Model Based Clustering of Functional Data via Mixtures of t Distributions

Cristina Anton, Iain Smith

Model based clustering of functional data via mixtures of t distributions

Abstract

We propose a procedure, called T-funHDDC, for clustering multivariate functional data with outliers which extends the functional high dimensional data clustering (funHDDC) method (Schmutz et al, 2020) by considering a mixture of multivariate t distributions. We define a family of latent mixture models following the approach used for the parsimonious models considered in funHDDC and also constraining or not the degrees of freedom of the multivariate t distributions to be equal across the mixture components. The parameters of these models are estimated using an expectation maximization (EM) algorithm. In addition to proposing the T-funHDDC method, we add a family of parsimonious models to C-funHDDC, which is an alternative method for clustering multivariate functional data with outliers based on a mixture of contaminated normal distributions (Amovin-Assagba et al, 2022). We compare T-funHDDC, C-funHDDC, and other existing methods on simulated functional data with outliers and for real-world data. T-funHDDC outperforms funHDDC when applied to functional data with outliers, and its good performance makes it an alternative to C-funHDDC. We also apply the T-funHDDC method to the analysis of traffic flow in Edmonton, Canada.

MSC Classification: 62H30 , 68T10 , 62F35

1 Introduction

Recently, with the rise of the internet of things (IoT), we use significantly more sensors and have access to frequently recorded functional data (Ramsay and Silverman, 2006). For example, in Section 4.4 we present an application regarding analyzing traffic data and finding trends in traffic flow related to speed differences and weather conditions (see Fig. 1).

In Fig. 1 we plot three observations from this four-dimensional functional data set. The data in the first two dimensions represent the number of cars having a speed 5-10 km/h under the speed limit (see Fig. 1 a), and the number of cars having a speed 0-5 km/h over the speed limit (see Fig. 1 b). They were recorded when the cars passed through a road section in Edmonton, Canada on

August 14, 15, and 20, 2018 for the green, red, and black curves respectively. Weather conditions for these different dates are illustrated by temperature (see Fig. 1 c) and visibility (see Fig. 1 d), and they are included in the third and the fourth dimensions.

The city of Edmonton collects a huge amount of traffic data that inevitably includes atypical observations (outliers) due to holidays, various events, and extreme weather conditions. To analyze these data, it is useful to separate the observations into groups, but to account for contamination with outliers we need to apply robust methods. Here we propose a model-based method for clustering functional data with outliers. We relax the normality assumption and consider mixtures of t distributions.

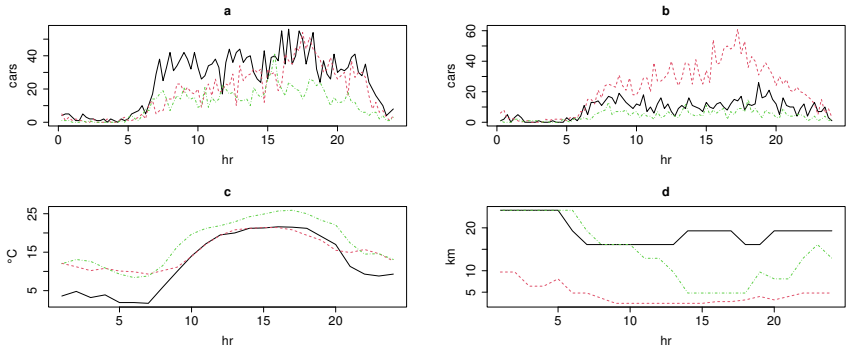

Fig. 1 Traffic data from Edmonton recorded on August 14 (green -.- lines), 15 (red - - lines) and 20 (black - lines), 2018. a. Number of cars having a speed 5-10 km/h under the speed limit b. Number of cars having a speed 0-5 km/h over the speed limit c. Temperature (Celsius) by hour. d. Visibility (km) by hour.

