Clustering multivariate functional data in group-specific functional subspaces

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

Clustering multivariate functional data in group-specific functional subspaces

Amandine Schmutz[a,b,d,*], Julien Jacques[b], Charles Bouveyron[c], Laurence Chèze[d], Pauline Martin[a]

[a] *Lim France, Chemin Fontaine de Fanny, Nontron, France*
[b] *Université de Lyon, Lyon 2, ERIC EA3083, Lyon, France*
[c] *Université Côte d'Azur, LJAD & Asclepios, Nice, France*
[d] *Université de Lyon, Lyon 1, LBMC UMR T9406, Lyon, France*

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 those 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-like 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 work. This algorithm is then applied for analyzing the pollution in U.S. cities for one year.

Keywords: Multivariate functional data, multivariate functional principal component analysis, model-based clustering, EM-algorithm.

*Corresponding author. aschmutz@lim-group.com

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 performance 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. 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 a same individual, like 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 $\mathbf{X}(t) = (X^1(t), ..., X^p(t))' \in \mathbb{R}^p, p \geq 2$. See 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 individualized predictions for example. Even though there exists numerous works for the clustering of univariate functional data (James and Sugar, 2003; Tarpey and Kinateder, 2003; Chiou and Li, 2007; Bouveyron and Jacques, 2011; Jacques and Preda, 2013; Bouveyron et al., 2015), only few methods exist for clustering multivariate functional data. Singhal and Seborg (2005) and Ieva et al. (2012) 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 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 (2014) 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. First, 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 given part of the information available is ignored. Second, the use of only a given number of principal components leads to an EM-like algorithm in which the quantity to maximize (a pseudo-likelihood) is not necessary increasing along with the iterations. In this paper, we propose to overcome these limitations by modeling all the non-null principal components. The resulting model can be viewed as an extension of Bouveyron and Jacques (2011) method, which adapts a parsimonious modeling of the eigenvalues scree of the group, to the multivariate case.

The paper is organized as follows. Section 2 presents principal component analysis for multivariate functional data as introduced in Jacques and Preda (2014). In section 3, we introduce the methodology of the clustering for multivariate functional data. Section 4 discusses the parameter estimation via an EM-like 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 Section 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 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 (2014) suggest a Multivariate Functional Principal Component Analysis (MFPCA) method with a normalization step if the units of measurement differ between functional variables. Chiou et al. (2014) present also a normalized MFPCA 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 (2014), it leads to a single set of scores for each individual. Both methods present a MFPCA model based on a normalized covariance operator. In Chen and Jiang (2016), a MFPCA that can handle data with different sampling schemes is proposed but their algorithm is applied only on univariate functional data. The method of Happ and Greven (2015) can deal with data from two-dimensional interval. This method can be applied to sparse functional data and data with measurement error because the MFPCA incorporates weights for the elements if they differ in domain, range or variation. This method includes the MFPCA proposed by Jacques and Preda (2014) in the case of data expressed in arbitrary basis expansions. In the present work, the MFPCA proposed by Jacques and Preda (2014) is used in combination with a fine probabilistic modeling of the group-specific densities. This method is therefore summarized below.

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:

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

With $i \in \{1, ..., n\}$, $j \in \{1, ..., p\}$, R_j the number of bases chosen and $(\phi_r^j(t))_{1\leq r\leq R_j}$ the basis of functions for the j-th component of the multivariate curve. The coefficients c_{ir}^j can be gathered in a matrix:

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

The matrix of basis functions $(\phi_r^j)_{1\leq r\leq R_j}$ can be concatenated into $\boldsymbol{\phi(t)}$:

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

So with these notations Equation 1 can be written in a matrix form: $\boldsymbol{X}(t) = \boldsymbol{C\phi}'(t)$.

2.2. Functional Principal Component Analysis for Multivariate functional data

The principle of the Multivariate Functional Principal Component Analysis can be summarized by finding the eigenvalues and eigenfunctions that solve the usual spectral decomposition of the covariance function:

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

with λ_l a set of positive eigenvalues, $\boldsymbol{f_l}$ multivariate eigenfunctions and ν the covariance operator. 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 f_l belongs to the linear space spanned by the basis $\boldsymbol{\phi}$:

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

with $\boldsymbol{b}_l = (b_{l11}, ..., b_{lm_1}, b_{l21}, ..., b_{l2m_2}, ..., b_{lp_1}, ..., b_{lpm_p})$.

Using the estimation 3 of ν, the eigen problem 2 becomes:

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

with $\boldsymbol{W} = \int_0^T \boldsymbol{\phi}'(t)\boldsymbol{\phi}(t)$ a matrix of dimension $n \times (\sum_{j=1}^p R_j)$ which contains the inner products between the basis function. The multivariate functional principal component analysis 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 $(\boldsymbol{f}_l)_{l\geq 1}$. 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 $\sum_{j=1}^p R_j = R$. Consequently, each $\boldsymbol{X}_i$ is identified by $\boldsymbol{\delta}_i = (\delta_{il})_{1\leq l\leq R}$.

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). Thus, $n_k = \sum_{i=1}^n z_{ik}$ is the number of curves within cluster k.

