Model-based Clustering of Time Series in Group-specific Functional Subspaces

Charles Bouveyron, Julien Jacques

Model-based Clustering of Time Series in Group-specific Functional Subspaces

Charles Bouveyron[1] & Julien Jacques[2]

[1]Laboratoire SAMM, University Paris I Panthéon-Sorbonne, Paris, France.
charles.bouveyron@univ-paris1.fr
[2] Laboratoire Paul Painlevé, UMR CNRS 8524 University Lille I, & INRIA Lille Nord-Europe, Lille, France.
julien.jacques@polytech-lille.fr

Abstract

This work develops a general procedure for clustering functional data which adapts the efficient clustering method HDDC, originally proposed in the multivariate context. The resulting clustering method, called funHDDC, is based on a functional latent mixture model which fits the functional data in group-specific functional subspaces. By constraining model parameters within and between groups, a family of parsimonious models is exhibited which allow to fit onto various situations. An estimation procedure based on the EM algorithm is proposed for estimating both the model parameters and the group-specific functional subspaces. Experiments on real-world datasets show that the proposed approach performs better or similarly than classical clustering methods while providing useful interpretations of the groups.
Keywords:Functional data, time series clustering, model-based clustering, group-specific functional subspaces, functional PCA.

1 Introduction

Cluster analysis consists in identifying groups of homogeneous data without using any prior knowledge on the group labels of the data. A lot of methods, from non-parametric k-means [10] or hierarchical classification to more recent probabilistic model-based clustering [2, 6], have been proposed along the years. The clustering of time series, or more generally of functions, is a difficult task since the data live in an infinite dimensional space. We refer for instance to [21] for a survey on time series clustering. Although non-parametric approaches to functional clustering, as for instance [8, 18], lead to powerful clustering algorithms, the present paper focuses on model-based clustering which have moreover interesting interpretability properties.

Unlike the finite dimensional cases, model-based methods for clustering functional data are not directly available since the notion of probability density function generally does not exist for such data [7]. Consequently, the use of model-based clustering methods on functional data consists usually in first transforming the infinite dimensional problem into a finite one and then in using a model-based clustering method designed for finite dimensional data. The expression of functions in a finite space can be carried out by either discretizing the time interval, decomposing the functions onto a basis of functions or onto some principal components resulting from a functional principal component analysis (FPCA) [16]. The discretization of the time interval is

usually straightforward since in practice the functions are already measured in a discrete scale. The functions can also be decomposed onto a basis of well-defined functions such as natural cubic splines which are very popular and enjoy some optimality properties [20]. The decomposition of the functions can be done as well through specific time series models such as ARMA or GARCH (see [9] for a clustering algorithm based on such models). Note that in the case of using functional principal components, the functions have to be also expressed in a basis of functions in order to solve the functional eigen-decomposition problem.

Unfortunately, the resulting vectors are often high-dimensional. In particular, the discretization or the decomposition of the functions onto a spline basis (for instance, 20 natural cubic splines will be used in the applications of the present paper) usually yield to high-dimensional datasets with sometimes less observations than dimensions. In such situations, model-based clustering methods suffer from numerical problems and regularized approaches have to be used. Among the regularized model-based clustering methods, we can cite the parsimonious Gaussian mixture models [2, 6], which assume specific covariance structures, mixture of probabilistic principal component analyzers (MixtPPCA, [19]) and high-dimensional data clustering (HDDC, [4]) which both assume that high-dimensional data live in group-specific subspaces. In particular, the latter method have been used with success in various application fields such as image analysis [4] or chemometry [11].

The clustering methods previously described all consist in a two-step methodology in which the functional data are first transformed into a finite dimensional vector (the *discretization* step) and then clustered. Only model-based methods have been mentioned but some non-parametric methods such as k-means could also be considered. Unfortunately, these two-step approaches do separately the discretization and the clustering steps, and this may lead to a loss of discriminative information. Recently, a new approach due to James and Sugar [12] allows the interaction between the discretization and the clustering steps by introducing a stochastic model onto the basis coefficients. This approach is announced to be particularly effective when the functional data are sparsely sampled. In the same spirit, we propose in the present paper to adapt the HDDC method to functional data in order to model and cluster the functional data in group-specific subspaces of low dimensionality. The modeling of the functions of each group in a specific subspace should, in addition to providing an interesting clustering of the data, ease the interpretation of the clustered data.

The paper is organized as follows. Section 2 presents the proposed functional latent mixture model as well as a family of parsimonious submodels and the associated maximum likelihood estimation. Section 3 first proposes an introductory example in order to highlight the main features of the proposed method. A benchmark comparison with state-of-the-art methods is also provided in Section 3 on real-world time series datasets. Finally, Section 4 provides some concluding remarks.

2 Model-based clustering in functional subspaces

This section introduces a family of latent mixture models designed for functional data which adapts the models of [4], proposed in the multivariate context. Model inference and estimation of hyper-parameters are also discussed.

2.1 The functional latent mixture model

Let us consider a set of n observed time series or curves $\{x_1,...,x_n\}$, where $x_i = \{x_i(t)\}_{t\in[0,T]}$ $(1 \leq i \leq n)$, that one wants to cluster into K homogeneous groups.

On the one hand, let us first assume that the observed curves are independent realizations of a L_2-continuous stochastic process $X = \{X(t)\}_{t\in[0,T]}$ for which the sample paths, *i.e.* the observed curves x_i, belong to $L_2[0,T]$. In practice, the functional expressions of the observed curves are not known and we only have access to the discrete observations $x_{ij} = x_i(t_{ij})$ at a finite set of times $\{t_{ij} : j = 1,\dots,m_i\}$. As explained in [1], it is thus necessary to reconstruct the functional form of the data from their discrete observations. A common way to do this is to consider that curves belong to a finite dimensional space spanned by a basis of functions (see, for example, [16]). Let us therefore consider such a basis $\{\psi_1,\dots,\psi_p\}$ and assume that the stochastic process X admits the following basis expansion:

$$X(t) = \sum_{j=1}^{p} \gamma_j \psi_j(t), \tag{1}$$

with $\gamma_j \in \mathbb{R}$ $(j = 1,\dots,p)$ and where the number p of basis functions is assumed to be known and fixed. The basis expansion of each observed curve $x_i(t) = \sum_{j=1}^{p} \gamma_{ij}\psi_j(t)$ can be estimated by an interpolation procedure, if the curves are observed without noise, or by least square smoothing, if they are observed with error.

Let also assume that there exists an unobserved variable $Z = (Z_1,\dots,Z_K) \in \{0,1\}^K$ such that z_{ik}, the values of Z_k for the curve x_i, indicates if x_i belongs to the kth group or not. The clustering task aims therefore to predict the value of Z for each observed curve x_i.

