Clustering multivariate functional data in group-specific functional subspaces

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

Advances in Data Analysis and Classification 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 in 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 EM 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 data · multivariate functional principal component analysis · model-based clustering · EM algorithm

Amandine Schmutz · Pauline Martin
Lim France, Chemin Fontaine de Fanny, Nontron, France
CWD-VetLab, Ecole Nationale Vétérinaire d'Alfort, Maisons-Alfort, F-94700, France
E-mail: aschmutz@lim-group.com

Julien Jacques
Université de Lyon, Lyon 2, ERIC EA3083, Lyon, France

Charles Bouveyron
Université Côte d'Azur, LJAD - UMR 7351 & Epione - Inria Sophia Antipolis, Nice, France

Laurence Chèze
Université de Lyon, Lyon 1, LBMC UMR T9406, Lyon, France

1 Introduction

The modern technologies ease the collection of high frequency data which is of interest to model and understand 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 the 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, the need of methods to identify homogeneous subgroups of data is increasing in order to make better individualized predictions for example. 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) and Bouveyron et al (2015). But 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 live in the same subspace. Other clustering methods based on dimension reduction techniques exist in order to obtain a low-dimensional representation of functions. 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 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). In this method, MFPCA scores are considered as random variables whose probability distributions are cluster specific. Although this last 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. Which is a significant

advantage because it will give a best modeling and consequently a best clustering. Moreover, our model allows to use an 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 guaranty. The resulting model can be viewed as an extension of Bouveyron and Jacques (2011) method to the multivariate case, that is why we will refer to it as funHDDC model in the following.
The paper is organized as follows. Section 2 presents principal component analysis for multivariate functional data as introduced in Jacques and Preda (2014b). In Section 3, we introduce the methodology of the clustering for multivariate functional data. Section 4 discusses parameter estimation via an EM algorithm and proposes criteria for the selection of number of clusters. Comparisons between the proposed method and existing ones on simulated and real datasets are presented in Section 5 and 6. A discussion concludes the paper in Section 7.

2 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 a Multivariate Functional Principal Component Analysis (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-Loeve expansion (Saporta, 1981). Chiou et al (2014) present also 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 multidimensional 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 below in Section 2.2.

2.1 Functional data reconstruction

In practice, the functional expressions of the observed curves are not known and we only have access to discrete observations at a finite set of times. A common way to reconstruct the functional form is to express them in a finite dimensional space spanned by a basis of functions. Let $\mathbf{X}_1, ..., \mathbf{X}_n$ be an i.i.d. sample of $\mathbf{X} = \boldsymbol{X}(t)_{t\in[0,T]}$. The observations $\mathbf{X}_1, ..., \mathbf{X}_n$ provide a set of n p-variate curves, with $\mathbf{X}_i = (X_i^1, ..., X_i^p)$. Each curve is assumed to be defined by a linear combination of basis functions:

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

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 chosen for $i \in \{1, ..., n\}$, $j \in \{1, ..., p\}$. The choice of the basis functions is important and there is no straight rules about how to choose the appropriate one (Jacques and Preda (2014a)). In practice, this choice is made by the user, for example Fourier basis are often used in the case of data with a repetitive pattern and Bspline otherwise.

The coefficients c_{ir}^j can be gathered in a matrix:

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

Let also introduce the matrix $\boldsymbol{\phi(t)}$:

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

With these notations, Equation (1) can be written in a matrix form:

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

The estimation of $\boldsymbol{C}$ can be done by least square smoothing.

2.2 Multivariate Functional Principal Component Analysis (MFPCA)

The principle of the MFPCA can be summarized by 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 estimation (3) of ν, 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 matrix of dimension $n \times (\sum_{j=1}^p R_j)$ which contains the inner products between the basis functions. The MFPCA is then reduced to the usual PCA 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 $(\delta_{il})_{l\geq 1} = \boldsymbol{C}\boldsymbol{W}\boldsymbol{b}'_l$.