Let suppose that these curves can be described into a low-dimensional functional latent subspace with intrinsic dimensions $d_k \leq R$, $k = 1, ..., K$. The group-specific basis is obtained from $\{\phi_r^j\}_{(1\leq j\leq p),(1\leq r\leq R_j)}$ through a linear transformation: $\varphi_{kj}(t) = \sum_{l=1}^{R} q_{k,jl}\phi(t)$ with $Q_k = (q_{k,jl}) = [U_k, V_k]$ the orthogonal $R\times R$ matrix of eigenfunction coefficients. Q_k is split for later use into two parts: U_k of size $R \times d_k$ and V_k of size $R \times (R - d_k)$ with $U_k'U_k = I_{d_k}$, $V_k'V_k = I_{R-d_k}$ 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 $\boldsymbol{\delta}_i^k \sim \mathbb{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ with $\boldsymbol{\mu}_k \in \mathbb{R}^R$ the mean function and $\boldsymbol{\Sigma}_k = \boldsymbol{Q_k\Delta_kQ_k^t}$ the variance matrix. Let assume moreover, for parsimony reasons, that $\boldsymbol{\Delta}_k$ can be written:

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

This assumption on $\boldsymbol{\Delta}_k$ allows to finely model the variance of the first d_k principal components only, the remaining ones being 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.

Different submodels can be defined depending on the constraints we could apply on model parameters, within or between groups, leading to more parsimonious submodels. This possibility allows to fit onto 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. $[a_k b Q_k d_k]$: the parameters a_k are fixed to be common within each group and b_k are fixed to be common between groups. $[a b_k Q_k d_k]$: the parameters a_{kj} are fixed to be common between and within groups. $[a b Q_k d_k]$: 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 estimation and choice of the number of clusters

4.1. EM-based 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 expectation of the completed log-likelihood using the current estimate of parameters; and a Maximisation step, which computes parameters maximizing the expected completed log-likelihood found at the E step. This section presents the update formula of the EM algorithm in the case of the $[a_{kj} b_k Q_k d_k]$ model.

The complete log-likelihood of the observed curves under the $[a_{kj} b_k Q_k d_k]$ model has the following form:

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

where $\theta = (\pi_k, \mu_k, \delta_{kj}, b_k, q_{kj})_{kj}$ for $1 \leq j \leq d_k$ and $1 \leq k \leq K$, q_{kj} is the jth column of Q_k and $C_k = \frac{1}{n_k}\sum_{i=1}^{n} Z_{ik}(\delta_i - \mu_k)^t(\delta_i - \mu_k)$, $\delta_1, ..., \delta_n$ are the scores

of the observed curves $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ in their subspace. As the group memberships Z_{ik} are unknown, it is necessary to compute their conditional expectation (*E step*) before to maximize the expected complete likelihood (*M step*).

E step. This step computes the posterior probability to belong to the kth cluster for each curve:

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

with $H_k^{(q-1)}(\delta)$ defined for $\delta \in \mathbb{R}^R$ as:

$$\begin{aligned} H_k^{(q-1)}(\delta) &= ||\mu_k^{(q-1)} - P_k(\delta)||_{d_k}^2 + \frac{1}{b_k^{(q-1)}}||\delta - P_k(\delta)||^2 \\ &\quad + \sum_{j=1}^{d_k} \log(a_{kj}^{(q-1)}) + (R - d_k)\log(b_k^{(q-1)}) - 2\log(\pi_k^{(q-1)}) \quad (8) \end{aligned}$$

where $||.||_{\mathcal{D}_k}^2$ is a norm on the latent space $\mathbb{E}_k$ defined by $||y||_{\mathcal{D}_k}^2 = y^t \mathcal{D}_k y$, $\mathcal{D}_k = \tilde{Q}\Delta_k^{-1}\tilde{Q}^t$ 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 $P_k(\delta) = WU_kU_k^tW^t(\delta - \mu_k) + \mu_k$.

M step. This step estimates the model parameters by maximizing the expectation of the complete likelihood conditionally on the posterior probabilities $t_{ik}^{(q)}$ computed in the previous step. Mixture proportions and means are updated by:

$$\pi_k^{(q)} = \frac{\eta_k^{(q)}}{n}, \;\; \mu_k^{(q)} = \frac{1}{\eta_k^{(q)}} \sum_{i=1}^{n} t_{ik}^{(q)} \delta_i \qquad (9)$$

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

Let us also introduce $C_k^{(q)} = \frac{1}{\eta_k^{(q)}} \sum_{i=1}^{n} t_{ik}^{(q)} (\delta_i - \mu_k^{(q)})(\delta_i - \mu_k^{(q)})^t$, 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} are, in the case of the $[a_{kj}b_kQ_kd_k]$, for $k = 1, ..., K$:

- the d_k first columns of the orientation matrix Q_k are updated by the coefficients of eigenfunctions associated with the largest eigenvalues of $W^{1/2}C_k^{(q)}W^{1/2}$,

- the variance parameters a_{kj}, $j = 1, ..., d_k$, are updated by the d_k largest eigenvalues of $W^{1/2}C_k^{(q)}W^{1/2}$,
- the variance parameters b_k are updated by $b_k^{(q)} = \frac{1}{R-d_j}[tr(W^{1/2}C_k^{(q)}W^{1/2}) - \sum_{j=1}^{d_k} \widehat{\delta}_{kj}^{(q)}]$.

To summarize, the algorithm for multivariate functional data clusters multivariate functional data through their projection into low dimensional subspaces. Those projections are obtained by perfoming a multivariate functional principal component analysis per cluster, each functional data being weighted in the MFPCA by the posterior probability to belong to the cluster $t_{ik}^{(q)}$.