On the other hand, let us assume that there exist K functional latent subspaces $\mathbb{E}_1[0,T],...,\mathbb{E}_K[0,T]$ ($\mathbb{E}_k[0,T] \subset L_2[0,T]$ for all $k = 1,\dots,K$) where the observed curves live conditionally to their group belonging. For each observed curve x_i, let y_i be its latent representation which lives in $\mathbb{E}_k[0,T]$ if $z_{ik} = 1$. We further assume that, in each group-specific functional subspace $\mathbb{E}_k$, the latent time series y_i, such that $z_{ik} = 1$, are also sample paths of a L_2-continuous stochastic process $Y = \{Y(t)\}_{t\in[0,T]}$ admitting a basis expansion depending on the group at hand:

$$Y(t)_{|Z_k=1} = \sum_{j=1}^{d_k} \alpha_{kj}\psi_j(t),$$

where $\{\psi_j\}_{j=1,d_k}$ is the same basis of functions as in Equation (1), but with a possible reduced number of functions d_k ($d_k \leq p$), which becomes a parameter of the model.

We finally assume that Y is linked to X, conditionally to Z, through a linear transformation:

$$X_{|Z_k=1} = \mathscr{U}_k Y_{|Z_k=1} + \varepsilon_{|Z_k=1},$$

where $\mathscr{U}_k$ is a linear operator defined from $L_2[0,T]$ to $\mathbb{E}_k[0,T]$ and represented by a $p \times d_k$ matrix U_k, and ε a noise function admitting the basis expansion $\varepsilon(t) = \sum_{j=1}^{p} \beta_j \psi_j(t)$.

We now make some distributional assumptions on the stochastic processes X, Y and ε through their respective basis expansions. Firstly, the basis coefficients $\{\alpha_1,...,\alpha_n\}$ of Y are assumed to be distributed, conditionally to Z, according to a multivariate Gaussian density:

$$\alpha_{|Z_k=1} \sim \mathscr{N}(m_k, S_k),$$

where m_k and $S_k = \mathrm{diag}(a_{k1},...,a_{kd_k})$ are respectively the mean and the covariance matrix of the kth group. Secondly, the basis coefficients $\{\beta_1,...,\beta_n\}$ of the noise function ε are assumed as well to be distributed, conditionally to Z, according to a multivariate Gaussian density:

$$\beta_{|Z_k=1} \sim \mathscr{N}(0, \Gamma_k).$$

With these distributional assumptions, the conditional distribution of the basis coefficients of X is:

$$\gamma_{|\alpha,Z_k=1} \sim \mathcal{N}(U_k\alpha, \Gamma_k),$$

and its marginal distribution is therefore a mixture of Gaussians:

$$p(\gamma) = \sum_{k=1}^{K} \pi_k \phi(\gamma; \mu_k, \Sigma_k),$$

where ϕ is the Gaussian density function, $\mu_k = U_k m_k$, $\Sigma_k = U_k S_k U_k^t + \Gamma_k$ and $\pi_k = P(Z_k = 1)$ is the prior probability of group k. Let us also define $Q_k = [U_k, V_k]$ a $p \times p$ matrix which satisfies $Q_k^t Q_k = Q_k Q_k^t = I_p$ and for which the $p \times (p - d_k)$ matrix V_k is the orthonormal complement of U_k defined above. We finally assume that Γ_k is such that $\Delta_k = Q_k^t \Sigma_k Q_k$ has the following form:

$$\Delta_k = \begin{pmatrix} a_{k1} & & 0 & & \\ & \ddots & & \mathbf{0} & \\ 0 & & a_{kd_k} & & \\ & & & b_k & 0 \\ & \mathbf{0} & & & \ddots \\ & & & 0 & b_k \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} d_k \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} (p - d_k) \end{matrix}$$

with $a_{kj} > b_k$ for $j = 1, ..., d_k$ and $k = 1, ..., K$. This model will be hereafter referred to as the $\text{FLM}_{[a_{kj} b_k Q_k d_k]}$ model or the FLM model for short. With these notations and from a practical point of view, one can say that the variance of the actual data of the kth group is therefore modeled by the $a_{k1}, ..., a_{kd_k}$ whereas the parameter b_k models the variance of the noise. Similarly, the dimension d_k can be considered as the intrinsic dimension of the latent subspace of the kth group.

2.2 The submodels of the FLM model

Following the strategy of [4], it is possible to obtain parsimonious submodels from the $\text{FLM}_{[a_{kj} b_k Q_k d_k]}$ model by constraining model parameters within or between groups. For instance, fixing the first d_k eigenvalues to be common within each class, we obtain the more restricted model $\text{FLM}_{[a_k b_k Q_k d_k]}$. We observed that the model $\text{FLM}_{[a_k b_k Q_k d_k]}$ often gives satisfying results and this suggests that the assumption that each matrix Δ_k contains only two different eigenvalues, a_k and b_k, seems to be an efficient way to regularize the estimation of Δ_k. Another possible type of regularization is to fix the parameters b_k to be common between the classes. This yields the models $\text{FLM}_{[a_{kj} b Q_k d_k]}$ and $\text{FLM}_{[a_k b Q_k d_k]}$ which both assume that the variance outside the class specific subspaces is common. This assumption can be viewed as modeling the noise outside the latent subspace of the group by a single parameter b which is a natural hypothesis when the data are obtained in a common acquisition process. Among the 28 models proposed in the original article [4], 6 models have been selected for their good practical behaviors to be considered in the experiments of Section 3. Table 1 lists those 6 models and their corresponding complexity (*i.e.* the number of parameters to estimate).

2.3 Model inference

In model-based clustering, the estimation of model parameters is traditionally done through the maximum likelihood estimation procedure. Given the basis expansion coefficients $\gamma_1, ..., \gamma_n$ of the

FLM model	Number of parameters	Nb of prms $K=4$, $d=10$, $p=100$
$[a_{kj}b_kQ_kd_k]$	$\rho+\overline{\tau}+2K+D$	4231
$[a_{kj}bQ_kd_k]$	$\rho+\overline{\tau}+K+D+1$	4228
$[a_kb_kQ_kd_k]$	$\rho+\overline{\tau}+3K$	4195
$[ab_kQ_kd_k]$	$\rho+\overline{\tau}+2K+1$	4192
$[a_kbQ_kd_k]$	$\rho+\overline{\tau}+2K+1$	4192
$[abQ_kd_k]$	$\rho+\overline{\tau}+K+2$	4189

Table 1: Properties of the FLM models: $\rho = Kp+K-1$ is the number of parameters required for the estimation of means and proportions, $\overline{\tau} = \sum_{k=1}^{K} d_k[p-(d_k+1)/2]$ and $\tau = d[p-(d+1)/2]$ are the number of parameters required for the estimation of orientation matrices Q_k, and $D = \sum_{k=1}^{K} d_k$. For asymptotic orders, the assumption that $K \ll d \ll p$ is made.