Several methods for clustering functional data have been proposed (see Jacques and Preda (2014a) for a survey). Model-based methods are not directly available for functional data because functional data live in an infinite dimensional space and the notion of probability density function generally does not exist for these data (Delaigle and Hall, 2010). There are many model-based methods for clustering multivariate data (see, for example, McLachlan and Peel, 2004, Celeux and Govaert, 1995, Bouveyron et al, 2007, Punzo and McNicholas, 2016, Punzo et al, 2020, Farcomeni and Punzo, 2020, Andrews and McNicholas, 2011, Andrews and McNicholas, 2012, Dang et al, 2015, Peel and McLachlan, 2000, Tomarchio et al, 2022, Bagnato et al, 2017), and a first approach, called the raw-data clustering (Jacques and Preda, 2014a), consists of directly applying multivariate clustering techniques to the finite discretizations of the functions. A second approach is to use a two-step method and first do a decomposition of the functional data in a basis of functions (such as Fourier basis, B-splines, etc.), and then directly apply multivariate clustering methods to the basis coefficients. A third approach, which allows the

interaction between the discretization and the clustering steps, is based on a probabilistic model for the basis coefficients (Bouveyron and Jacques, 2011, Jacques and Preda, 2013, Schmutz et al, 2020,Amovin-Assagba et al, 2022). We use this approach, and we propose a robust model-based method, called T-funHDDC, which extends the functional high dimensional data clustering (funHDDC) (Bouveyron and Jacques, 2011, Schmutz et al, 2020) to clustering functional data with outliers.

In Chapter 2 in Ritter (2014) outliers are separated into two types. The first type of outliers, called mild outliers, are usually sampled from a population different from the assumed model, so we need to choose a model flexible enough to accommodate them. The second type of outliers, called gross outliers, are unpredictable and incalculable observations that cannot be modeled by a distribution, so it is recommended to suppress gross outliers by trimming them. The methods presented in this paper are designed for dealing with mild outliers.

Outlier detection for functional data is studied in several papers based on functional depths, which measure the centrality of a given curve within a group of trajectories. Depth measures were originally introduced in multivariate data analysis to provide a way to order points in the Euclidean space from center to outward, such that points near the center should have higher depth and points far from the center should have lower depth. Functional outliers are curves that are expected to be far away from the center of the data, so they will correspond to curves of significantly low depth. The first functional data depth was introduced in Fraiman and Muniz (2001). The performance of five notions of data depth is analyzed in Cuevas et al (2007). In Febrero-Bande et al (2008) a procedure is proposed to detect functional outliers that avoids masking. Masking appears when true outliers mask the presence of others, so if a set of outliers is masked in one iteration, they may be found in a later iteration after removing detected outliers.

In addition to the methods based on functional depths, several robust methods based on trimming an *a priori* known proportion of outliers were also proposed. The trimmed k-means method (Cuesta-Albertos et al, 1997) is extended to functional data in García-Escudero and Gordaliza (2005). In Rivera-García et al (2019) a robust model-based functional clustering method is proposed. This approach is based on the ideas in Delaigle and Hall (2010) and Jacques and Preda (2013) for an approximation of the "density" for functional data, together with the simultaneous use of trimming and constraints. Our approach does not involve trimming the outliers and it is inspired by the method teigen (Andrews and McNicholas, 2012, Andrews et al, 2018) for clustering multivariate data with outliers.

The method C-funHDDC was proposed in Amovin-Assagba et al (2022) (see also Anton and Smith (2023) for the univariate case) and it is an extension of the method CNmixt (Punzo and McNicholas, 2016, Punzo et al, 2018) to the functional setting. Multivariate functional data are modeled into a functional subspace using multivariate functional principal component analysis (Jacques

and Preda, 2013) and a model for the basis coefficients based on a mixture of contaminated multivariate normal distributions. A multivariate contaminated normal distribution (Punzo and McNicholas, 2016) is a two-component normal mixture in which the abnormal observations (outliers) are represented by a component with a small prior probability and an inflated covariance matrix. Here we extend the method C-funHDDC to include five more parsimonious models.

We also propose the new robust model-based method T-funHDDC. We follow the approach used by the model-based clustering procedure funHDDC proposed in Bouveyron and Jacques (2011), and extended to multivariate functional data in Schmutz et al (2020). To fit the outliers well, instead of multivariate normal distributions we consider within-cluster multivariate t distributions that have heavier tails. This method allows us to consider model selection criteria for finding the number of clusters, and it does not include trimming, so we do not need to know *a priori* the proportion of outliers.