4.2. Choice of number of clusters and number of intrinsic dimensions

We now focus on the choice of the hyper-parameters K and d_k. The choice of these hyper-parameters is here viewed as model selection problems. Classical model selection tools are Akaike information criterion (AIC) (Akaike (1974)),

$$AIC = L(\theta) - m,$$

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

$$BIC = L(\theta) - \frac{m}{2} \times log(n),$$

with $L(\theta)$ the log-likelihood, 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 allocations to make the decision concerning the model to be selected. Contrary to BIC, it tends to choose groups more separated. An other method, that has proved its usefulness, is the slope heuristic (Birge and Massart (2007)). This data-driven method provides a criterion whose penalty is known up to a multiplicative factor which is provided by the slope heuristic:

$$SH = L(\theta) - 2sm,$$

where s is the slope of the linear part of $L(\theta)$ and m the number of model parameters. This method however requires to test a large number of clusters or a large number of models.

In order to choose the intrinsic dimensions d_k the Cattell's scree-test (Cattell (1966)) is used. This test can be used to select the number of dimensions of the MFPCA, looking 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 one of the criterion described above.

In the rest of this paper, we will use the BIC which is commonly used in model selection and the slope heuristic because it usually gives good results in practical situations. BIC and 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:

1. illustrate the behavior of the model (Section 5.2),
2. compare the efficiency of the BIC and slope heuristic criteria for selecting the number of clusters (Section 5.3), and
3. confront the proposed algorithm to competitors of the literature: Funclust (Jacques and Preda (2014)), kmeans-d1 and kmeans-d2 (Ieva et al. (2012)) (Section 5.4).

The R code (R Core Team (2017)) for multivariate functional clustering is available on request from the authors, and the release of an R package is planned.

5.1. Simulation set up

Scenario A. For this first scenario, the number of clusters is fixed to $K = 3$. A sample of 300 bivariate curves is simulated according to the following model for

$t \in [0,1]$:

$$\begin{aligned}
\text{Group 1} : X_1(t) &= sin((10+\gamma_1)t) + (1+\gamma_1) + \epsilon_1(t),\\
X_2(t) &= sin((5+\gamma_2)t) + (0.5+\gamma_2) + \epsilon_2(t),\\
\text{Group 2} : X_1(t) &= sin((5+\gamma_2)t) + (0.5+\gamma_2) + \epsilon_2(t),\\
X_2(t) &= sin((15+\gamma_1)t) + (1+\gamma_1) + \epsilon_1(t),\\
\text{Group 3} : X_1(t) &= sin((15+\gamma_1)t) + (1+\gamma_1) + \epsilon_1(t),\\
X_2(t) &= sin((10+\gamma_1)t) + (1+\gamma_1) + \epsilon_1(t),
\end{aligned}$$

where $\epsilon_1(t)$ is a white noise of variance $\gamma_1/2$, $\epsilon_2(t)$ is a white noise of variance $\gamma_2/2$, $\gamma_1 \sim \mathbb{N}(0, 0.2)$ and $\gamma_2 \sim \mathbb{N}(0, 0.3)$. The mixing proportions are equal and the curves are observed in 100 equidistant points. Curves are smoothed using a basis of 20 cubic Bsplines functions (cf. Figure 1).

Scenario B. The second simulation setting is inspired by the data simulation process of Ferraty and Vieu (2003); Preda (2007); Jacques and Preda (2014). For this simulation study, the number of clusters is fixed to $K = 4$. A sample of 1000 curves is simulated according to the following model for $t \in [1, 21]$:

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

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).

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

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

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, 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)) computed with *adjust-*

edRandIndex function from mclust package (Fraley and Raftery (2002)).

The settings of the algorithm 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 initialization of the algorithm is done with a *random* partition, and 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.