observed curves $x_1,...,x_n$, the complete log-likelihood for the FLM model proposed above has the following form:

$$\begin{aligned}\ell(\theta;\gamma_1,...,\gamma_n) &= -\frac{1}{2}\sum_{k=1}^{K} n_k \left[\sum_{j=1}^{d_k}\left(\log(a_{kj})+\frac{q_{kj}^t W_k q_{kj}}{a_{kj}}\right)\right. \\ &\quad \left.+\sum_{j=d_k+1}^{p}\left(\log(b_k)+\frac{q_{kj}^t W_k q_{kj}}{b_k}\right)-2\log(\pi_k)\right]+\xi,\end{aligned}$$

where q_{kj} is the jth column of Q_k, $W_k = \frac{1}{n_k}\sum_{i=1}^{n} z_{ik}(\gamma_i-\mu_k)^t(\gamma_i-\mu_k)$, $n_k = \sum_{i=1}^{n} z_{ik}$ and ξ is a constant term. However, since the unsupervised classification context is considered here, *i.e.* the group labels of time series are unknown, the direct maximization of the complete log-likelihood is intractable and it is necessary to use an iterative procedure to maximize it. In the model-based clustering context, the EM algorithm is the traditional tool to do so. The EM algorithm alternates between an E step and a M step. For the FLM model introduced above, the EM algorithm takes the following form:

E step This first step aims to compute, at iteration (q), the expectation of the complete log-likelihood conditionally to the current value of the parameter $\theta^{(q-1)}$, which, in practice, reduces to the computation of $t_{ik}^{(q)} = E[z_{ik}|y_i,\theta^{(q-1)}]$. For the $\text{FLM}_{[a_kb_kQ_kd_k]}$ model, the posterior probability t_{ik} can be computed as follows at iteration (q):

$$t_{ik} = 1/\sum_{\ell=1}^{K}\exp\left(H_k(\gamma_i)-H_\ell(\gamma_i)\right),$$

with $H_k(\gamma)$ defined as:

$$H_k(\gamma) = ||\mu_k-P_k(\gamma)||^2_{D_k}+\frac{1}{b_k}||\gamma-P_k(\gamma)||^2+\sum_{j=1}^{d_k}\log(a_{kj})+(p-d_k)\log(b_k)-2\log(\pi_k),$$

where $||.||^2_{\mathscr{D}_k}$ is a norm on the latent space $\mathbb{E}_k$ defined by $||y||^2_{\mathscr{D}_k} = y^t\mathscr{D}_k y$, $\mathscr{D}_k = \tilde{Q}\Delta_k^{-1}\tilde{Q}^t$ and $\tilde{Q}$ is a $p\times p$ matrix containing the d_k vectors of U_k completed by zeros such as $\tilde{Q} = [U_k, 0_{p-d}]$, P_k is the matrix representing the projection operator on the latent space $\mathbb{E}_k$, *i.e.* $P_k(y) = U_kU_k^t y$.

M step This second step estimates the model parameters by maximizing the expectation of the complete likelihood conditionally to the posterior probabilities t_{ik} computed in the previous step. Mixture proportions and means are estimated as usual by:

$$\hat{\pi}_k = \frac{n_k}{n},\ \hat{\mu}_k = \frac{1}{n_k}\sum_{i=1}^{n} t_{ik}\gamma_i.$$

where $n_k = \sum_{i=1}^{n} t_{ik}$. Let us also introduce $C_k = \frac{1}{n_k}\sum_{i=1}^{n} t_{ik}(\gamma_i - \hat{\mu}_k)^t(\gamma_i - \hat{\mu}_k)$, the sample covariance matrix of group k, and W is the inner products between the basis functions, $W = (w_{jk})_{1\leq j,k\leq p} = \int_0^T \phi_j(t)\phi_k(t)dt$. With these notations, the update formula for the other model parameters are in the case of the $\text{FLM}_{[a_k b_k Q_k d_k]}$ model, for $k = 1,...,K$:

- the d_k first columns of Q_k are estimated by the eigenvectors associated with the largest eigenvalues of $C_k W$,
- the variance parameters a_{kj}, $j = 1,...,d_k$, are estimated by the d_k largest eigenvalues of $C_k W$,
- the variance parameters b_k is estimated by $\hat{b}_k = \text{trace}(C_k W) - \sum_{j=1}^{d_k} \hat{a}_{kj}$.

Proof of these results can be deduce from the proof of [4], by substituting the usual metric by the metric induced by the basis functions (W here). The inference algorithm presented here will be referred to as funHDDC in the following.

To summarize and roughly speaking, the funHDDC algorithm models and clusters the time series through their projections in group-specific functional principal subspaces. These functional principal subspaces per group are obtained by doing functional principal component analysis conditionally to the posterior probabilities t_{ik}. However, it is important to notice that even if the modeling and the clustering are done in low-dimensional subspace, no discriminative information is lost thanks to the noise term b_k which models the variance outside the subspaces.

2.4 Estimation of hyper-parameters

The use of the EM algorithm for parameter estimation makes the funHDDC algorithm almost automatic, except for the estimation of the hyper-parameters d_k and K. Indeed, the parameters d_k and K can not be determined by maximizing the likelihood since they both control the model complexity. The estimation of the intrinsic dimensions d_k is a difficult problem with no unique technique to use. In [4], the authors proposed a strategy based on the eigenvalues of the class conditional covariance matrix Σ_k of the kth class. The j^{th} eigenvalue of Σ_k corresponds to the fraction of the full variance carried by the j^{th} eigenvector of Σ_k. The class specific dimension d_k, $k = 1,...,K$ is estimated through the scree-test of Cattell [5] which 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. The threshold can be provided by the user or selected using BIC [17]. The number of clusters K may have to be estimated as well and and can be also selected thanks to the BIC criterion. In the specific case of the models $[a_k b_k Q_k d_k]$ and $[abQ_k d_k]$, it has been recently proved [3] that the maximum likelihood estimate of d_k is asymptotically consistent.

2.5 Links with related models

At this point, it is possible to establish some links with the methods presented in Section 1. The closest strategy is obviously the direct use of HDDC on the basis coefficients. This implies that a "standard" PCA is applied, conditionally to the posterior probabilities, to the data of each group.