The paper is organized as follows. In the next section we introduce notation and mixture models for the T-funHDDC method. In Section 3 we present parameters estimation, proposed criteria for selecting the number of clusters, and computational details. Comparisons between C-funHDDC, T-funHDDC, and other existing methods on simulated and real data sets are presented in Section 4. In Section 5 we include a discussion and final conclusions.

2 Multivariate functional data

We assume that the n p-variate curves $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$ are independent realizations of a L^2- continuous stochastic process $\boldsymbol{Y} = \{\boldsymbol{Y}(t)\}_{t\in[0,T]} = \{(Y^1(t), \ldots, Y^p(t))\}_{t\in[0,T]}$ for which the sample paths (i.e. the curves $\boldsymbol{X}_i = (X_i^1, \ldots, X_i^p)$) are such that $X_i^s \in L^2[0,T] = \{f : [0,T] \to \mathbb{R}, \int_{[0,T]} f^2(t)dt < \infty\}$, $i = 1, \ldots, n$, $s = 1, \ldots, p$ (Jacques and Preda, 2014b). For each curve $\boldsymbol{X}_i$ we have access to a finite set of values $x_i^s(t_{i1}) \ldots, x_i^s(t_{im_i})$, where $0 \leq t_{i1} < t_{i2} < \cdots < t_{im_i} \leq T$, $s = 1, \ldots, p$, $i = 1, \ldots, n$. To reconstruct the functional form of the data we assume that the curves belong to a finite dimensional space, and we have:

$$X_i^s(t) = \sum_{r=1}^{R_s} c_{ir}^s \xi_r^s(t) \tag{1}$$

where $\{\xi_r^s\}_{1\leq r\leq R_s}$ is the basis for the s^{th} component of the multivariate curves, c_{ir}^s are the coefficients, and R_s is the number of basis functions. Gathering the coefficients and the basis functions we rewrite (1) as

$$\boldsymbol{X}(t) = \boldsymbol{C}\boldsymbol{\xi}^\top(t), \quad \boldsymbol{X}(t) = (\boldsymbol{X}_1(t), \ldots, \boldsymbol{X}_n(t))^\top, \tag{2}$$

with

$$\boldsymbol{C} = \begin{pmatrix} c_{11}^1 & \dots & c_{1R_1}^1 & c_{11}^2 & \dots & c_{1R_2}^2 & \dots & c_{11}^p & \dots & c_{1R_p}^p \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\ c_{n1}^1 & \dots & c_{nR_1}^1 & c_{n1}^2 & \dots & c_{nR_2}^2 & \dots & c_{n1}^p & \dots & c_{nR_p}^p \end{pmatrix},$$

$$\boldsymbol{\xi}(t) = \begin{pmatrix} \xi_1^1(t) & \dots & \xi_{R_1}^1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & \xi_1^2(t) & \dots & \xi_{R_2}^2(t) & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & \dots & \xi_1^p(t) & \dots & \xi_{R_p}^p(t) \end{pmatrix}.$$

Supposing that the curves are observed with noise, we use least square smoothing to get the expansion for each curve (Ramsay and Silverman, 2006). The choice and the number of the basis functions depends on the data. Fourier bases are usually used for data with a repetitive pattern and B-splines functions for smooth curves (Schmutz et al, 2020).

The notion of probability density function does not generally exist for functional data (Delaigle and Hall, 2010), but it can be approximated with the probability density of the functional principal components (FPCA) scores (Ramsay and Silverman, 2006). Using (2) the FPCA scores can be obtained directly from a PCA of the coefficients $\boldsymbol{C}$ with a metric based on the inner products between the basis functions.

2.1 The functional latent mixture models

We want to cluster the n observed curves $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ in K homogeneous groups. We suppose that there exists a latent variable $\boldsymbol{Z}_i = (Z_{i1}, \dots, Z_{iK})^\top$, associated to each observation $\boldsymbol{x}_i$, where $Z_{ik} = 1$ if the observation $\boldsymbol{x}_i$ belongs to the cluster k and $Z_{ik} = 0$ otherwise.