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 $\sum_{j=1}^p R_j = R$.

3 A generative model for the clustering of multivariate functional data

Our goal is to separate $\boldsymbol{X}_1, ..., \boldsymbol{X}_n$ into K clusters. 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 in a group-specific basis, which is determined thanks to the model, and is obtained from

$\{\phi_r^j\}_{(1\leq j\leq p, 1\leq r\leq R_j)}$ through a linear transformation:

$$\varphi_{kj}(t) = \sum_{l=1}^{R} q_{k,jl}\phi_l(t)$$

where $Q_k = (q_{k,jl})_{l\geq 1}$ is the orthogonal $R \times R$ matrix of eigenfunction coefficients. 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:

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

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

Different submodels can be defined depending on the constraints we apply on model parameters, within or between groups, leading to more parsimonious submodels. This possibility allows to fit into various situations. For example the model $[a_kb_kQ_kd_k]$ is used if the first d_k eigenvalues are fixed to be common within each group. So 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.

4 Model inference and choice of the number of clusters

4.1 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 an Expectation step, which computes the conditional expectation of the complete log-likelihood using the current estimate of parameters; and a Maximisation step, which computes parameter estimates maximizing the expected complete log-likelihood found at the E step.

This section presents the update formula of the EM algorithm in the case of the $[a_{kj}b_kQ_kd_k]$ model. Scores are considered as random variables whose probability distribution is group specific. A Gaussian mixture model is then used on scores with density:

$$p(\boldsymbol{\delta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\delta}; \boldsymbol{\mu}_k, \boldsymbol{\Delta_k}), \tag{6}$$

where $\mathcal{N}$ is the Gaussian density function and $\pi_k = P(Z_k = 1)$. The complete log-likelihood of the observed curves under the $[a_{kj}b_kQ_kd_k]$ model has the following form:

$$\begin{aligned}\ell_c(\theta) = -\frac{1}{2}\sum_{k=1}^{K} n_k \Bigg[&\sum_{j=1}^{d_k}\left(\log(a_{kj}) + \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{a_{kj}}\right) \\ &+ \sum_{j=d_k+1}^{R}\left(\log(b_k) + \frac{q_{kj}^t W^{1/2} C_k W^{1/2} q_{kj}}{b_k}\right) - 2\log(\pi_k)\Bigg] \\ &+ \frac{nR}{2} log(2\pi),\end{aligned} \tag{7}$$

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 jth 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. Proof of this result is provided in Appendix A.1.

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 computes the posterior probability to belong to the kth cluster for each curve:

$$t_{ik}^{(q)} = E[Z_{ik}|c_i, \theta^{(q-1)}] = 1/\sum_{l=1}^{K} \exp[\frac{1}{2}(H_k^{(q-1)}(c_i) - H_l^{(q-1)}(c_i))], \tag{8}$$

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

$$H_k^{(q-1)}(c) = ||\boldsymbol{\mu}_k^{(q-1)} - P_k(c)||^2_{D_k} + \frac{1}{b_k^{(q-1)}}||c - P_k(c)||^2 + \sum_{j=1}^{d_k} \log(a_{kj}^{(q-1)}) + (R - d_k)\log(b_k^{(q-1)}) - 2\log(\pi_k^{(q-1)}), \quad (9)$$

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 $P_k(c) = \boldsymbol{W}U_k U_k^t \boldsymbol{W}^t(c - \boldsymbol{\mu}_k) + \boldsymbol{\mu}_k$. 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 loglikelihood conditionally on the posterior probabilities $t_{ik}^{(q)}$ computed in the previous step. Mixture proportions and means are updated by:

$$\pi_k^{(q)} = \frac{\eta_k^{(q)}}{n}, \;\; \mu_k^{(q)} = \frac{1}{\eta_k^{(q)}} \sum_{i=1}^{n} t_{ik}^{(q)} c_i, \quad (10)$$

where $\eta_k^{(q)} = \sum_{i=1}^n t_{ik}^{(q)}$.