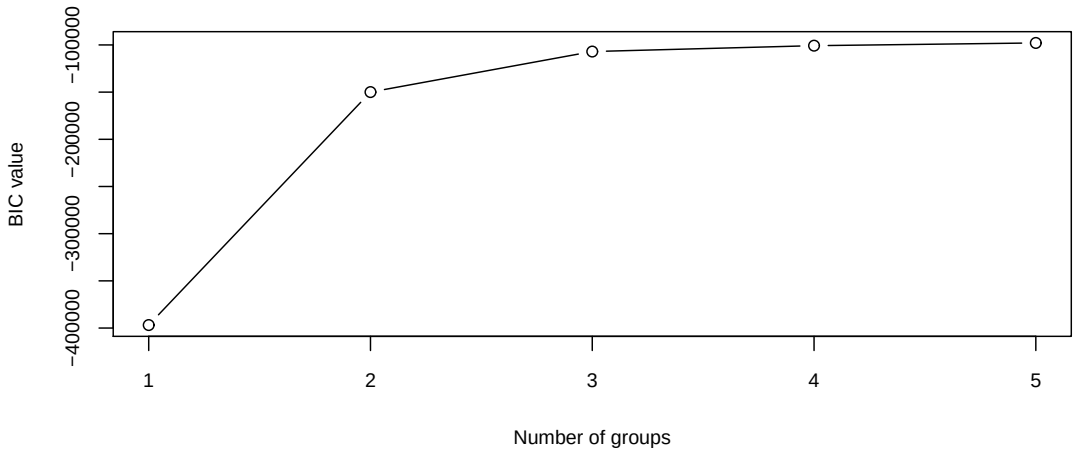


Figure 1: Selection of the number K of groups with the BIC criterion for the Canadian temperature dataset.

The main difference between HDDC and its functional version is the use of a metric specific to the functional data in the eigenspace projection. It is also possible to directly use HDDC on the discretized data. In this case, the functional nature of the data is not considered at all, what could be especially problematic when the curves are observed with noise. We believe that the use of the functional version of HDDC will both improve the clustering results and ease the interpretation of the results by looking at the group-specific functional harmonics.

3 Experimental results

This section presents the results of experiments which aim to both illustrate the funHDDC features and compare the proposed method to existing approaches.

3.1 An introductory example: the Canadian temperature dataset

In this first experiment, the Canadian temperature data (available in the **R** package *fda* and presented in details by [16]) are used to illustrate the main features of the proposed functional clustering method. The dataset consists in the daily measured temperatures at 35 Canadian weather stations across the country. The funHDDC algorithm was applied here with the $[a_{kj}b_kQ_kd_k]$ model, which is the most general FLM model, using a basis of 20 natural cubic splines. Once the funHDDC algorithm has converged, several informations are available and some of them are of particular interest. Group means, intrinsic dimensions of the group-specific subspaces and functional principal components of each group could in particular help the practitioner in understanding the clustering of the dataset at hand.

As discussed before, it is first important to select an appropriate number of components for the dataset to cluster and this can be done using the BIC criterion. Figure 1 shows the BIC values obtained with funHDDC on the Canadian temperature dataset according to the number K of groups. As one can observe, the BIC value increases until $K = 4$ and then stabilizes. This behaviour indicates that 4 groups seem sufficient to model the dataset with funHDDC. Figure 2 presents the

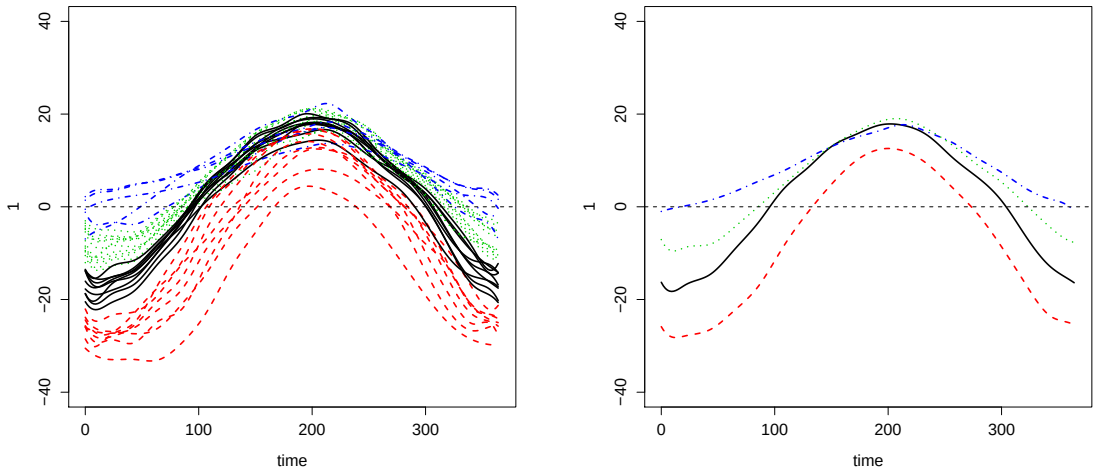
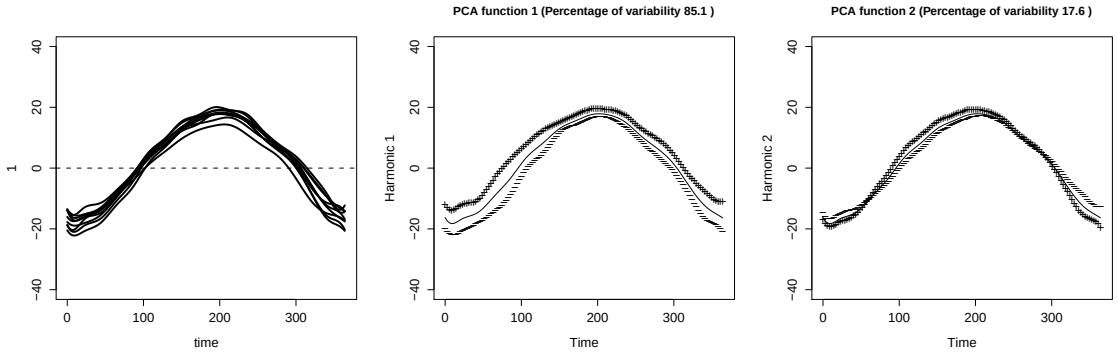
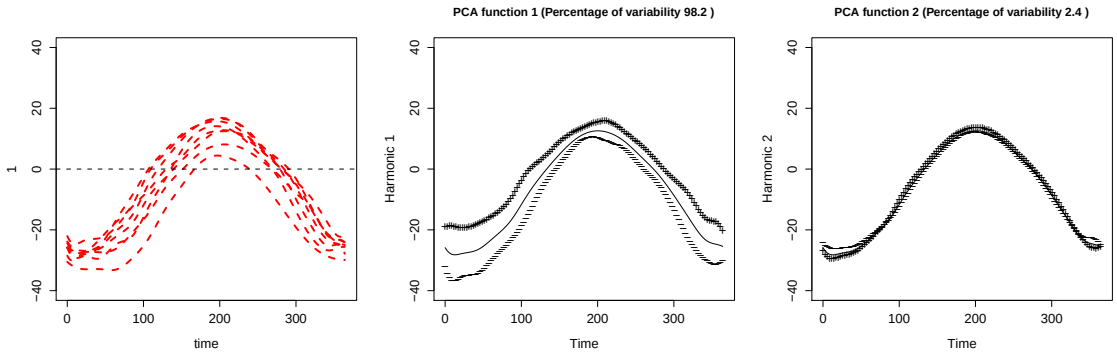


Figure 2: Clustering obtained with funHDDC (model $[a_{kj}b_kQ_kd_k]$) and estimated means of the groups for the Canadian temperature dataset.