We assume that for every $k \in \{1, \dots, K\}$ the stochastic process associated with the kth cluster can be described in a lower dimensional subspace $\mathbb{E}^k[0,T] \subset L^2[0,T]$ with dimension $d_k \leq R = \sum_{s=1}^p R_s$ and spanned by the first d_k elements of a group specific basis of functions $\{\boldsymbol{\zeta}_{kr}, r = 1, \dots, R\}$ that can be obtained from $\{\xi_r^s, s = 1, \dots, p, r = 1, \dots, R\}$ by a linear transformation using a multivariate functional principal component analysis (MFPCA) such that we have

$$\boldsymbol{\zeta}_{kr}(t) = \sum_{l=1}^{R} q_{krl} \boldsymbol{\xi}_l(t), \quad r = 1, \dots, R,$$

where $\boldsymbol{Q}_k = (q_{krl})_{r,l=1,\dots,R}$ is the orthogonal $R \times R$ matrix containing the coefficients of the eigenfunctions expressed in the initial basis $\boldsymbol{\xi}$. We suppose that the first d_k eigenfunctions contain the main information of the MFPCA of cluster k and we split $\boldsymbol{Q}_k = [\boldsymbol{U}_k, \boldsymbol{V}_k]$ such that $\boldsymbol{U}_k$ is of size $R \times d_k$, $\boldsymbol{V}_k$ is of

size $R \times (R - d_k)$ and we have

$$\boldsymbol{Q}_k^\top \boldsymbol{Q}_k = \boldsymbol{I}_R, \quad \boldsymbol{U}_k^\top \boldsymbol{U}_k = \boldsymbol{I}_{d_k}, \quad \boldsymbol{V}_k^\top \boldsymbol{V}_k = \boldsymbol{I}_{R-d_k}, \quad \boldsymbol{U}_k^\top \boldsymbol{V}_k = \boldsymbol{0}.$$

The relationship between the coefficients $\boldsymbol{c}_i$ in the ith row of the matrix $\boldsymbol{C}$ and the score $\boldsymbol{\delta}_i$ is

$$\boldsymbol{c}_i = \mathbf{W}^{-1/2} \boldsymbol{U}_k \boldsymbol{\delta}_i + \boldsymbol{\epsilon}_i, \quad \boldsymbol{W} = \int_0^T \boldsymbol{\xi}(t)^\top \boldsymbol{\xi}(t) dt,$$

where $\boldsymbol{W}$ is the symmetric block-diagonal $R \times R$ matrix of inner products between the basis functions and $\boldsymbol{\epsilon}_i$ is the noise. We can make distribution assumptions on the scores $\boldsymbol{\delta}_i$ (Delaigle and Hall, 2010), such that the coefficients $\boldsymbol{c}_i$, $i = 1, \ldots, n$ arise from a parametric mixture distribution

$$p(\boldsymbol{c}_i;\ \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k), \quad \sum_{k=1}^{K} \pi_k = 1, \tag{3}$$

where $\pi_k \in [0, 1]$ are the mixing proportions, $\boldsymbol{\theta} = \bigcup_{k=1}^{k} (\boldsymbol{\theta}_k \cup \{\pi_k\})$ is the set formed with the parameters, and $f_k(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k)$ are the component densities.

For the model associated with the funHDDC method (Schmutz et al, 2020), we assume that independently for $i = 1, \ldots, n$

$$\boldsymbol{\epsilon}_i \mid Z_{ik} = 1 \sim N(\boldsymbol{0}, \boldsymbol{\Lambda}_k), \text{ and } \boldsymbol{\delta}_i \mid Z_{ik} = 1 \sim N(\boldsymbol{m}_k, \boldsymbol{\Delta}_k).$$

Thus $p(\boldsymbol{c}_i;\ \boldsymbol{\theta})$ is the density of a mixture of Gaussian distributions with $f_k(\mathbf{c}_i;\ \boldsymbol{\theta}_k) = \phi(\boldsymbol{c}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, the density for the $R-$variate normal distribution $N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

$$\phi(\boldsymbol{c}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-R/2} \mid \boldsymbol{\Sigma}_k \mid^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{c}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{c}_i - \boldsymbol{\mu}_k)\right). \tag{4}$$