Let us also introduce $C_k^{(q)} = \frac{1}{\eta_k^{(q)}} \sum_{i=1}^n t_{ik}^{(q)}(c_i - \mu_k^{(q)})^t(c_i - \mu_k^{(q)})$, the sample covariance matrix of group k. With these notations, the update formula for the other model parameters a_{kj}, b_k and q_{kj}, in the case of the $[a_{kj}b_kQ_kd_k]$ for $k = 1, ..., K$, are:

- 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 $b_k^{(q)} = \frac{1}{R-d_j}[tr(\boldsymbol{W}^{1/2}C_k^{(q)}\boldsymbol{W}^{1/2}) - \sum_{j=1}^{d_k} \widehat{a}_{kj}^{(q)}]$.

Proof of these results are provided in Appendix A.3.

To summarize, the algorithm introduced above, and named hereafter funHDDC, clusters multivariate functional data through their projection into low dimensional subspaces. Those projections are obtained by perfoming a MFPCA per cluster thank to an iterative algorithm. In order to do so, a MFPCA on the whole set of curves is performed, weighting each curve with it posterior probability to belong to the cluster $t_{ik}^{(q)}$.

4.2 Estimation of intrinsic dimensionalities

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 break 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 BIC, AIC, ICL or slope heuristic (described below). This estimation of number of intrinsic dimensions is done in the M step of EM algorithm. It may allow those dimensions to vary along iterations in order to fit well data.

4.3 Choice of 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 are Akaike information criterion (AIC, Akaike (1974)),

$$AIC = l(\hat{\theta}) - m,$$

and Bayesian information criterion (BIC, Schwarz (1978)),

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

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

The Integrated completed likelihood (ICL, Biernacki et al (2000)) criterion can also be used in the aim of selecting the number of clusters with

$$ICL = BIC - \sum_{k=1}^{K} \sum_{i=1}^{n} z_{ik} \times log(z_{ik}).$$

This criterion, unlike BIC, uses the observations and the group allocations to make the decision concerning the model to be selected. In comparison with BIC, it tends to choose a model with more separated clusters.

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:

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

where s is the slope of the linear part of the maximum log-likelihood value $l(\hat{\theta})$ when plotted against the model complexity. This method however requires to test a large number of clusters or a large number of models.

In the rest of this paper, we will use BIC which is commonly used in model selection and the slope heuristic because it usually gives good results in practical situations for the selection of the number of clusters. BIC and the slope heuristic will be compared in the Section 5.

5 Numerical experimentation on simulated data

This section presents numerical experiments on simulated data in order to illustrate the behavior of the proposed methodology and confront it to competitors of the literature. The R code (R Core Team, 2017) for our multivariate functional clustering algorithm is available on CRAN in the funHDDC package.

5.1 Simulation setup

We consider 3 simulation scenarios designed as follows.

Scenario A For this first scenario, a sample of 1000 bivariate curves are simulated based on $[a_k b_k Q_k D_k]$ model. In order to do that, scores are simulated according 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 2.2. 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. Figure 1 top).

Scenario B The second simulation setting is inspired by the data simulation process of Ferraty and Vieu (2003); Preda (2007); Bouveyron et al (2015). For this simulation study, the number of clusters is fixed to $K = 4$. A sample of 1000 bivariate curves is simulated according to the following model for

