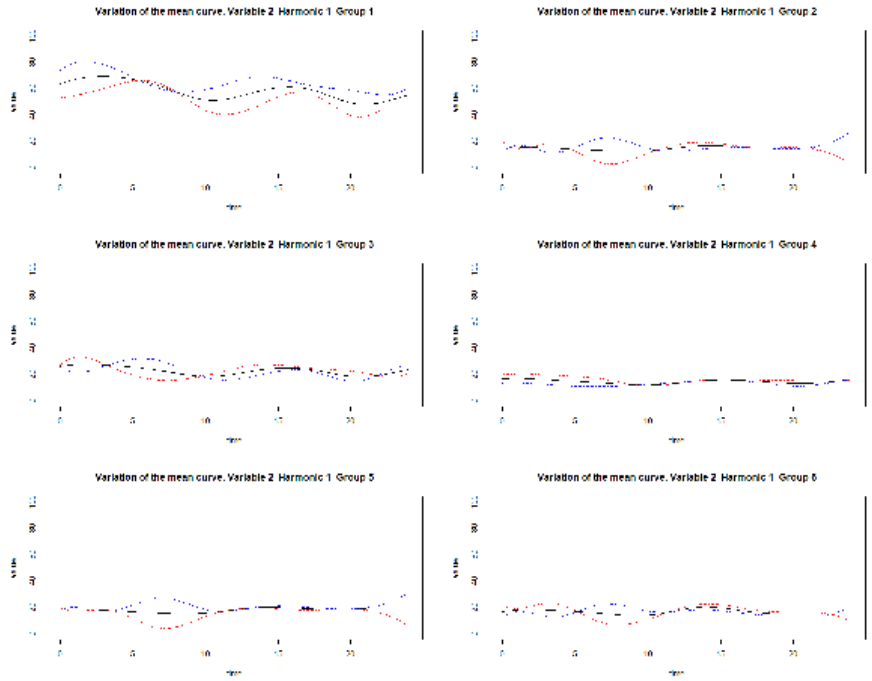


Fig. 8 PM10 particles overall mean curve and its variation expressed in each group functional subspace. Blue (resp. red) curve shows the effect of adding (resp. subtracting) a multiple of the first eigenfunction and represents the first source of the overall mean variation

clustering multivariate functional data. Notice also that this new algorithm works in the univariate case as well and, therefore, generalizes the original funHDDC algorithm (Bouveyron and Jacques (2011)). It is available on CRAN as the funHDDC package. The proposed methodology has been applied to analyze one-year pollution records in 84 cities in France, with meaningful results. It is worth noticing that smoothing data with basis functions allows to both filter the level of information one wants to keep and to deal with missing data. Let also remark that wavelet smoothing may keep more information in the case of peaked data than B-spline smoothing. It can be the subject of future work because a new model will have to be adapted to this smoothing. Similarly, further developments of the proposed approach could be investigated in order to take into account dependency between observations, following the large literature about dependent functional data.

Conflict of interest

The authors declare that they have no conflict of interest.