Here $\mid \boldsymbol{\Sigma}_k \mid$ denotes the determinant of $\boldsymbol{\Sigma}_k$, and

$$\boldsymbol{\mu}_k = \mathbf{W}^{-1/2} \boldsymbol{U}_k \boldsymbol{m}_k, \quad \boldsymbol{\Sigma}_k = \boldsymbol{W}^{-1/2} \boldsymbol{U}_k \boldsymbol{\Delta}_k \boldsymbol{U}_k^\top \boldsymbol{W}^{-1/2} + \boldsymbol{\Lambda}_k, \tag{5}$$

where the noise covariance $\boldsymbol{\Lambda}_k$ is such that the covariance $\boldsymbol{D}_k$ of the data in the space generated by the eigenfunctions $\boldsymbol{\zeta}_{kr}$ is a diagonal matrix given by

$$\boldsymbol{D}_k = \boldsymbol{Q}_k^\top \boldsymbol{W}^{1/2} \boldsymbol{\Sigma}_k \boldsymbol{W}^{1/2} \boldsymbol{Q}_k = \text{diag}(a_{k1}, \ldots, a_{kd_k}, b_k, \ldots, b_k), \tag{6}$$

with $a_{k1} > a_{k2} > \cdots > a_{kd_k} > b_k$.

For C-funHDDC (Amovin-Assagba et al, 2022) we consider also a latent variable $\boldsymbol{\Upsilon}_i = (\Upsilon_{i1}, \ldots, \Upsilon_{iK}) \in \{0, 1\}^K$ where $\Upsilon_{ik} = 0$ if the observation $\boldsymbol{x}_i$ in

cluster k is an outlier and $\Upsilon_{ik} = 1$ if the observation $\boldsymbol{x}_i$ in cluster k is not an outlier. We assume that (Amovin-Assagba et al, 2022)

$$\begin{aligned}
\boldsymbol{\epsilon}_i \mid Z_{ik} = 1 &\sim N(\mathbf{0}, \boldsymbol{\Lambda}_k),\\
\boldsymbol{\delta}_i \mid Z_{ik} = 1, v_{ik} = 1 &\sim N(\boldsymbol{m}_k, \boldsymbol{\Delta}_k),\\
\boldsymbol{\delta}_i \mid Z_{ik} = 1, v_{ik} = 0 &\sim N(\boldsymbol{m}_k, \eta_k\boldsymbol{\Delta}_k),
\end{aligned}$$

with $\eta_k > 1$ an inflation parameter measuring the increase in variability due to outliers. Consequently

$$\boldsymbol{c}_i \mid z_{ik} = 1, v_{ik} = 1 \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad \boldsymbol{c}_i \mid z_{ik} = 1, v_{ik} = 0 \sim N(\boldsymbol{\mu}_k, \eta_k\boldsymbol{\Sigma}_k)$$

with $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ as in (5) and $\boldsymbol{\Lambda}_k$ such that we have (6).

Thus $\boldsymbol{c}_i$ arise from a mixture of contaminated Gaussians with $f_k(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k) = f_c(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k)$, the density of a multivariate contaminated normal distribution

$$f_c(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k) = \alpha_k\phi(\boldsymbol{c}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + (1-\alpha_k)\phi(\boldsymbol{c}_i;\ \boldsymbol{\mu}_k, \eta_k\boldsymbol{\Sigma}_k),$$

where $\alpha_k \in (0.5, 1)$, $\eta_k > 1$, $\boldsymbol{\theta}_k = \{\alpha_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \eta_k\}$, and $\phi(\boldsymbol{c}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is given in (4). Here α_k defines the proportion of uncontaminated data in the kth cluster. Conditions for the identifiability of this model are given in Amovin-Assagba et al (2022) (see also Punzo and McNicholas, 2016, Propositions 1 and 2).

We refer to this model as FCLM$[a_{kj}, b_k, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$ (functional contaminated latent mixture) and we consider the parsimonious sub-models:

- FCLM$[a_{kj}, b, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$: the parameters b_k are common between the clusters
- FCLM$[a_k, b_k, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$: the first d_k diagonal elements of $\mathbf{D}_k$ are common within each class
- FCLM$[a, b_k, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$: the first d_k diagonal elements of $\boldsymbol{D}_k$ are common within each class and between the clusters
- FCLM$[a_k, b, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$: the parameters b_k are common between the clusters and the first d_k diagonal elements of $\boldsymbol{D}_k$ are common within each class
- FCLM$[a, b, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$: the parameters b_k are common between the clusters and the first d_k diagonal elements of $\boldsymbol{D}_k$ are common within each class and between the clusters