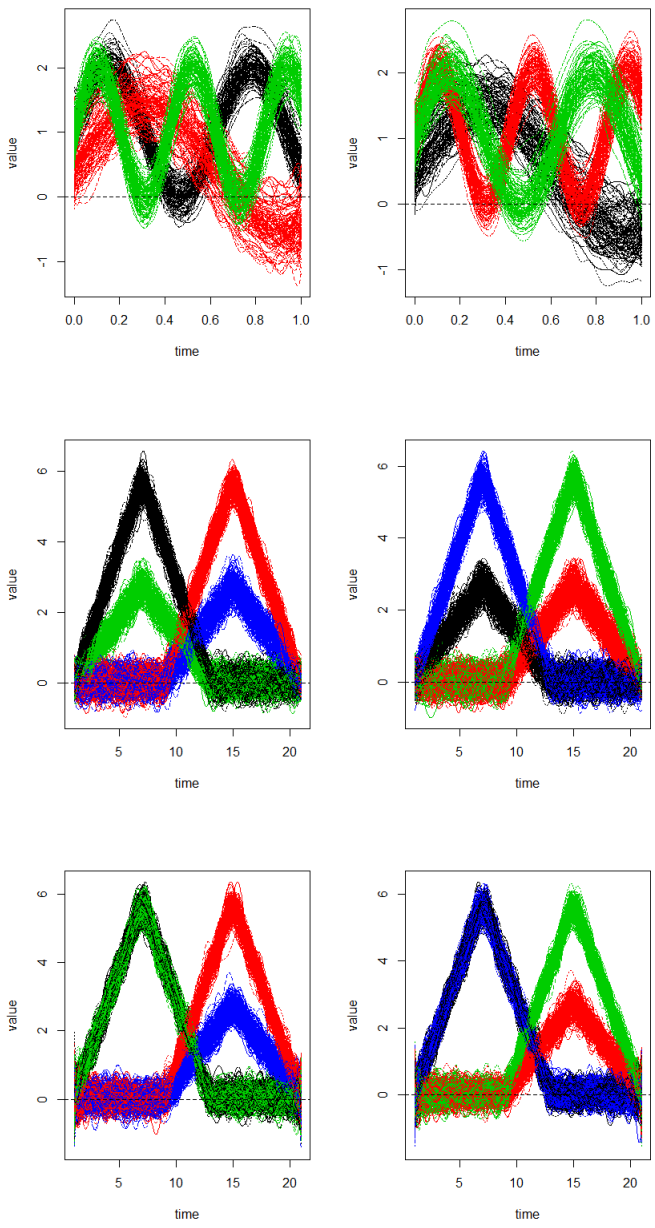

Figure 1: Data simulated for scenario A (top), scenario B (middle) and scenario C (bottom) colored by group for one simulation

5.2. Introductory example

In order to illustrate the behavior of the inference algorithm, data are generated according to the *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. The best results are obtained for the model $[a_k b_k Q_k D_k]$ and for the model $[a_k b Q_k D_k]$, which is a model close to $[a_k b_k Q_k D_k]$ but less flexible. This second model considers that the noise is common between groups, which is an acceptable hypothesis because data have been obtained in the same acquisition process. The other constrained models followed with an ARI a bit smaller than those of the first two models. Moreover, the evolution of the log-likelihood function along with the iteration is represented in Figure 2 for the best model according to ARI. We note in that example that the log-likelihood quickly converges (after 4 iterations only), since the clusters are well separeted. The number of iterations depends on the model used and the initialization chosen (not presented here). Usually the convergence is accelerated by a *kmeans* initialization.

Model	Mean(SD)
$[a_{kj} b_k Q_k D_k]$	0.97(0.13)
$[a_{kj} b Q_k D_k]$	0.96(0.15)
$[a_k b_k Q_k D_k]$	0.98(0.11)
$[a_k b Q_k D_k]$	0.98(0.11)
$[a b_k Q_k D_k]$	0.96(0.15)
$[a b Q_k D_k]$	0.90(0.22)

Table 1: Mean (and s.d.) of ARI for 50 simulations with random initialization

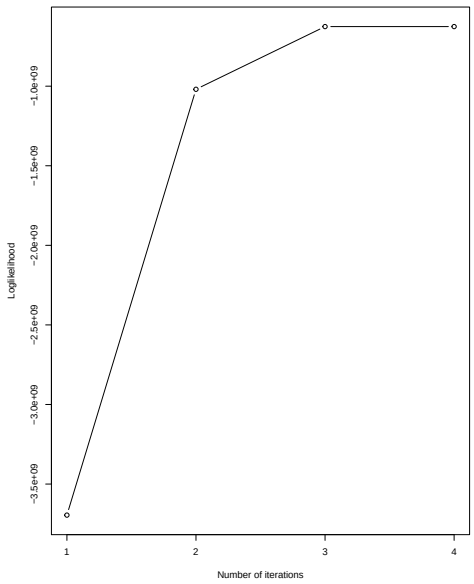

Figure 2: Convergence of the log-likelihood for one simulation for the model $[a_k b Q_k D_k]$

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.

The results obtained are summarized in Table 2 for the BIC criterion and in Table 3 for the slope heuristic. The BIC criteria tends to overestimate K. Indeed, depending on the simulation, BIC selects between 5 or 6 clusters instead of 3. The slope heuristic is conversely more efficient to recover the actual number of groups, in about 74% of simulations.

	Number K of clusters								
Model	2	3	4	5	6	7	8	9	10
$[a_{kj}b_kQ_kD_k]$	-	-	-	26	21	2	-	-	-
$[a_{kj}bQ_kD_k]$	-	-	3	14	20	6	-	-	-
$[a_kb_kQ_kD_k]$	-	-	1	17	21	3	-	-	-
$[a_kbQ_kD_k]$	-	-	1	21	16	6	-	-	-
$[ab_kQ_kD_k]$	-	-	-	13	26	4	-	-	-
$[abQ_kD_k]$	-	-	-	18	21	7	-	-	-

Table 2: Best model selected by BIC for 50 simulations

	Number K of clusters								
Model	2	3	4	5	6	7	8	9	10
$[a_{kj}b_kQ_kD_k]$	3	45	1	-	-	-	-	-	-
$[a_{kj}bQ_kD_k]$	3	35	4	1	-	-	-	-	-
$[a_kb_kQ_kD_k]$	4	37	1	-	-	-	-	-	-
$[a_kbQ_kD_k]$	1	42	1	-	-	-	-	-	-
$[ab_kQ_kD_k]$	2	40	1	-	-	-	-	-	-
$[abQ_kD_k]$	4	41	1	-	-	-	-	-	-

Table 3: Best model selected by heuristic slope for 50 simulations

Figure 3 shows for one simulation the values of BIC and slope heuristic for the model $[a_kb_kQ_kD_k]$ (the best model according to ARI). On this simulation the slope heuristic succeeds in selecting the right number of clusters but the BIC doesn't. The slope heuristic plot on the left 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 panel.