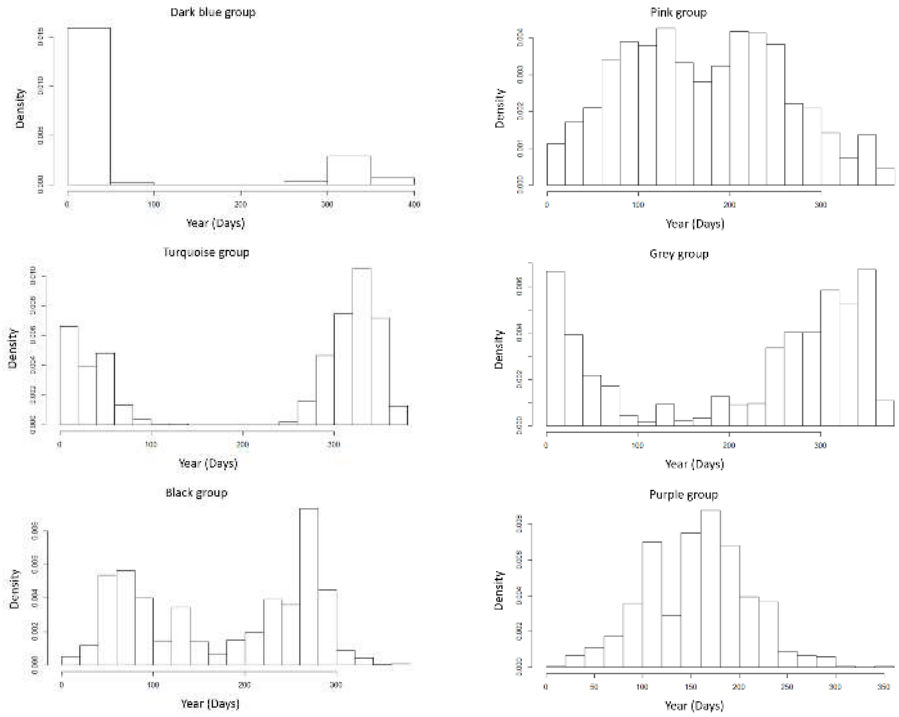


Fig. 9 Histogram of days in each group, from 1/01/2017 (0) to 31/12/2010 (364). Spring is from day 79 to 171, Summer from day 172 to 264, Fall from day 265 to 354 and Winter from day 355 to 365 and day 1 to 78.

A Appendix: proofs

A.1 Proof of Proposition 1.

$$l(\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ki} log(\pi_k f(x_i, \theta_k)),$$

where z_{ki}=1 if x_i belongs to the cluster k and $z_{ki} = 0$ otherwise. $f(x_i, \theta_k)$ is a Gaussian density, with parameters $\theta_k = \{\mu_k, \Sigma_k\}$. So the complete log-likelihood is written:

$$\begin{aligned} l(\theta) &= \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ki} log[\pi_k \frac{1}{(2\pi)^{R/2}|\Sigma_k|^{1/2}} exp(\frac{-1}{2}(x_i-\mu_k)^t \Sigma_k^{-1}(x_i-\mu_k))] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ki}[log(\pi_k) - \frac{1}{2}log|\Sigma_k| - \frac{1}{2}(x_i-\mu_k)^t\Sigma_k^{-1}(x_i-\mu_k) - \frac{R}{2}log(2\pi)]. \end{aligned}$$

For the $[a_{kj}b_kQ_kd_k]$ model, we have:

$$\begin{aligned} l(\theta) &= \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} z_{ki}[-2log(\pi_k) + log(\prod_{j=1}^{d_k} a_{kj} \prod_{j=d_k+1}^{R} b_k) + (x_i-\mu_k)^t Q_k \Delta_k^{-1} Q_k^t (x_i-\mu_k)] \\ &- \frac{nR}{2}log(2\pi) \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} z_{ki}[-2log(\pi_k) + \sum_{j=1}^{d_k} log(a_{kj}) + \sum_{j=d_k+1}^{R} log(b_k)] \\ &+ (x_i-\mu_k)^t Q_k \Delta_k^{-1} Q_k^t (x_i-\mu_k)] - \frac{nR}{2}log(2\pi). \end{aligned}$$

Let $n_k = \sum_{i=1}^{n} z_{ki}$ be the number of curves within cluster k, the complete log-likelihood is then written:

$$\begin{aligned} l(\theta) = -\frac{1}{2}\sum_{k=1}^{K} n_k[-2log(\pi_k) + \sum_{j=1}^{d_k} log(a_{kj}) + \sum_{j=d_k+1}^{R} log(b_k) \\ + \frac{1}{n_k}\sum_{i=1}^{n} z_{kj}(x_i-\mu_k)^t Q_k \Delta_k^{-1} Q_k^t (x_i-\mu_k)] - \frac{nR}{2}log(2\pi). \end{aligned}$$

The quantity $(x_i-\mu_k)^t Q_k \Delta_k^{-1} Q_k^t (x_i-\mu_k)$ is a scalar, so it is equal to it trace:

$$\frac{1}{n_k}\sum_{i=1}^{n} z_{ki}(x_i-\mu_k)^t Q_k \Delta_k^{-1} Q_k^t (x_i-\mu_k) = \frac{1}{n_k}\sum_{i=1}^{n} z_{ki} tr((x_i-\mu_k)^t Q_k \Delta_k^{-1} Q_k^t (x_i-\mu_k)).$$