A different way to account for the presence of outliers is to assume that for the kth cluster, $\boldsymbol{c}_i$ arises from the multivariate t distribution with ν_k degrees of freedom and density (McLachlan and Peel, 2004, Section 7.5)

$$f_t(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k) = \frac{\Gamma(\frac{\nu_k+R}{2}) \mid \boldsymbol{\Sigma}_k \mid^{-1/2}}{(\pi\nu_k)^{R/2}\Gamma(\frac{\nu_k}{2})\left(1 + \frac{(\boldsymbol{c}_i-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{c}_i-\boldsymbol{\mu}_k)}{\nu_k}\right)^{\frac{\nu_k+R}{2}}}, \tag{7}$$

where $\Gamma(\cdot)$ is the gamma function and $\boldsymbol{\theta}_k = \{\nu_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, k = 1, \ldots, K\}$. We proceed as in Section 7.5 in McLachlan and Peel (2004) and we characterize the multivariate t distributions by introducing latent variable H_i, $i = 1, \ldots, n$ such that we have

$$\begin{aligned}\boldsymbol{\delta}_i \mid H_i = h_i, Z_{ik} = 1 &\sim N(\boldsymbol{m}_k, \boldsymbol{\Delta}_k/h_i),\\ \boldsymbol{\epsilon}_i \mid H_i = h_i, Z_{ik} = 1 &\sim N(\mathbf{0}, \boldsymbol{\Lambda}_k/h_i),\end{aligned}$$

independently for $i = 1, \ldots, n$, and $H_i \mid Z_{ik} = 1 \sim \text{Gamma}(\nu_k/2, \nu_k/2)$, independently for $i = 1, \ldots, n$. Here the density of a gamma distribution $\text{Gamma}(\alpha_1, \alpha_2)$ with parameters $\alpha_1 > 0$, $\alpha_2 > 0$ is

$$g(y;\ \alpha_1, \alpha_2) = \frac{\alpha_2^{\alpha_1} y^{\alpha_1 - 1} e^{-\alpha_2 y}}{\Gamma(\alpha_1)} I_{y>0}.$$

Hence

$$\boldsymbol{c}_i \mid H_i = h_i, Z_{ik} = 1 \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k/h_i),$$

with $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ as in (5) and $\boldsymbol{\Lambda}_k$ such that we have (6).

Each curve $\boldsymbol{X}_i$ has a basis expansion with coefficient $\boldsymbol{c}_i$ whose distribution (3) is a mixture of multivariate t distributions with density $f_k(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k) = f_t(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k)$ given in (7).

We refer to this model as FTLM$[a_{kj}, b_k, \boldsymbol{Q}_k, d_k, \nu_k]$(functional t latent mixture), and we also consider the parsimonious sub-models:

- FTLM$[a_{kj}, b, \boldsymbol{Q}_k, d_k, \nu_k]$: the parameters b_k are common between the clusters
- FTLM$[a_k, b_k, \boldsymbol{Q}_k, d_k, \nu_k]$: the first d_k diagonal elements of $\boldsymbol{D}_k$ are common within each class
- FTLM$[a, b_k, \boldsymbol{Q}_k, d_k, \nu_k]$: the first d_k diagonal elements of $\boldsymbol{D}_k$ are common within each class and between the clusters
- FTLM$[a_k, b, \boldsymbol{Q}_k, d_k, \nu_k]$: the parameters b_k are common between the clusters and the first d_k diagonal elements of $\boldsymbol{D}_k$ are common within each class
- FTLM$[a, b, \boldsymbol{Q}_k, d_k, \nu_k]$: the parameters b_k are common between the clusters and the first d_k diagonal elements of $\boldsymbol{D}_k$ are common within each class and between the clusters

If we constrain the degrees of freedom $\nu_k = \nu$, $k = 1, \ldots, K$ to be the same for all the K groups we obtain six more parsimonious sub-models.

In Holzmann et al (2006) it is shown that finite location-scatter mixtures from the multivariate t distribution, even with variable degree of freedom, are identifiable. Consequently, in the space of coefficients $\boldsymbol{c}_i$ we can deduce the identifiability of the the model (3) with density $f_k(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k)$ given by the density of the multivariate t distribution $f_t(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k)$.