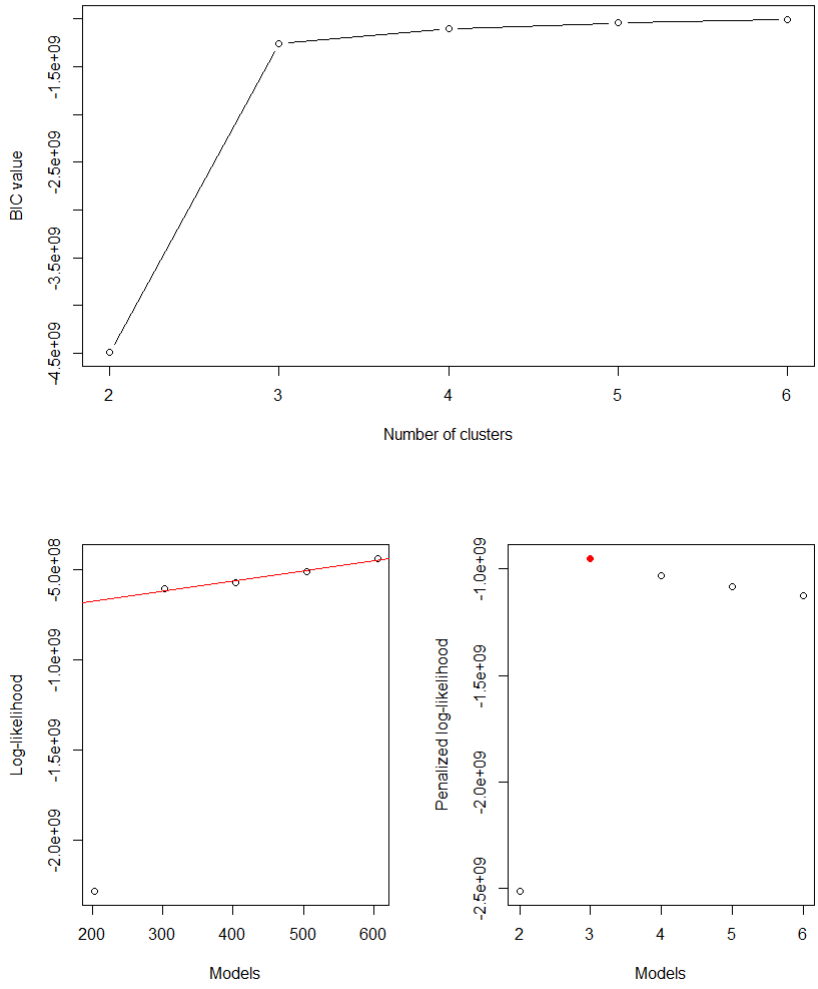

Figure 3: BIC (top) and slope heuristic (bottom) for one simulation for the model $[a_k b_k Q_k D_k]$

5.4. Benchmark with existing methods

In this section the proposed clustering algorithm is compared to competitors of the literature: Funclust (Jacques and Preda (2014), with the R package funclustering), kmeans-d1 and kmeans-d2 (Ieva et al. (2012), with our own implementation). The methods are compared on the basis of the three simulation

settings and according to the adjusted rand index.

Table 4 presents clustering accuracies for the 9 tested models. Scenario A turns out to be an easy situation since all models, except Funclust, perform well in this case. Scenario B and C are conversely harder situations and our algorithm clearly outperforms its competitors here.

Model	Scenario A	Scenario B	Scenario C
$[a_{kj}b_kQ_kD_k]$	0.90(0.21)	0.82(0.19)	0.80(0.19)
$[a_{kj}bQ_kD_k]$	0.92(0.19)	0.92(0.15)	0.78(0.21)
$[a_kb_kQ_kD_k]$	0.92(0.19)	0.90(0.17)	0.75(0.21)
$[a_kbQ_kD_k]$	0.96(0.15)	0.85(0.18)	0.78(0.18)
$[ab_kQ_kD_k]$	0.90(0.21)	0.87(0.18)	0.80(0.20)
$[abQ_kD_k]$	0.89(0.22)	0.81(0.18)	0.79(0.19)
$Funclust$	0.23(0.15)	0.36(0.25)	0.45(0.20)
$kmeans - d_1$	0.90(0.29)	0.37(0.48)	0.32(0.46)
$kmeans - d_2$	0.84(0.35)	0.30(0.43)	0.18(0.36)

Table 4: Mean (and s.d) of ARI for all tested models on 50 simulations

6. Case study: analysis of the pollution in U.S. cities

This section focuses on the analysis of pollution data in U.S. 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 analysis to date.

6.1. Data

This dataset deals with pollution in 144 U.S. cities (available on https://www.kaggle.com/sogun3/uspollution) and has been documented by the U.S. Environmental Protection Agency. It gathered 2 major pollutants: Nitrogen Dioxide and Ozone, measured daily from 2000 to 2016. For each pollutant

the arithmetic mean, the maximum value, the instant at which this maximum has been reached and the air quality index (which represents how polluted the air is according to government agencies) are recorded. This four variables are calculated four times per day. In this study we choose to work on the mean value of Nitrogen Dioxide and Ozone for each day. A sample of curves is shown in Figure 4 (top). A period of one year from 1/01/2014 to 31/12/2014 is considered. Among all the available cities, we keep the 50 cities that have less than 30 missing values in that period and for which there is no missing values at the beginning and at the end of the period. The functional form of the data is reconstructed using a cubic Bspline smoothing with 15 basis functions (cf. Figure 4, bottom). Our algorithm has been applied on smoothed data with a varying number of clusters, from 2 to 8. The more general model $[a_{kj}b_kQ_kD_k]$ has been chosen, and the inference algorithm is considered with same settings than in 5.1 but with *kmeans* initialization in order to accelerate its convergence. The slope heuristic criteria is used to choose an appropriate number of clusters.