Well $tr([(x_i-\mu_k)^t Q_k] \times [\Delta_k^{-1} Q_k^t (x_i-\mu_k)]) = tr([\Delta_k^{-1} Q_k^t (x_i-\mu_k)] \times [(x_i-\mu_k)^t Q_k])$, consequently:

$$\begin{aligned} \frac{1}{n_k}\sum_{i=1}^{n} z_{ki}(x_i-\mu_k)^t Q_k \Delta_k^{-1} Q_k^t (x_i-\mu_k) &= \frac{1}{n_k}\sum_{i=1}^{n} z_{ki} tr(\Delta_k^{-1} Q_k^t (x_i-\mu_k)(x_i-\mu_k)^t Q_k) \\ &= tr(\Delta_k^{-1} Q_k^t [\frac{1}{n_k}\sum_{i=1}^{n} z_{ki}(x_i-\mu_k)^t (x_i-\mu_k)] Q_k) \\ &= tr(\Delta_k^{-1} Q_k^t C_k Q_k), \end{aligned}$$

where $C_k = \frac{1}{n_k}\sum_{i=1}^{n} z_{ki}(x_i-\mu_k)^t(x_i-\mu_k)$ is the empirical covariance matrix of the k-th element of the mixture model. The Δ_k matrix is diagonal, so we can write:

$$\frac{1}{n_k}\sum_{i=1}^{n} z_{ki}(x_i-\mu_k)^t Q_k \Delta_k^{-1} Q_k^t (x_i-\mu_k) = \sum_{j=1}^{d_k} \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{a_{kj}} + \sum_{j=d_k+1}^{R} \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{b_k},$$

where q_{kj} is j-th column of Q_k.
Finally,

$$l(\theta) = -\frac{1}{2}\sum_{k=1}^{K} n_k[-2log(\pi_k) + \sum_{j=1}^{d_k} log(a_{kj}) + \sum_{j=d_k+1}^{R} log(b_k) + \sum_{j=1}^{d_k} \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{a_{kj}} + \sum_{j=d_k+1}^{R} \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{b_k}] + \frac{nR}{2} log(2\pi).$$

A.2 Proof of Proposition 2.

$$\begin{aligned} H_k(x) &= -2log(\pi_k f(x,\theta_k)) \\ &= -2log(\pi_k) - 2log(f(x,\theta_k)) \\ &= -2log(\pi_k) - 2log(\frac{1}{(2\pi)^{R/2}|\Sigma_k|^{1/2}} exp(\frac{-1}{2}(x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k)) \\ &= -2log(\pi_k) - 2log(\frac{1}{(2\pi)^{R/2}|\Sigma_k|^{1/2})}) + (x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k) \\ &= -2log(\pi_k) + Rlog(2\pi) + log|\Sigma_k| + (x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k). \end{aligned}$$

But, $\Sigma_k = Q_k \Delta_k Q_k^t$ and $Q_k^t Q_k = I_R$, hence:

$$H_k(x) = -2log(\pi_k) + Rlog(2\pi) + log|\Sigma_k| + (x_k-\mu_k)^t (Q_k \Delta_k Q_k^t)^{-1}(x_k-\mu_k).$$

Let $Q_k = \tilde{Q_k} + \bar{Q_k}$ where $\tilde{Q_k}$ is the $R \times R$ matrix containing the d_k first columns of Q_k completed by zeros and where $\bar{Q_k} = Q_k - \tilde{Q_k}$. Notice that $\tilde{Q_k}\Delta_k^{-1}\bar{Q_k}^t = \bar{Q_k}\Delta_k^{-1}\tilde{Q_k}^t = O_p$ where O_p is the null matrix. So,

$$Q_k \Delta_k^{-1} Q_k^t = (\tilde{Q_k} + \bar{Q_k})\Delta_k^{-1}(\tilde{Q_k} + \bar{Q_k}) = \tilde{Q_k}\Delta_k^{-1}\tilde{Q_k} + \bar{Q_k}\Delta_k^{-1}\bar{Q_k}.$$

Hence,

$$\begin{aligned} H_k(x) = &-2log(\pi_k) + Rlog(2\pi) + log|\Sigma_k| + (x-\mu_k)^t \tilde{Q_k}\Delta_k^{-1}\tilde{Q_k^t}(x-\mu_k) \\ &+ (x-\mu_k)^t \bar{Q_k}\Delta_k^{-1}\bar{Q_k^t}(x-\mu_k). \end{aligned}$$