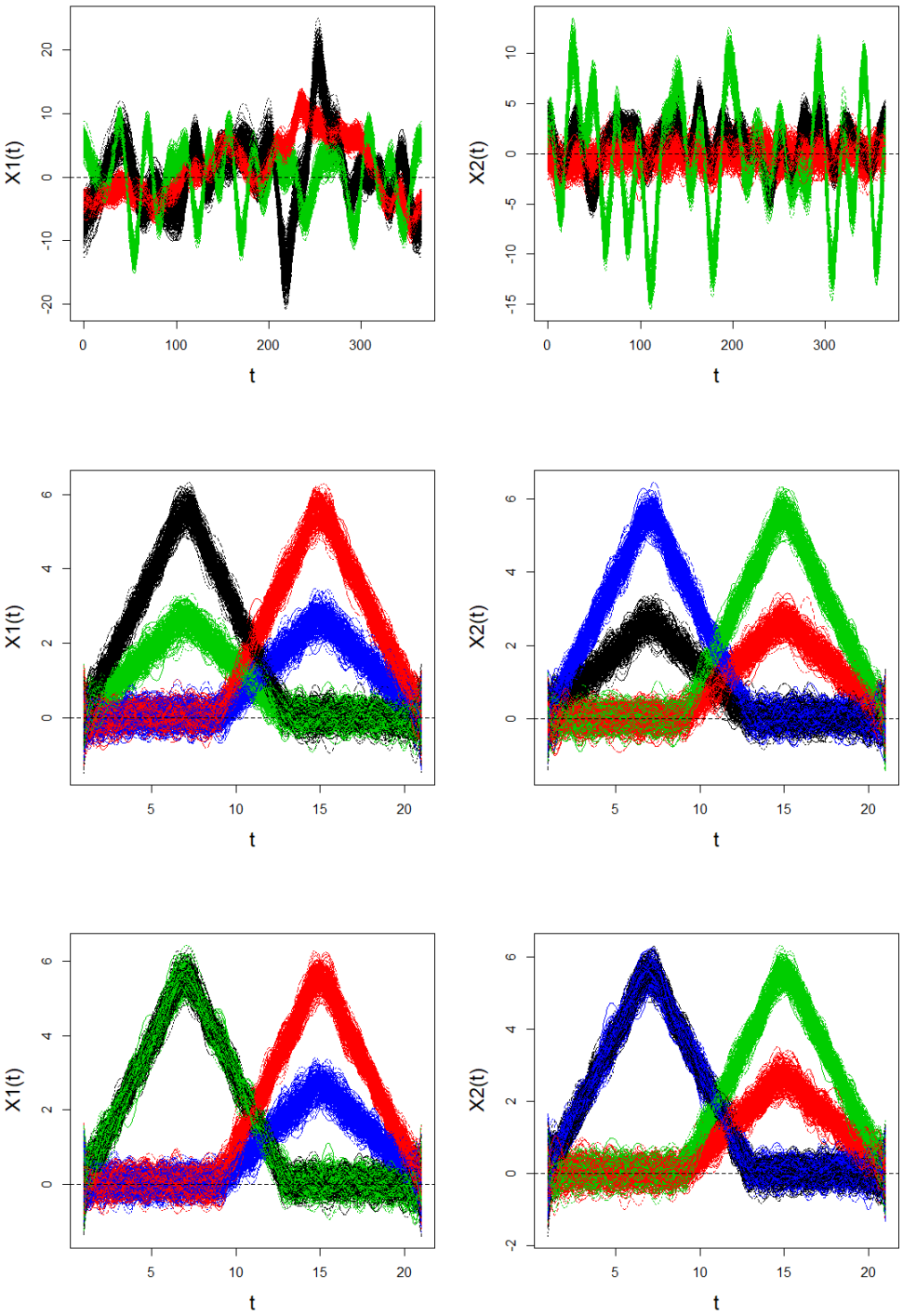


Fig. 1 Smooth data simulated for variable 1 (left) and variable 2 (right) for scenario A (top), scenario B (middle) and scenario C (bottom) colored by group for one simulation

$t \in [1, 21]$:

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

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 Bspline basis smoothing with 25 basis functions (cf. Figure 1 middle).