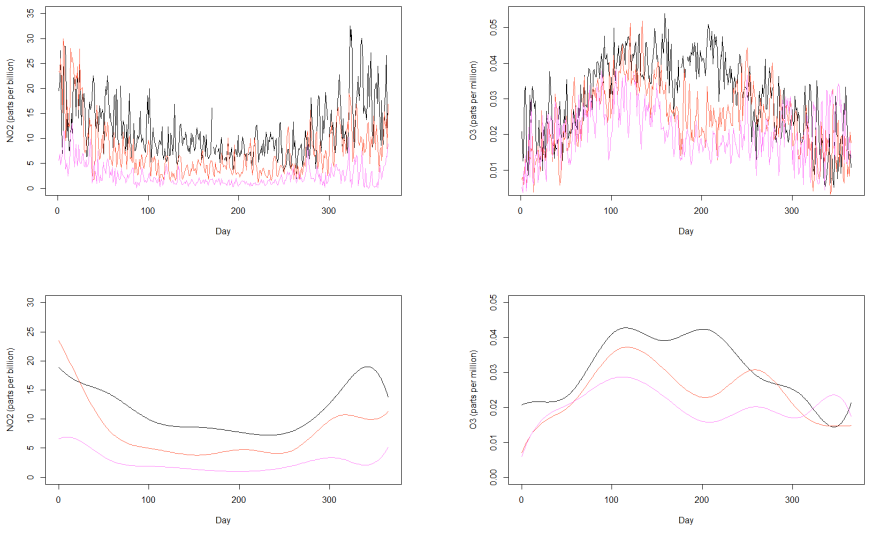

Figure 4: Pollutants real curves (top) and smooth curves (bottom) for Albuquerque (black), Concord (orange) and Eureka (pink)

6.2. Results

Figure 5 both presents the values of BIC and slope heuristic according to the number of groups. As it can be seen, the slope heuristic criterion clearly have a peak at K=4 clusters. Let remark that with *kmeans* initialization, the log-likelihood converges quickly (in 8 iterations, cf. Figure 5).

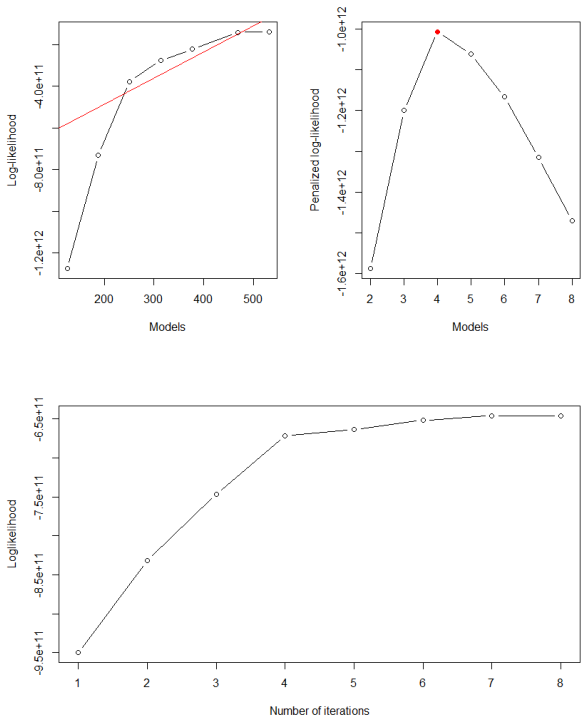

Figure 5: Slope heuristic (top) for 2 to 8 clusters and growth of log-likelihood for the solution with 4 groups (bottom)

Figure 6 plots the cities on a map of the USA, with a color depending on their clusters belonging. The obtained clusters can also be described with their mean curves(cf. Figure 7) . The light-green group has a higher mean concentration of NO2 than the other groups. This group gathers the most contaminated cities. The dark-green group is characterized by the lower NO2 concentration along the year but the higher O3 concentration in winter, spring and autumn