With definitions $\tilde{Q_k}[\tilde{Q_k^t}\tilde{Q_k}] = \tilde{Q_k}$ and $\bar{Q_k}[\bar{Q_k^t}\bar{Q_k}] = \bar{Q_k}$, we can rephrase $H_k(x)$ as:

$$\begin{aligned} H_k(x) = &-2log(\pi_k) + Rlog(2\pi) + log|\Sigma_k| + (x-\mu_k)^t \tilde{Q_k}\tilde{Q_k^t}\tilde{Q_k}\Delta_k^{-1}\tilde{Q_k^t}\tilde{Q_k}\tilde{Q_k^t}(x-\mu_k) \\ &+ (x-\mu_k)^t \bar{Q_k}\bar{Q_k^t}\bar{Q_k}\Delta_k^{-1}\bar{Q_k^t}\bar{Q_k}\bar{Q_k^t}(x-\mu_k) \\ = &-2log(\pi_k) + Rlog(2\pi) + log|\Sigma_k| + [\tilde{Q_k}\tilde{Q_k^t}(x-\mu_k)]^t \tilde{Q_k}\Delta_k^{-1}\tilde{Q_k}^t[\tilde{Q_k}\tilde{Q_k^t}(x-\mu_k)] \\ &+ [\bar{Q_k}\bar{Q_k^t}(x-\mu_k)]^t \bar{Q_k}\Delta_k^{-1}\bar{Q_k}^t[\bar{Q_k}\bar{Q_k^t}(x-\mu_k)]. \end{aligned}$$

We define $\mathcal{D}_k = \tilde{Q_k}\Delta_k^{-1}\tilde{Q_k^t}$ and the norm $||.||_{\mathcal{D}_k}$ on $\mathbb{E}_k$ such as $||x||_{\mathcal{D}_k} = x^t\mathcal{D}_k x$. So, on one hand:

$$[\tilde{Q_k}\tilde{Q_k^t}(x-\mu_k)]^t\tilde{Q_k}\Delta_k^{-1}\tilde{Q_k^t}[\tilde{Q_k}\tilde{Q_k^t}(x-\mu_k)] = ||\tilde{Q_k}\tilde{Q_k^t}(x-\mu_k)||^2_{\mathcal{D}_k}.$$

On the other hand:

$$[\bar{Q_k}\bar{Q_k^t}(x-\mu_k)]^t\bar{Q_k}\Delta_k^{-1}\bar{Q_k^t}[\bar{Q_k}\bar{Q_k^t}(x-\mu_k)] = \frac{1}{b_k}||\bar{Q_k}\bar{Q_k^t}(x-\mu_k)||^2.$$

Consequently,

$$H_k(x) = -2log(\pi_k) + Rlog(2\pi) + log|\Sigma_k| + ||\tilde{Q_k}\tilde{Q_k^t}(x-\mu_k)||^2_{\mathcal{D}_k} + \frac{1}{b_k}||\bar{Q_k}\bar{Q_k^t}(x-\mu_k)||^2.$$

Knowing P_k, $P_k^\perp$ and $||\mu_k - P_k^\perp||^2 = ||x - P_k(x)||^2$, we have:

$$H_k(x) = ||\mu_k - P_k(x)||^2_{\mathcal{D}_k} + \frac{1}{b_k}||x - P_k(x)||^2 + log|\Sigma_k| - 2log(\pi_k) + Rlog(2\pi).$$

Moreover, $log|\Sigma_k| = \sum_{j=1}^{d_k} log(a_{kj}) + (R-d_k)log(b_k)$.
Finally,

$$H_k(x) = ||\mu_k - P_k(x)||^2_{\mathcal{D}_k} + \frac{1}{b_k}||x - P_k(x)||^2 + \sum_{j=1}^{d_k} log(a_{kj}) + (R-d_k)log(b_k) - 2log(\pi_k) + Rlog(2\pi).$$

A.3 Proof of Proposition 3.