Scenario C For this third 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 Bsplines 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)). 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 choosen using BIC or slope heuristic), the stopping criterion for the EM algorithm is a growth of the log-likelihood lower than 10^{-3} or a maximal number of iterations of 200, the initialization of the algorithm is done with a *random* partition for scenario B, C, and A in the introductory example, and a *kmeans* partition for scenario A in the model selection and the benchmark, in order to speed up the convergence.

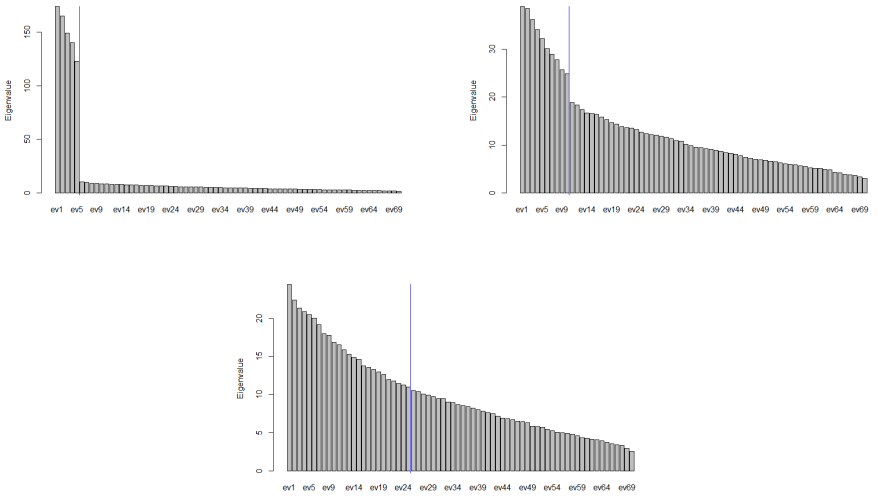
5.2 Introductory example

In order to illustrate the good behavior of the inference algorithm, we want to prove that we can recover parameters we fixed in simulated data. The easiest way to do that is to generate scores with fixed parameters a, b and d from 2 functional variables, reconstruct the coefficients of the corresponding variable and see if the algorithm can get back to the fixed parameters. So, data are generated according to *Scenario A*. The algorithm is applied for $K = 3$ groups with all 6 submodels and the simulation setting is repeated 50 times.
The quality of the estimated partitions are summarized by the ARI given in Table 1. As expected, the best results are obtained for the model $[a_k b_k Q_k D_k]$ which is used to generate data. The other constrained models followed with an ARI a bit smaller.

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

Method	Model	Mean (SD)
funHDDC	$[a_{kj} b_k Q_k D_k]$	0.99 (0.08)
funHDDC	$[a_{kj} b Q_k D_k]$	0.85 (0.26)
funHDDC	$[a_k b_k Q_k D_k]$	**1 (0)**
funHDDC	$[a_k b Q_k D_k]$	0.88 (0.26)
funHDDC	$[a b_k Q_k D_k]$	0.95 (0.16)
funHDDC	$[a b Q_k D_k]$	0.49 (0.36)

We can also note that the $[a_k b_k Q_k D_k]$ model get back parameters a, b and d that we use to generate data (cf. Figure 2). d_k is chosen for each group thanks to Cattell's Scree-test (with a threshold of 0.2), and the true dimensions are found for 2 clusters out of 3. The model has some difficulties to get back to the true dimensions of the third group, it can be explained by the low signal/noise ratio. Parameters a and b are also very close to the one we chose.



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

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

5.3 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 submodels have been estimated for a number of clusters from 2 to 10.

Figure 3 and Figure 4 show for one simulation with the model $[a_k b_k Q_k D_k]$, the values of BIC and slope heuristic. For this simulation both slope heuristic and BIC succeed in selecting the right number of clusters. The slope heuristic left plot corresponds to the log-likelihood function with regard to the number of free model parameters. The red line is estimated using a robust linear regression and its coefficient is used to compute the penalized log-likelihood function shown on the right plot.