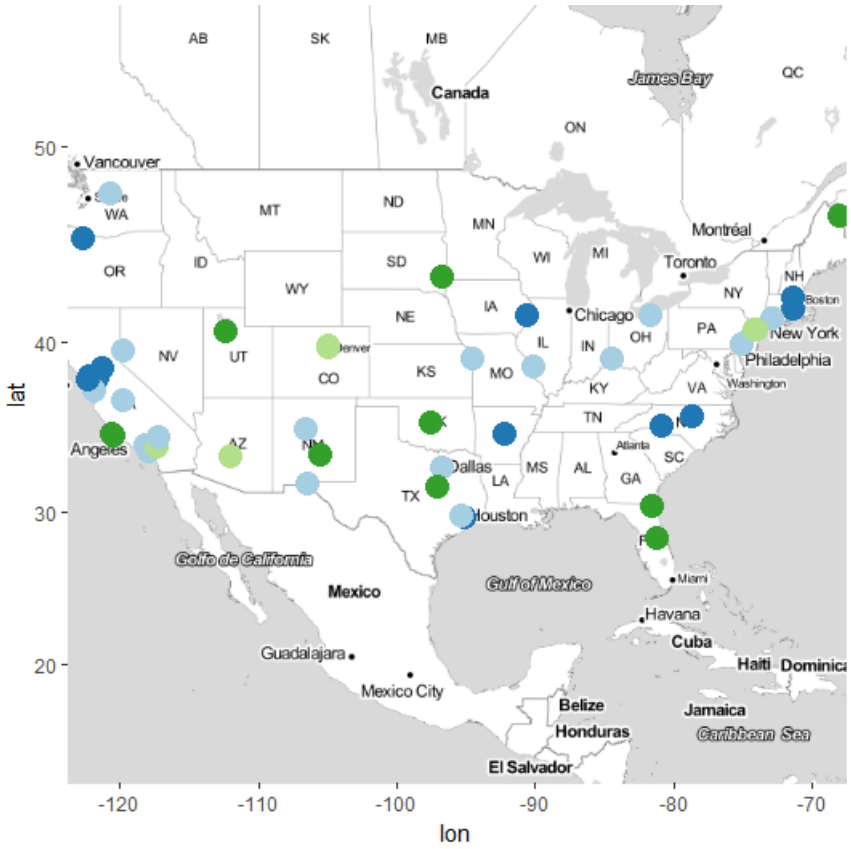

Figure 6: Map of the clustering results

(cf. Figure 8). The light-blue curves and the dark blue ones are cities with a medium pollution of NO2. Those cities have the same pattern than the light-green one but correspond to smaller cities with less population density. We can also see a common pattern between groups: the NO2 concentration is higher during the end of fall and winter for all cities. This pollutant mostly come from the combustion of fossil fuels, so it can be correlated to the increased use of heater or air-conditioning during this period. The O3 curves have a totally different pattern, it forms a hyperbole which is centered on summer. O3 is a product of photochemical reaction between various pollutants.

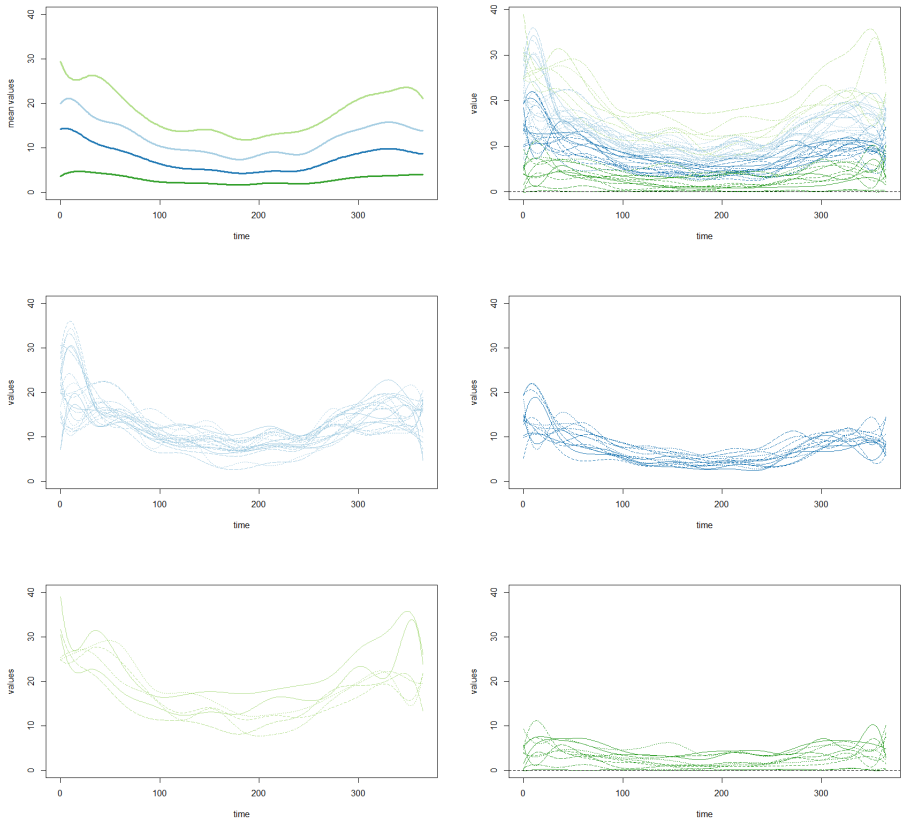

Figure 7: Mean NO2 curves (top left), total NO2 curves (top right), group 1 curves (middle left), group 2 curves (middle right), group 3 curves (bottom left) and group 4 curves (middle right)