Parameter Q_k We have to maximise the log-likelihood under the constraint $q_{kj}^t q_{kj} = 1$, which is equivalent to looking for a saddle point of the Lagrange function:

$$\mathcal{L} = -2l(\theta) - \sum_{j=1}^{R}\omega_{kj}(q_{kj}^t q_{kj} - 1),$$

where ω_{kj} are Lagrange multiplier. So we can write:

$$\begin{aligned}\mathcal{L} = &\sum_{k=1}^{K}\eta_k[\sum_{j=1}^{d_k}(log(a_{kj}) + \frac{q_{kj}^t W^{1/2}C_k W^{1/2}q_{kj}}{a_{kj}}) \\ &+ \sum_{j=d_k+1}^{R}(log(b_k) + \frac{q_{kj}^t W^{1/2}C_k W^{1/2}q_{kj}}{b_k}) - 2log(\pi_k)] + \frac{nR}{2}log(2\pi) \\ &- \sum_{j=1}^{R}\omega_{kj}(q_{kj}^t q_{kj} - 1).\end{aligned}$$

Therefore, the gradient of $\mathcal{L}$ in relation to q_{kj} is:

$$\begin{aligned}\nabla_{q_{kj}}\mathcal{L} = \nabla_{q_{kj}}(&\sum_{k=1}^{K}\eta_k[\sum_{j=1}^{d_k}\frac{q_{kj}^t W^{1/2}C_k W^{1/2}q_{kj}}{a_{kj}} + \sum_{j=d_k+1}^{R}\frac{q_{kj}^t W^{1/2}C_k W^{1/2}q_{kj}}{b_k}] \\ &- \sum_{j=1}^{R}\omega_{kj}(q_{kj}^t q_{kj} - 1)).\end{aligned}$$

As a reminder, when W is symmetric, then $\frac{\partial}{\partial x}(x-s)^T W(x-s) = 2W(x-s)$ and $\frac{\partial}{\partial x}(x^T x) = 2x$ (cf. Petersen and Pedersen (2012)), so:

$$\nabla_{q_{kj}} \mathcal{L} = \eta_k [2 \frac{W^{1/2} C_k W^{1/2}}{\sigma_{kj}} q_{kj}] - 2\omega_{kj} q_{kj}$$

where σ_{kj} is the j-th diagonal term of matrix Δ_k.
So,

$$\begin{aligned} q_{kj}^t \nabla_{q_{kj}} \mathcal{L} = 0 &\Leftrightarrow \omega_{kj} q_{kj} = \frac{\eta_k}{\sigma_{kj}} q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj} \\ &\Leftrightarrow W^{1/2} C_k W^{1/2} q_{kj} = \frac{\omega_{kj}\sigma_{kj}}{\eta_k} q_{kj}. \end{aligned}$$

q_{kj} is the eigenfunction of $W^{1/2} C_k W^{1/2}$ which match the eigenvalue $\lambda_{kj} = \frac{\omega_{kj}\sigma_{kj}}{\eta_k} = W^{1/2} C_k W^{1/2}$. We can write $q_{kj}^t q_{kl} = 0$ if $j \neq l$. So the log-likelihood can be written:

$$-2l(\theta) = \sum_{k=1}^{K} \eta_k [\sum_{j=1}^{d_k} (log(a_{kj}) + \frac{\lambda_{kj}}{a_{kj}}) + \sum_{j=d_k+1}^{R} (log(b_k) + \frac{\lambda_{kj}}{b_k})] + C^{te},$$

we substitute the equation $\sum_{j=d_k+1}^{R} \lambda_{kj} = tr(W^{1/2} C_k W^{1/2}) - \sum_{j=1}^{d_k} \lambda_{kj}$:

$$\begin{aligned} -2l(\theta) &= \sum_{k=1}^{K} \eta_k [\sum_{j=1}^{d_k} log(a_{kj}) + \sum_{j=1}^{d_k} \frac{\lambda_{kj}}{a_{kj}} + \sum_{j=d_i+1}^{R} log(b_k) + \frac{1}{b_k}(tr(W^{1/2} C_k W^{1/2}) - \sum_{j=1}^{d_k} \lambda_{kj})] + C^{te} \\ &= \sum_{k=1}^{K} \eta_k [\sum_{j=1}^{d_k} log(a_{kj}) + \sum_{j=1}^{d_k} \lambda_{kj} (\frac{1}{a_{kj}} - \frac{1}{b_k}) + \sum_{j=d_i+1}^{R} log(b_k) + \frac{1}{b_k} tr(W^{1/2} C_k W^{1/2})] + C^{te} \\ &= \sum_{k=1}^{K} \eta_k [\sum_{j=1}^{d_k} log(a_{kj}) + \sum_{j=1}^{d_k} \lambda_{kj} (\frac{1}{a_{kj}} - \frac{1}{b_k}) + (p - d_k) log(b_k) + \frac{tr(W^{1/2} C_k W^{1/2})}{b_k}] + C^{te}. \end{aligned}$$