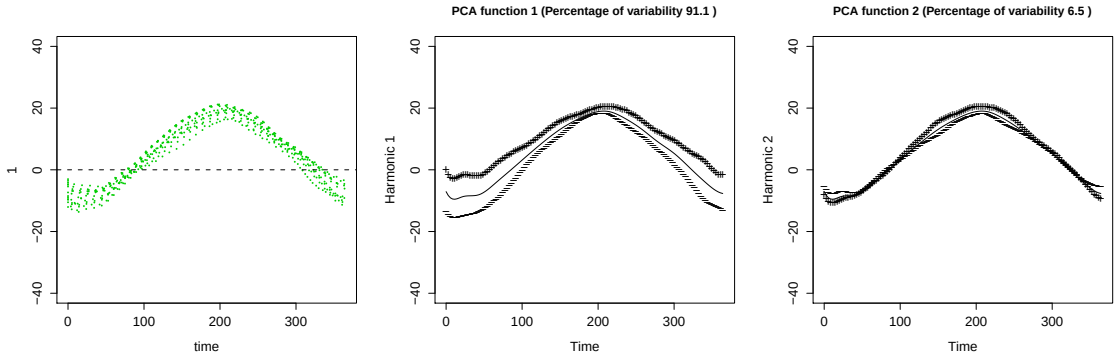
Figure 3: Geographical positions of the Canadian weather stations according to their group belonging provided by funHDDC.



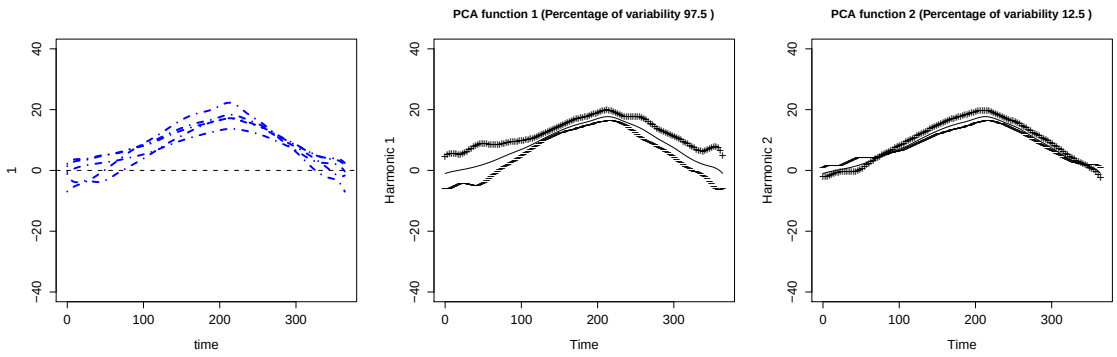
(a) Continental



(b) Arctic



(c) Atlantic



(d) Pacific

Figure 4: The group means of the Canadian temperature data obtained with funHDDC and the effects of adding (+) and subtracting (−) a suitable multiple (± 2 standard deviation) of each functional principal component curve.

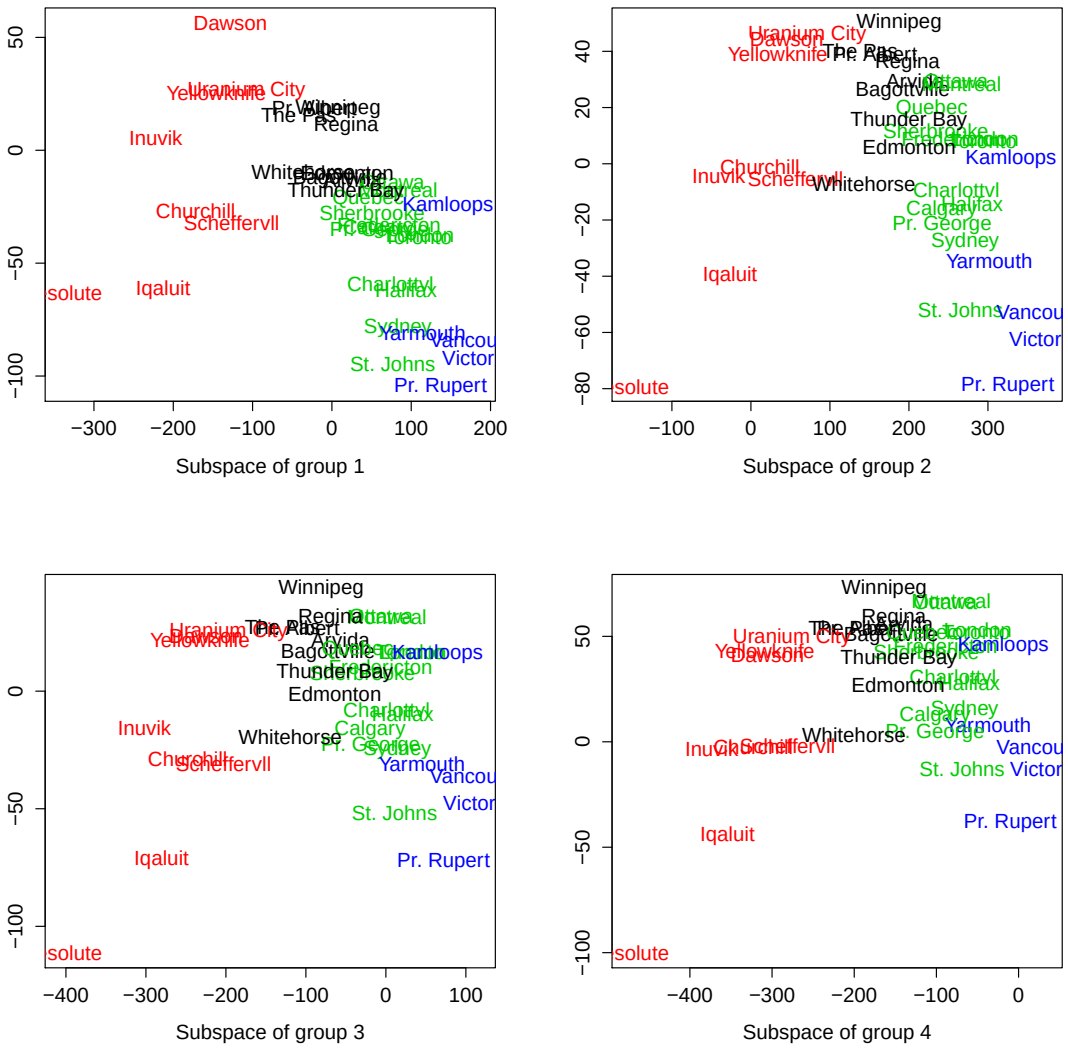


Figure 5: Projection of the observed Canadian temperature curves into the specific functional *subspace* of each group estimated with funHDDC.