Next we analyze the complexity (i.e. the number of free parameters to be estimated) of the FTLM$[a_{kj}, b_k, \boldsymbol{Q}_k, d_k, \nu_k]$ model. Let $\tau_1 = KR + K - 1$ be

the number of parameters required for the estimation of the means $\boldsymbol{\mu}_k$ and the proportions π_k; $\tau_2 = \sum_{k=1}^{K} d_k[R - (d_k + 1)/2]$ be the number of parameters required for the estimation of the matrices $\boldsymbol{Q}_k$; $\tau_3 = K + \sum_{k=1}^{K} d_k$ be the number of parameters required for the estimation of b_k, a_{kj}. Then, after we add K more parameters for estimating the degrees of freedom ν_k, the total number of parameters to be estimated for the model FTLM$[a_{kj}, b_k, \boldsymbol{Q}_k, d_k, \nu_k]$ is $\tau = \tau_1 + \tau_2 + \tau_3 + K$. Thus, this model has K more parameters than the funHDDC model FLM$[a_{kj}, b_k, \boldsymbol{Q}_k, d_k]$ (Bouveyron and Jacques, 2011), and K less parameters than C-funHDDC model FCLM$[a_{kj}, b_k, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$ (Amovin-Assagba et al, 2022). If instead of a functional approach we follow a raw-data clustering approach and we apply the teigen (Andrews and McNicholas, 2012) or the CNmixt (Punzo and McNicholas, 2016) methods directly to the discretized data we obtain a larger complexity because we have to work with very high dimensional vectors (Amovin-Assagba et al, 2022).

3 Model inference

In model-based clustering the parameters estimation is usually done using the expectation-maximization (EM) algorithm (Dempster et al, 1977). In the expectation (E) step the conditional expectation of the complete log-likelihood is computed using the current estimates of the parameters. Then in the maximization (M) step the estimates of the parameters are updated with the values that maximize the expected complete log-likelihood. The algorithm consists of successive iterations of the E and the M steps until convergence is achieved. For the models associated with the funHDDC method an EM algorithm is constructed in Schmutz et al (2020).

3.1 Inference for the C-funHDDC models

The parameters of the C-funHDDC models are estimated using an expectation-conditional maximization (ECM) algorithm (Amovin-Assagba et al, 2022, Anton and Smith, 2023). The ECM algorithm (Meng and Rubin, 1993) is a variant of the EM algorithm in which we replace the M-step by two simpler CM-steps given by a partition of the set of parameters. For C-funHDDC the partition is $\boldsymbol{\theta} = \{\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2\}$, where $\boldsymbol{\Psi}_1 = \{\pi_k, \alpha_k, \boldsymbol{\mu}_k, a_{kj}, b_k, q_{kj}, k = 1, \ldots, K, j = 1, \ldots, d_k\}$ and $\boldsymbol{\Psi}_2 = \{\eta_k, k = 1, \ldots, K\}$.

The detailed presentation of the ECM algorithm for the FCLM$[a_{kj}, b_k, \mathbf{Q}_k, d_k, \alpha_k, \eta_k]$ model is given in Amovin-Assagba et al (2022). Here we consider a slight modification for the estimation of α_k and we extend the method to the other five parsimonious sub-models presented in Section 2.1.

On the mth iteration in the E-step we calculate $t_{ik}^{(m)} := E[Z_{ik} \mid \mathbf{c}_1, \ldots, \mathbf{c}_n, \boldsymbol{\theta}^{(m-1)}]$ and $\nu_{ik}^{(m)} := E[\Upsilon_{ik} \mid \mathbf{Z}_1, \mathbf{c}_1, \ldots, \mathbf{Z}_n, \mathbf{c}_n, \boldsymbol{\theta}^{(m-1)}]$ (Amovin-Assagba et al, 2022). In the first CM-step on the mth iteration of the ECM algorithm we calculate the values in $\boldsymbol{\Psi}_1^{(m)}$ with $\boldsymbol{\Psi}_2$ fixed at $\boldsymbol{\Psi}_2^{(m-1)}$. We have