In order to minimize $-2l(\theta)$ compared to q_{kj}, we minimize the quantity $\sum_{k=1}^{K} \eta_k \sum_{j=1}^{d_k} \lambda_{kj} (\frac{1}{a_{kj}} - \frac{1}{b_k})$ compared to λ_{kj}. Knowing that $(\frac{1}{a_{kj}} - \frac{1}{b_k}) \leq 0, \forall j = 1, ..., d_k$, λ_{kj} has to be as high as feasible. So, the j-th column q_{kj} of matrix Q is estimated by the eigenfunction associated to the j-th highest eigenvalue of $W^{1/2} C_k W^{1/2}$.

Parameter a_{kj} As a reminder $(ln(x))' = \frac{x'}{x}$ and $(\frac{1}{x})' = -\frac{1}{x^2}$. The partial derivative of $l(\theta)$ in relation to a_{kj} is:

$$-2 \frac{\partial l(\theta)}{\partial a_{kj}} = \eta_k (\frac{1}{a_{kj}} - \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{a_{kj}^2})$$

The condition $\frac{\partial l(\theta)}{\partial a_{kj}} = 0$ is equivalent to:

$$\begin{aligned} \eta_k (\frac{1}{a_{kj}} - \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{a_{kj}^2}) = 0 \\ \Leftrightarrow \frac{1}{a_{kj}} = \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{a_{kj}^2} \\ \Leftrightarrow a_{kj} = q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj} \\ \Leftrightarrow a_{kj} = \lambda_{kj} \end{aligned}$$

Parameter b_k The partial derivative of $l(\theta)$ in relation to b_k is:

$$\begin{aligned}-2\frac{\partial l(\theta)}{\partial b_k} &= \eta_k \sum_{j=d_k+1}^{R} (\frac{1}{b_k} - \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{b_k^2}) \\ &= \eta_k (\frac{R-d_k}{b_k} - \sum_{j=d_k+1}^{R} \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{b_k^2})\end{aligned}$$

But,

$$\sum_{j=d_k+1}^{R} q_j^t W^{1/2} C_k W^{1/2} q_j = tr(W^{1/2} C_k W^{1/2}) - \sum_{j=1}^{d_k} q_j^t W^{1/2} C_k W^{1/2} q_j,$$

so:

$$\begin{aligned}-2\frac{\partial l(\theta)}{\partial b_k} &= \eta_k \frac{(R-d_k)}{b_k} - \frac{\eta_k}{b_k^2}(tr(W^{1/2} C_k W^{1/2}) - \sum_{j=1}^{d_k} q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}) \\ &= \eta_k \frac{(R-d_k)}{b_k} - \frac{\eta_k}{b_k^2}(tr(W^{1/2} C_k W^{1/2}) - \sum_{j=1}^{d_k} \lambda_{kj})\end{aligned}$$

The condition $\frac{\partial l(\theta)}{\partial b_k} = 0$ is equivalent to:

$$\begin{aligned}&\eta_k \frac{(R-d_k)}{b_k} - \frac{\eta_k}{b_k^2}(tr(W^{1/2} C_k W^{1/2}) - \sum_{j=1}^{d_k} \lambda_{kj}) = 0 \\ \Leftrightarrow\ &\eta_k \frac{(R-d_k)}{b_k} = \frac{\eta_k}{b_k^2}(tr(W^{1/2} C_k W^{1/2}) - \sum_{j=1}^{d_k} \lambda_{kj}) \\ \Leftrightarrow\ &b_k = \frac{\eta_k}{\eta_k (R-d_k)}(tr(W^{1/2} C_k W^{1/2}) - \sum_{j=1}^{d_k} \lambda_{kj}) \\ \Leftrightarrow\ &b_k = \frac{1}{(R-d_k)}(tr(W^{1/2} C_k W^{1/2}) - \sum_{j=1}^{d_k} \lambda_{kj})\end{aligned}$$

References

Akaike H (1974) A new look at the statistical model identification. IEEE Transactions on Automatic Control 9:716–723

Berrendero J, Justel A, Svarc M (2011) Principal components for multivariate functional data. Computational Statistics and Data Analysis 55:2619–263

Biernacki C, Celeux G, Govaert G (2000) Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans PAMI 22:719–725

Birge L, Massart P (2007) Minimal penalties for gaussian model selection. Probability theory and related fields 138:33–73