clustering into 4 groups obtained with funHDDC and the estimated mean functions of the groups for the temperature dataset. At this point, it is very interesting to have a look at the name of the weather stations gathered in the different groups. Indeed, it appears that the group of red curves gathers the stations of the North of Canada, the "black" group is made of continental stations, the "blue" group contains the stations of the Pacific coast whereas the Atlantic stations are gathered in the "green" group. For instance, the group of the green curves contains stations such as Halifax (Nova Scotia) and St Johns (Newfoundland) whereas the "blue" group has stations such as Vancouver and Victoria (both in British Columbia). Figure 3 provides a map of the weather stations where the colors indicate their group belonging. This figure shows that the obtained clustering with funHDDC is very satisfying and rather coherent with the actual geographical positions of the stations (the clustering accuracy is 71% here). We recall that this partition of the data has been obtained without any other information than the temperature curves. In addition, the observation of the temperature means of the 4 groups confirms the common idea that seasons are more rude in the North of Canada than in the South and that the continental cities have lower temperatures than coast cities during the winter.

Another interesting thing, but not necessary easy to visualize, is the specific functional subspace of each group. A classical way to observe principal component functions is to plot the group mean function as well as the functions obtained by adding and subtracting a suitable multiple of the principal component function in question [16]. Figure 4 shows such a plot for the "continental", "arctic", "Atlantic coast" and "Pacific coast" groups of weather stations. It first appears on the first principal component of each group that there is more variance between the weather stations in winter than in summer. In particular, the first principal component of the "Pacific coast" group (blue curves) reveals a specific phenomenon which occurs at the beginning and the end of the winter. Indeed, we can observe a high variance in the temperatures of the Pacific coast stations at these periods of time which can be explained by the presence of mountain stations in this group. The analysis of the second principal components reveals more fine phenomena. For instance, the second principal component of the "continental" group (black curves) shows a slight shift between the + and − along the year which indicates a time-shift effect. This may mean that some cities of this group have their seasons shifted, *e.g.* late entry and exit in the winter. Similarly, the inversion of the + and − on the second principal component of the Pacific and Atlantic groups (blue and green curves) suggests that, for these groups, the coldest cities in winter are also the warmest cities in summer. On the second principal component of the "arctic" group (red curves), the fact that the + and − curves are almost superimposed shows that the North stations have very similar temperature variations (different temperature means but same amplitude) along the year.

Finally, Figure 5 presents the scores of the curves into the two first functional principal components of each group. These figures provide useful and interpretable maps of the temperature functions. For instance, the first axis of each subspace seems to discriminate the North and South cities. The figures also highlight the similarity between the temperatures of Atlantic and Pacific stations. It also appears that, in this case, the four functional subspaces seem to be parallel (same orientations but different means). To summarize, this first experiment has highlighted that funHDDC, in addition to providing a meaningful partition of the data, allows interpretations which would be certainly helpful in many application fields.

3.2 Benchmark study: data and experimental setup

In the two following benchmark experiments, four real datasets will be under studies: *Kneading*, *CBF*, *Face* and *ECG*. These four datasets are plotted on Figure 6. The first dataset (*Kneading*) comes from a study which consisted in predicting the quality of cookies (good, adjustable or

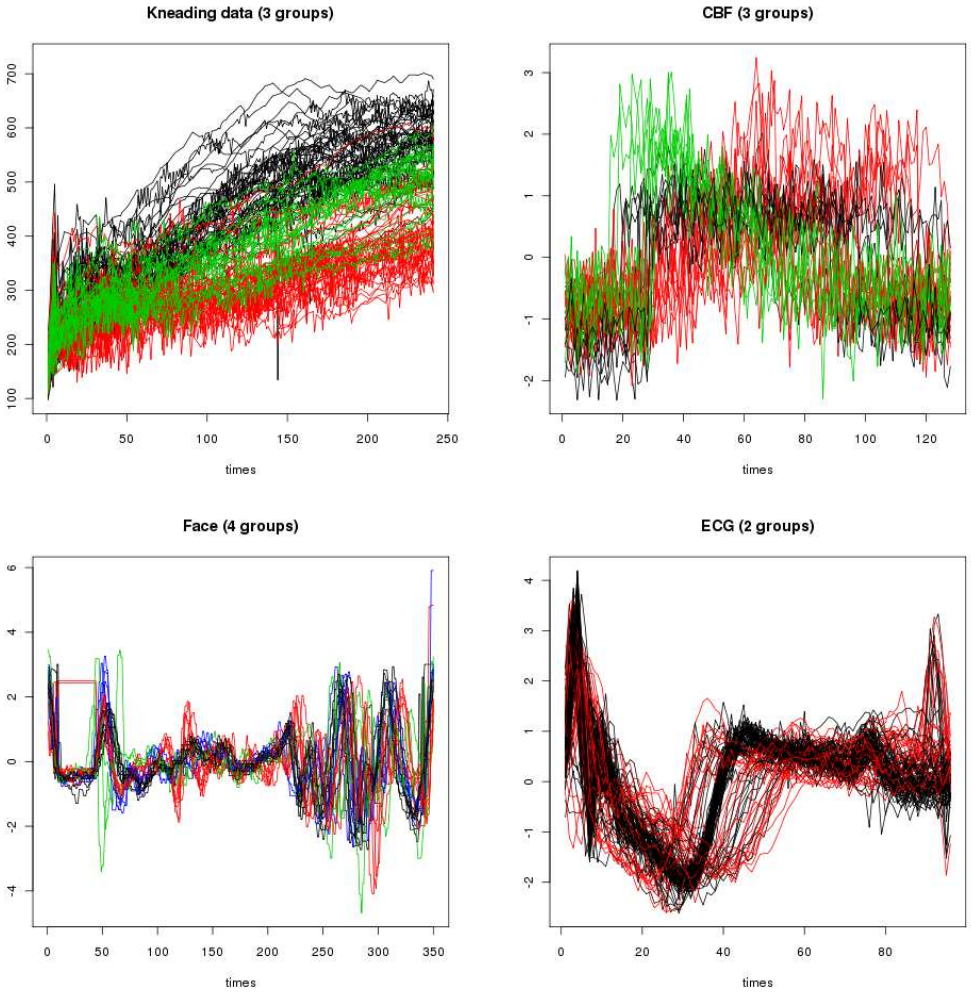


Figure 6: *Kneading*, *CBF*, *Face* and *ECG* datasets.

bad) from the kneading curve representing the resistance (density) of dough observed during the kneading process. The corresponding dataset is made of 115 curves observed at 241 equispaced instants of the time. Among the 115 cookies, 50 have been judged good, 25 adjustable and 40 bad. These data, provided by the Danone company, have been already studied in a supervised context [13, 15]. These data are known to be hard to discriminate, even for supervised classifiers, partly because of the adjustable class. The three other datasets come from the *UCR Time Series Classification and Clustering* website[1]. The *CBF* dataset is made of 930 curves sampled from 3 groups at 128 instants of time. The *Face* dataset [22] consists of 112 curves sampled from 4 groups at 350 instants of time. Finally, the *ECG* dataset [14] consists of 200 curves from 2 groups sampled at 96 time instants.