Table 2 summarized the results of the 50 simulations for the BIC and the slope heuristic. The BIC criterion has some difficulties to estimate the true 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 re-

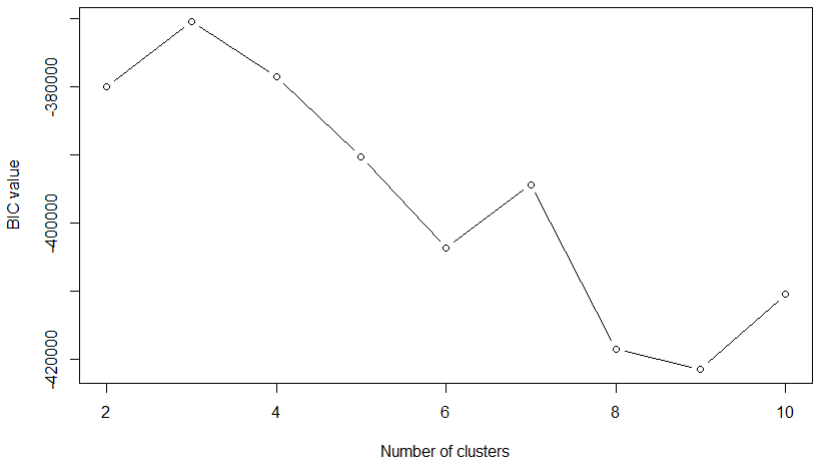


Fig. 3 BIC for one simulation for the model $[a_k b_k Q_k D_k]$

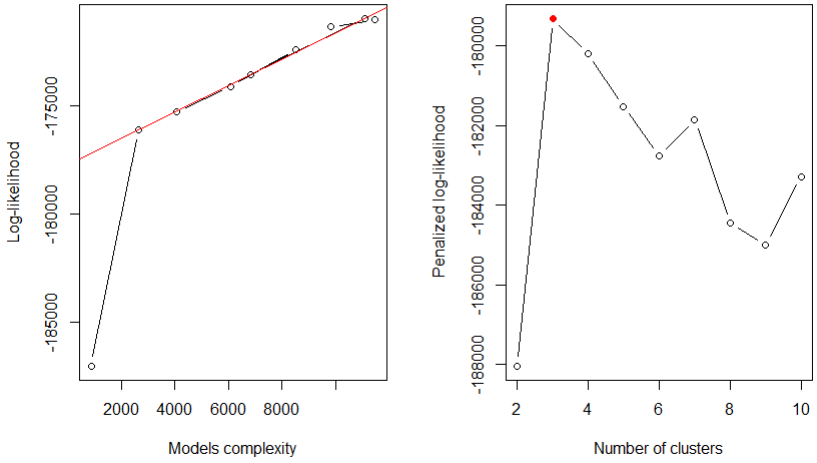


Fig. 4 Slope heuristic for one simulation for the model $[a_k b_k Q_k D_k]$

Table 2 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	-

cover the actual number of groups, in about 66% of simulations in the case of $[a_kb_kQ_kD_k]$ model.

5.4 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)). These methods are compared on the basis of the 3 simulation settings and according to the adjusted Rand index.

Table 3 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, and FGRC for 2 out of 3. FGRC 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 to model selection. One can also wonder if this counter performance of the SH is not linked to the small number of models tested.

Table 3 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)

6 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 that they could help cities in designing their policy against pollution. Let us remind that pollution kills at least nine million people and costs trillions of dollars every year, according to the most comprehensive global analyses to date.