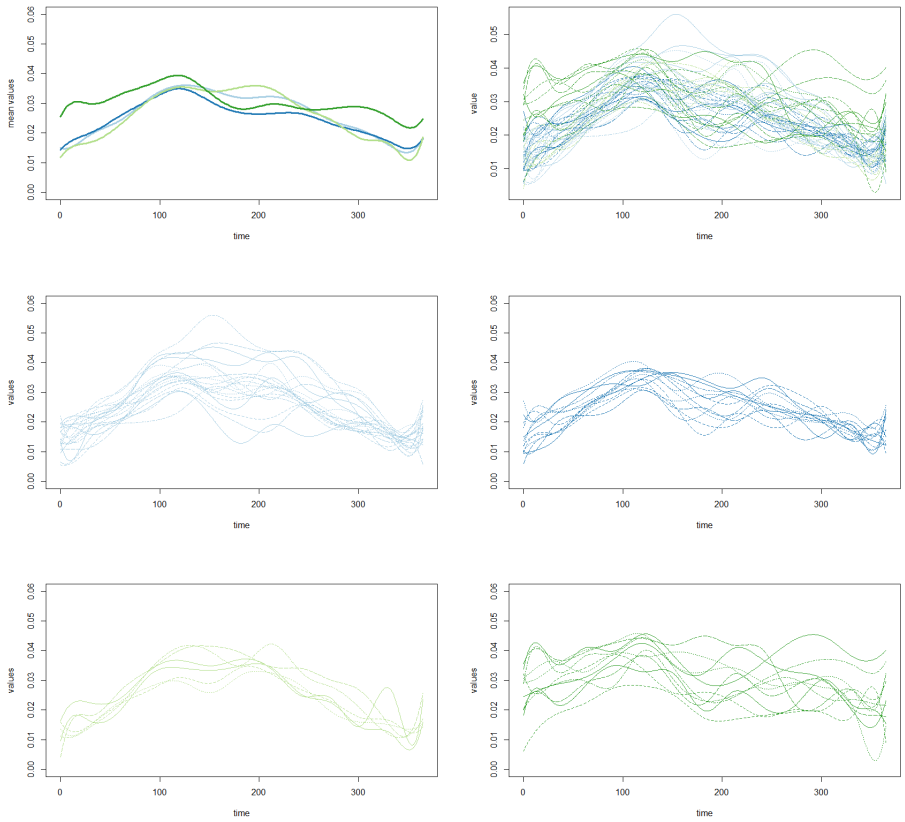

Figure 8: Mean O3 curves (top left), total O3 curves (top right), group 1 curves (middle left), group 2 curves (middle right), group 3 curves (bottom left) and group 4 curves (bottom right)

7. Discussion and conclusion

This work was motivated by the will to provide a new clustering method for multivariate functional data which takes into account the possibility that data live in subspaces of different dimensions. This 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. From the computational point of view, this novel method is faster than Funclust and as fast as kmeans on big datasets. Notice also that this new algorithm works in the univariate case as well and, therefore, generalizes the funHDDC algorithm (Bouveyron and Jacques (2011)). The proposed methodology has been successfully applied to analyze one-year pollution records in 50 cities in the USA. It is worth noticing that smoothing on basis functions allows to both filter the level of information one want to keep in the data and to deal with missing data.

Bibliography

Akaike H. A new look at the statistical model identification. IEEE Transactions on Automatic Control 1974;9:716–23.

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

Biernacki C, Celeux G, Govaert G. Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans PAMI 2000;22:719–25.

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

Bouveyron C, Come E, Jacques J. The discriminative functional mixture model for the analysis of bike sharing systems. Annals of Applied Statistics 2015;9(4):1726–60.

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

Cattell R. The scree test for the number of factors. Multivariate Behaviour Research 1966;1(2):245–76.

Chen L, Jiang C. Multi-dimensional functional principal component analysis. Statistics and Computing 2016;27:1181–92.

Chiou J, Chen Y, Yang Y. Multivariate functional principal component analysis: a normalization approach. Statistica Sinica 2014;24:1571–96.

Chiou JM, Li PL. Functional clustering and identifying substructures of longitudinal data. Journal of the Royal Statistical Society Series B Statistical Methodology 2007;69(4):679–99.

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

Ferraty F, Vieu P. Curves discrimination: a nonparametric approach. Computational Statistics and Data Analysis 2003;44:161–73.

Fraley C, Raftery A. Model-Based Clustering, Discriminant Analysis and Density Estimation. Journal of the American Statistical Association 2002;97:611–31.

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

Ieva F, Paganoni A, Pigoli D, Vitelli V. Multivariate Functional Clustering for the Morphological Analysis of ECG Curves. Journal of the Royal Statistical Society, Series C (Applied Statistics) 2013;62(3):401–18.

Jacques J, Preda C. Funclust: a curves clustering method using functional random variable density approximation. Neurocomputing 2013;112:164–71.

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

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

Kayano M, Dozono K, Konishi S. Functional Cluster Analysis via Orthonormalized Gaussian Basis Expansions and Its Application. Journal of Classification 2010;27:211–30.

Preda C. Regression models for functional data by reproducing kernel hilbert spaces methods. Journal of Statistical Planning and Inference 2007;137:829–40.

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

Ramsay JO, Silverman BW. Functional data analysis. 2nd ed. Springer Series in Statistics. New York: Springer, 2005.

Rand WM. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 1971;66(336):846–50.

Schwarz G. Estimating the dimension of a model. The Annals of Statistics 1978;6(2):461–4.

Singhal A, Seborg D. Clustering multivariate time-series data. Journal of Chemometrics 2005;19:427–38.

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

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

Yamamoto M. Clustering of Functional Data in a Low-Dimensional Subspace. Advances in Data Analysis and Classification 2012;6:219–47.

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

Yamamoto M, Terada Y. Functional Factorial k-Means Analysis. Computational Statistics and Data Analysis 2014;79:133–48.