In the following, two benchmark experiments will allow to compare the clustering ability of the funHDDC method with state-of-the-art methods. First, funHDDC will be compared to the *fclust* method of James and Sugar, described in Section 1, which has also the advantage to take into account the functional nature of the data. Second, funHDDC will be compared to usual two-step methods in which the functional data are first transformed into a finite dimensional vector (simple time discretization, projection into a natural cubic spline basis or onto functional principal components) and then clustered by an usual clustering method (HDDC [4], MixtPPCA [19], kmeans or GMM [2, 6] through the **R** package *mclust*).

3.3 Benchmark study: comparison with *fclust*

A package implementing *fclust* for the **R** software is available on the author's website. However, because of a memory limitation in this package, we had to select a reduced number of curves from the original four datasets. For the Kneading data, 50 curves have been randomly chosen in the 115 original ones, and for the three other datasets, which are separated into a training and a test sample on the UCR website (for supervised classification purpose), only the training part have been kept. For funHDDC, a basis of 20 natural cubic splines has been chosen for each dataset. The clustering results are provided by Table 2 which indicates the correct classification rates for both methods, the BIC values and the intrinsic dimensions for each group-specific functional subspace for funHDDC. These results clearly show that funHDDC outperforms *fclust* on all the datasets. Moreover, it appears that the BIC criterion, used for choosing the number of dimensions (tuned by a common threshold) and the most appropriate submodel, leads to often select the most efficient funHDDC models (for three datasets among four). It should nevertheless be noticed that *fclust* has been developed especially for sparsely sampled functional data, and it would be interesting to compare both methods on such data too.

3.4 Benchmark study: comparison with usual two-step methods

In this section, the clustering performance of funHDDC is compared to the usual two-step methods described in Section 1. The clustering results are summarized in Table 3. For the four datasets, the correct classification rates of each funHDDC submodels is provided, as well as for four classical clustering methods: HDDC, MixtPPCA, *mclust* and k-means. All these two-step methods are successively applied on discretized data, on the coefficients in a natural cubic splines basis expansion (20 splines) and on functional PCA scores. For funHDDC, applied also with a basis of 20 natural cubic splines, the correct classification of the best model according to BIC is underlined.

For the Kneading dataset, HDDC on discretized data appears to be the best method with a correct classification rate of 66.09% whereas the best funHDDC models leads to a rate of 64.35%

[1] *http://www.cs.ucr.edu/∼eamonn/time_series_data/*

dataset	Kneading			CBF		
groups number	3			3		
size	50^2			30		
method	cc	BIC	d	cc	BIC	d
Fun-HDDC $A_{kj}B_kQ_kD_k$	70	-2403	(2,1,1)	63.3	-2430	(1,1,1)
Fun-HDDC $A_{kj}BQ_kD_k$	66.6	-2498	(1,1,1)	63.3	-2498	(1,1,1)
Fun-HDDC $A_kB_kQ_kD_k$	**70**	**-2193**	(1,1,1)	56.6	-2514	(1,1,1)
Fun-HDDC $A_kBQ_kD_k$	66.6	-2402	(1,1,1)	63.3	-2402	(1,1,1)
Fun-HDDC $AB_kQ_kD_k$	66.6	-2195	(1,2,1)	56.6	-2523	(1,1,1)
Fun-HDDC ABQ_kD_k	66.6	-2397	(1,1,1)	**63.3**	**-2397**	(1,1,1)
fclust[3]	60			56.6		

dataset	Face			ECG		
groups number	4			2		
size	24			100		
method	cc	BIC	d	cc	BIC	d
Fun-HDDC $A_{kj}B_kQ_kD_k$	62.5	-2162	(1,1,2,1)	77	-6667	(1,1)
Fun-HDDC $A_{kj}BQ_kD_k$	50	-2286	1,1,1,1)	76	-6428	(1,1)
Fun-HDDC $A_kB_kQ_kD_k$	**62.5**	**-2078**	(2,1,1,1)	77	-6333	(1,1)
Fun-HDDC $A_kBQ_kD_k$	58.3	-2083	(1,2,1,1)	77	-6191	(1,1)
Fun-HDDC $AB_kQ_kD_k$	66.6	-2092	(2,1,2,1)	77	-6317	(1,1)
Fun-HDDC ABQ_kD_k	58.3	-2080	(2,1,1,1)	**77**	**-6167**	(1,1)
fclust[4]	41.6			75		

Table 2: Percentages of correct classification (cc), BIC values (if available), and dimension of each class-specific functional subspace (d) for methods *fclust* and funHDDC on parts of the Kneading, CBF, Face and ECG datasets.

Fun-HDDC	Kneading functional	2-steps methods	Kneading discretized (241 instants)	Kneading spline coeff. (20 splines)	Kneading FPCA scores (4 components)
$A_{kj}B_kQ_kD_k$	64.35	HDDC	**66.09**	53.91	44.35
$A_{kj}BQ_kD_k$	62.61	MixtPPCA	65.22	64.35	62.61
$A_kB_kQ_kD_k$	64.35	mclust	63.48	50.43	60
$A_kBQ_kD_k$	62.61	kmeans	62.61	62.61	62.61
$AB_kQ_kD_k$	64.35				
ABQ_kD_k	62.61				

Fun-HDDC	CBF functional	2-steps methods	CBF discretized (128 instants)	CBF spline coeff. (20 splines)	CBF FPCA scores (17 components)
$A_{kj}B_kQ_kD_k$	64.84	HDDC	68.60	51.18	68.17
$A_{kj}BQ_kD_k$	70.43	MixtPPCA	65.59	51.29	68.27
$A_kB_kQ_kD_k$	64.09	mclust	61.18	62.79	68.06
$A_kBQ_kD_k$	**70.65**	kmeans	64.95	54.09	64.84
$AB_kQ_kD_k$	70.65				
ABQ_kD_k	70.65				