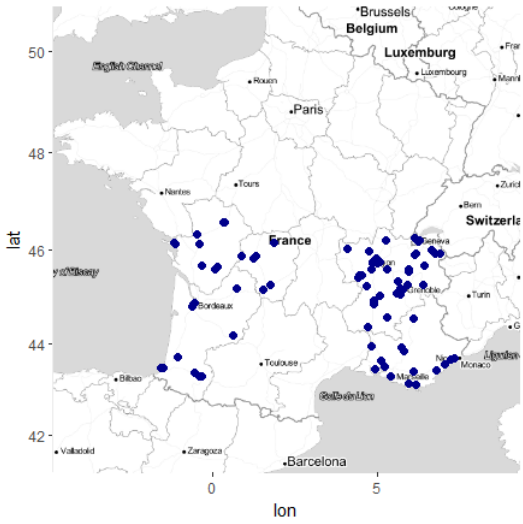


Fig. 5 Location of measured cities (dark blue)

6.1 Data

This dataset deals with pollution in some French cities (available on 3 different websites [1]). 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 and PM10 particles measured in 84 South of France 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, data are cut daily and we kept all days that have less than 4 missing values, and for which there is no missing values at the beginning or at the end of the period.

The functional form of the data is reconstructed using a cubic Bspline 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 not enough models to use the slope heuristic criteria.

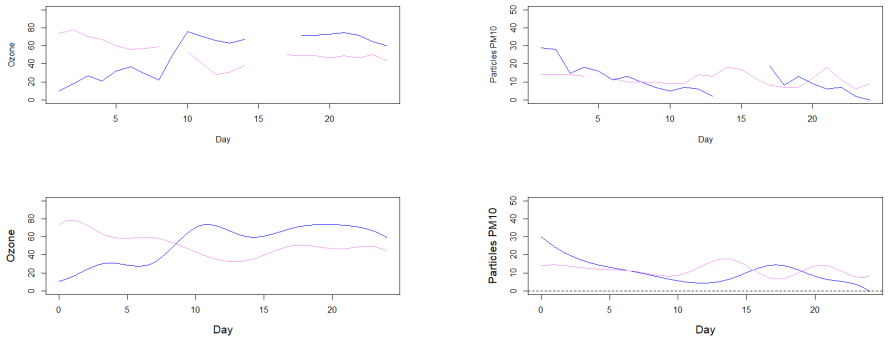


Fig. 6 Pollutants real curves (top) and smooth curves (bottom) for Avignon day 12 (blue) and La Rochelle day 177 (pink)

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

6.2 Results

According to BIC, the best partition for $[a_{kj}bQ_kD_k]$ model is with 6 clusters (cf. Table 4).

Table 4 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

The obtained groups can be described with their mean curves (cf. Figure 7 for Ozone and Figure 8 for particles PM10). 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 5). Ozone is a product of photochemical reaction between various pollutants when there is a lot of sunshine. The low duration of sunshine during winter can explain those low values. But this group gathers days the most contaminated by particles PM10 (with the higher concentration along the day). For that matter, the European Union recommend 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. The turquoise group is different from the dark blue one with lower values of PM10 (cf. Figure 8). It gathers fall and winter days (cf. Figure 9). The black group has the highest values of Ozone (cf. Figure 7). It maximum is reached between 5pm and 10pm. This can be due to exhaust gas when people commute from their work to their home.