Bongiorno EG, Goia A (2016) Classification methods for hilbert data based on surrogate density. Comput Stat Data Anal 99(C):204–222, DOI 10.1016/j.csda.2016.01.019, URL `http://dx.doi.org/10.1016/j.csda.2016.01.019`

Bouveyron C, Jacques J (2011) Model-based clustering of time series in group-specific functional subspaces. Advances in Data Analysis and Classification 5(4):281–300

Bouveyron C, Come E, Jacques J (2015) The discriminative functional mixture model for the analysis of bike sharing systems. Annals of Applied Statistics 9(4):1726–1760

Cattell R (1966) The scree test for the number of factors. Multivariate Behaviour Research 1(2):245–276

Chen L, Jiang C (2016) Multi-dimensional functional principal component analysis. Statistics and Computing 27:1181–1192

Chiou J, Chen Y, Yang Y (2014) Multivariate functional principal component analysis: a normalization approach. Statistica Sinica 24:1571–1596

Chiou JM, Li PL (2007) Functional clustering and identifying substructures of longitudinal data. Journal of the Royal Statistical Society Series B Statistical Methodology 69(4):679–699

Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39(1):1–38

Dias R, Collazos J, Zambom A (2018) Functional data clustering via hypothesis testing k-means. Computational Statistics DOI 10.1007/s00180-018-0808-9

Ferraty F, Vieu P (2003) Curves discrimination: a nonparametric approach. Computational Statistics and Data Analysis 44:161–173

Happ C, Greven S (2015) Multivariate functional principal component analysis for data observed on different (dimensional) domains. Journal of the American Statistical Association p in press

Ieva F, Paganoni A (2013) Risk prediction for myocardial infarction via generalized functional regression models. Statistical methods in medical research 25, DOI 10.1177/0962280213495988

Ieva F, Paganoni A, Pigoli D, Vitelli V (2013) Multivariate Functional Clustering for the Morphological Analysis of ECG Curves. Journal of the Royal Statistical Society, Series C (Applied Statistics) 62(3):401–418

Jacques J, Preda C (2013) Funclust: a curves clustering method using functional random variable density approximation. Neurocomputing 112:164–171

Jacques J, Preda C (2014a) Functional data clustering: a survey. Advances in Data Analysis and Classification 8(3):231–255

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

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

Kayano M, Dozono K, Konishi S (2010) Functional Cluster Analysis via Orthonormalized Gaussian Basis Expansions and Its Application. Journal of Classification 27:211–230

Petersen KB, Pedersen MS (2012) The matrix cookbook. URL `http://www2.imm.dtu.dk/pubdb/p.php?3274`, version 20121115

Preda C (2007) Regression models for functional data by reproducing kernel hilbert spaces methods. Journal of Statistical Planning and Inference 137:829–840

R Core Team (2017) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL `https://www.R-project.org/`

Ramsay JO, Silverman BW (2005) Functional data analysis, 2nd edn. Springer Series in Statistics, Springer, New York

Rand WM (1971) Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66(336):846–850

Saporta G (1981) Méthodes exploratoires d'analyse de données temporelles. Cahiers du Buro 37–38

Schwarz G (1978) Estimating the dimension of a model. The Annals of Statistics 6(2):461–464

Singhal A, Seborg D (2005) Clustering multivariate time-series data. Journal of Chemometrics 19:427–438

Tarpey T, Kinateder K (2003) Clustering functional data. Journal of Classification 20(1):93–114

Tokushige S, Yadohisa H, Inada K (2007) Crisp and fuzzy k-means clustering algorithms for multivariate functional data. Computational Statistics 22:1–16

Traore OI, Cristini P, Favretto-Cristini N, Pantera L, Vieu P, Viguier-Pla S (2019) Clustering acoustic emission signals by mixing two stages dimension reduction and nonparametric approaches. Computational Statistics DOI 10.1007/s00180-018-00864-w

Yamamoto M (2012) Clustering of Functional Data in a Low-Dimensional Subspace. Advances in Data Analysis and Classification 6:219–247

Yamamoto M, Hwang H (2017) Dimension-Reduced Clustering of Functional Data via Subspace Separation. Journal of Classification 34:294–326

Yamamoto M, Terada Y (2014) Functional Factorial k-Means Analysis. Computational Statistics and Data Analysis 79:133–148