Fun-HDDC	Face functional	2-steps methods	Face discretized (350 instants)	Face spline coeff. (20 splines)	Face FPCA scores (3 components)
$A_{kj}B_kQ_kD_k$	56.25	HDDC	59.82	58.03	63.39
$A_{kj}BQ_kD_k$	54.44	MixtPPCA	54.54	61.36	**64.77**
$A_kB_kQ_kD_k$	51.78	mclust	62.5	57.14	55.36
$A_kBQ_kD_k$	54.44	kmeans	59.09	53.41	59.09
$AB_kQ_kD_k$	60.71				
ABQ_kD_k	57.14				

Fun-HDDC	ECG functional	2-steps methods	ECG discretized (96 instants)	ECG spline coeff. (20 splines)	ECG FPCA scores (19 components)
$A_{kj}B_kQ_kD_k$	75	HDDC	74.5	73.5	74.5
$A_{kj}BQ_kD_k$	-	MixtPPCA	74.5	73.5	74.5
$A_kB_kQ_kD_k$	76.5	mclust	81	80.5	**81.5**
$A_kBQ_kD_k$	74.5	kmeans	74.5	72.5	74.5
$AB_kQ_kD_k$	76.5				
ABQ_kD_k	75				

Table 3: Percentages of correct classification for funHDDC (underlined for the best model according BIC) and usual two-steps methods on the Kneading, CBF, Face and ECG datasets.

and the model selected by BIC obtains 62.61%. For the CBF data, the best method is funHDDC with the model selected by BIC, with a correct classification rate of 70.65%, whereas the best classification rate of the two-step methods (still provided by HDDC on discretized data) is 68.6%. For the Face data, the best approach is MixtPPCA on the functional PCA scores (64.77% versus 60.71% for funHDDC) and *mclust* is the most efficient method on the ECG data also on the FPCA scores.

Each of the studied method, except k-means, turned out to be the best method at least once over the four datasets and this benchmark study is therefore not able to elect a clear winner. The conclusion of these experiments could be that funHDDC is nevertheless a good alternative to two-step clustering methods for the clustering of functional data. Indeed, funHDDC presents the advantage of always providing satisfying results in addition to not requiring to transform the functional data into finite dimensional data. This is an important point since this benchmark study has also highlighted that there are no absolute best way to discretize the functional data. Table 3 in fact shows that each discretization has allowed at least once a two-step method to win. In addition, since the corresponding space in which the functions are represented are not similar, model selection criteria cannot be used to choose between such strategies in an unsupervised classification context. From this point of view, the use of funHDDC appears to be more tenable than two-step methods, since the funHDDC submodel selected by BIC leads to a satisfying classification rate for each dataset.

4 Conclusion

The main objective of the present work was to adapt the HDDC clustering method to functional data. The resulting algorithm, called funHDDC, models and clusters the high-dimensional functional data of each group in a specific functional subspace. The clustering and interpretation abilities of funHDDC have been illustrated on several real-world datasets. In particular, funHDDC has been applied to the well-known Canadian temperature dataset and it provided meaningful and understandable results. The proposed method has also been compared on four benchmark datasets with a recent functional clustering method, *fclust*, and with classical two-step methods. On the one hand, funHDDC turned out to clearly outperforms its functional challenger *fclust*. On the other hand, funHDDC appeared to be always satisfying and more stable than the two-step methods which furthermore suffer from the difficulty to choose the discetization strategy. An extension of this work would be to adapt the funHDDC method to multi-dimensional time series. This would be possible by using a Gaussian model with block-diagonal covariance matrices within the group-specific functional subspaces.

Acknowledgements

The authors would like to thank Professor Cristian Preda (Université Lille 1, France) for his useful comments and the interesting discussions they have had with him.

References

[1] A.M. Aguilera, M. Escabiasa, C. Preda, and G. Saporta. Using basis expansions for estimating functional pls regression. applications with chemometric data. *Chemometrics and Intelligent Laboratory Systems*, 104(2):289–305, 2011.

[2] J.D. Banfield and A.E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49:803–821, 1993.

[3] C. Bouveyron, G. Celeux, and S. Girard. Intrinsic dimension estimation by maximum likelihood in probabilistic PCA. Technical Report 440372, Université Paris 1, 2010.

[4] C. Bouveyron, S. Girard, and C. Schmid. High Dimensional Data Clustering. *Computational Statistics and Data Analysis*, 52:502–519, 2007.

[5] R. Cattell. The scree test for the number of factors. *Multivariate Behav. Res.*, 1(2):245–276, 1966.

[6] G. Celeux and G. Govaert. Gaussian parsimonious clustering models. *The Journal of the Pattern Recognition Society*, 28:781–793, 1995.

[7] A. Delaigle and P. Hall. Defining pobability density for a distribution of random functions. *The Annals of Statistics*, 38:1171–1193, 2010.

[8] F. Ferraty and P. Vieu. *Nonparametric functional data analysis*. Springer Series in Statistics. Springer, New York, 2006.

[9] S. Frühwirth-Schnatter and S. Kaufmann. Model-based clustering of multiple time series. *Journal of Business and Economic Statistics*, 26:78–89, 2008.

[10] J.A. Hartigan and M.A. Wong. Algorithm as 1326 : A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1978.

[11] J. Jacques, C. Bouveyron, S. Girard, O. Devos, L Duponchel, , and C. Ruckebusch. Gaussian mixture models for the classification of high-dimensional vibrational spectroscopy data. *Journal of Chemometrics*, 24:719–727, 2010.

[12] G.M. James and C.A. Sugar. Clustering for sparsely sampled functional data. *J. Amer. Statist. Assoc.*, 98(462):397–408, 2003.

[13] C. Lévéder, P.A. Abraham, E. Cornillon, E. Matzner-Lober, and N. Molinari. Discrimination de courbes de prétrissage. In *Chimiométrie 2004*, pages 37–43, Paris, 2004.

[14] R.T. Olszewski. *Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2001.

[15] C. Preda, G. Saporta, and C. Lévéder. PLS classification of functional data. *Comput. Statist.*, 22(2):223–235, 2007.

[16] J. O. Ramsay and B. W. Silverman. *Functional data analysis*. Springer Series in Statistics. Springer, New York, second edition, 2005.

[17] G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6:461–464, 1978.

[18] T. Tarpey and K.J. Kinateder. Clustering functional data. *J. Classification*, 20(1):93–114, 2003.

[19] M. E. Tipping and C. Bishop. Mixtures of principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[20] G. Wahba. *Spline models for observational data*. SIAM, Philadelphia, 1990.

[21] T. Warren Liao. Clustering of time series data – a survey. *Pattern Recognition*, 38:1857–1874, 2005.

[22] X. Xi, E. Keogh, C. Shelton, L. Wei, and C.A. Ratanamahatana. Fast time series classification using numerosity reduction. In *23rd International Conference on Machine Learning (ICML 2006)*, Pittsburgh, PA, 2006.