Table 5 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	

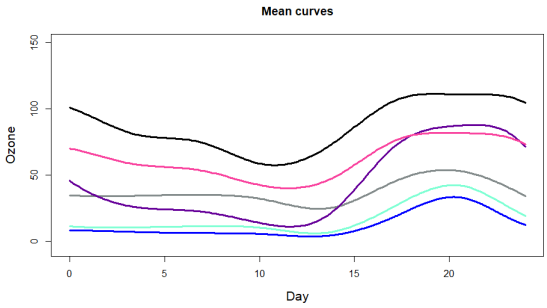


Fig. 7 O3 mean curves ($\mu g/m^3$) per day colored by cluster

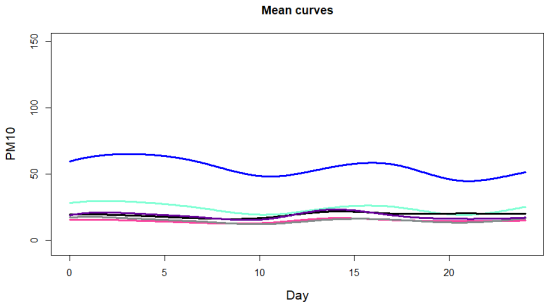


Fig. 8 Particles PM10 mean curves ($\mu g/m^3$) per day colored by cluster

We can also see a common pattern between groups. For the particles PM10, 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 pattern in the O3 curves. During a day, the Ozone concentration has a tendency to be decreasing from midnight to midday and increasing until reaching a plateau between 5pm and 8pm for the first pattern. For the second one, the Ozone concentration is stable from midnight to 2pm and increases until reaching a maximum at 8 pm.

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 highlight 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 so are not always representative of the pollution of the whole in which they are located.

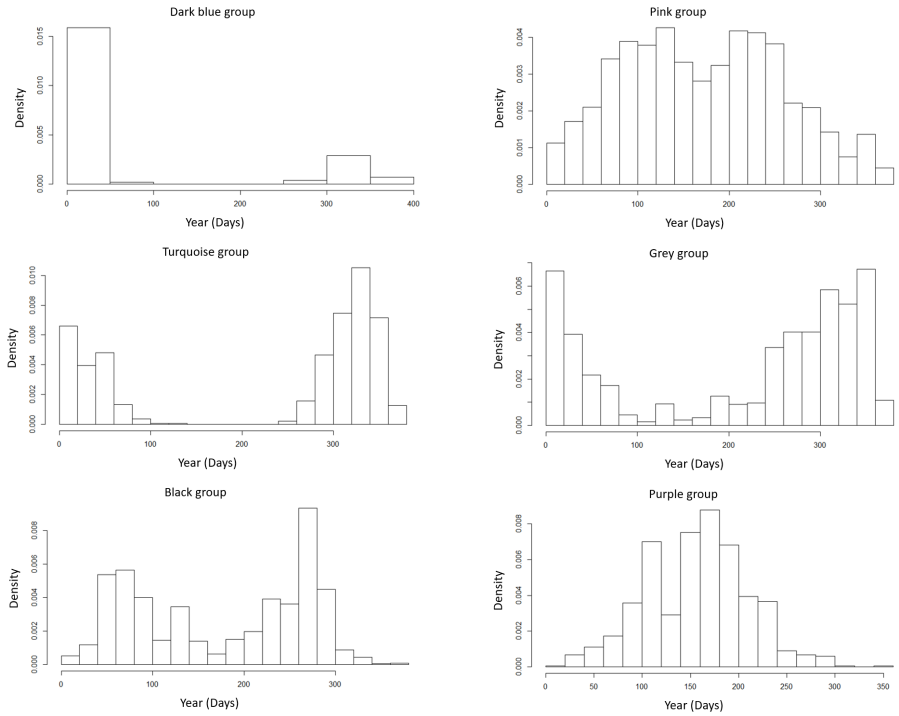


Fig. 9 Histogram of days in each group, from 1/01/2017 (0) to 31/12/2010 (364). Spring is from day 79 to 171, Summer from day 172 to 264, Fall from day 265 to 354 and Winter from day 355 to 365 and day 1 to 78.

7 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 proved on simulated datasets and the proposed technique outperforms *state-of-the-art* methods for 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 successfully applied to analyze one-year pollution records in 84 cities in France. It is worth noticing that smoothing data with basis functions allows to both filter the level of information one want 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 Bspline smoothing. It can be the subject of future work because a new model will have to be adapted to this smoothing.

Acknowledgements 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)

A Appendix: proofs

A.1 Proof of complete log-likelihood, Equation (7)

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

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 Cost function, Equation (9)

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

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 update formula of model parameters

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 loglikelihood can be written:

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

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

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

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

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

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, 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

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