(Amovin-Assagba et al, 2022)

$$\pi_k^{(m)} = \frac{\sum_{i=1}^n t_{ik}^{(m)}}{n}, \quad \alpha_k^{(m)} = \frac{\sum_{i=1}^n t_{ik}^{(m)} \nu_{ik}^{(m)}}{\sum_{i=1}^n t_{ik}^{(m)}} \tag{8}$$
$$\boldsymbol{\mu}_k^{(m)} = \frac{\sum_{i=1}^n t_{ik}^{(m)} \left(\nu_{ik}^{(m)} + \frac{1-\nu_{ik}^{(m)}}{\eta_k^{(m-1)}}\right) \boldsymbol{c}_i}{\sum_{i=1}^n t_{ik}^{(m)} \left(\nu_{ik}^{(m)} + \frac{1-\nu_{ik}^{(m)}}{\eta_k^{(m-1)}}\right)}$$

As in Punzo and McNicholas (2016), we introduce a value α^* and we constrain $\alpha_k \in (\alpha^*, 1)$. If $\alpha_k^{(m)}$ in (8) is less than α^*, we use the *optimize*() function in the *stats* package in R to do a numerical search for $\alpha_k^{(m)}$ that gives the maximum of

$$\sum_{i=1}^n t_{ik}^{(m)} \left(\nu_{ik}^{(m)} \log(\alpha_k) + (1 - \nu_{ik}^{(m)}) \log(1 - \alpha_k)\right)$$

with respect to α_k over the interval $(\alpha^*, 1)$. In the applications presented in Section 4 we consider that less than half of the observations are outliers, and we empirically fix $\alpha^* \in (0.5, 1)$.

Let us define the sample covariance matrix of the cluster k by

$$\boldsymbol{H}_k^{(m)} = \frac{1}{\sum_{i=1}^n t_{ik}^{(m)}} \sum_{i=1}^n t_{ik}^{(m)} \left(\nu_{ik}^{(m)} + \frac{1-\nu_{ik}^{(m)}}{\eta_k^{(m-1)}}\right) (\boldsymbol{c}_i - \boldsymbol{\mu}_k^{(m)})(\boldsymbol{c}_i - \boldsymbol{\mu}_k^{(m)})^\top$$

For the model FCLM$[a_{kj}, b_k, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$ we get the updated values $a_{kj}^{(q)}, b_k^{(q)}, q_{kj}^{(q)}, k = 1, \ldots, K, j = 1, \ldots, d_k$ using the sample covariance matrix $\boldsymbol{H}_k^{(m)}$ of cluster k and the matrix of inner products between the basis functions $\boldsymbol{W}$ (Amovin-Assagba et al, 2022)

- $\boldsymbol{q}_{kj}^{(m)}$, $k = 1, \ldots, K, j = 1, \ldots, d_k$ are updated as the eigenfunctions associated with the d_k largest eigenvalues of $\boldsymbol{W}^{1/2} \boldsymbol{H}_k^{(m)} \boldsymbol{W}^{1/2}$;
- $a_{kj}^{(m)}$, $k = 1, \ldots, K, j = 1, \ldots, d_k$ are updated by the d_k largest eigenvalues of $\boldsymbol{W}^{1/2} \boldsymbol{H}_k^{(m)} \boldsymbol{W}^{1/2}$;
- $b_k^{(m)}$, $k = 1, \ldots, K$ are updated by

$$b_k^{(m)} = \frac{1}{R - d_k} \left(\text{trace}\left(\boldsymbol{W}^{1/2} \boldsymbol{H}_k^{(m)} \boldsymbol{W}^{1/2}\right) - \sum_{j=1}^{d_k} a_{kj}^{(m)}\right) \tag{9}$$

Proceeding as in Bouveyron et al (2007) we extend these results for the simplified models:

- FCLM$[a_{kj}, b, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$: the estimator of b is

$$b^{(m)} = \frac{\text{trace}\left(\sum_{k=1}^{K} \pi_k^{(m)} \boldsymbol{W}^{1/2} \boldsymbol{H}_k^{(m)} \boldsymbol{W}^{1/2}\right) - \sum_{k=1}^{K} \pi_k^{(m)} \sum_{j=1}^{d_k} a_{kj}^{(m)}}{R - \sum_{k=1}^{K} \pi_k^{(m)} d_k} \tag{10}$$

- FCLM$[a_k, b_k, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$: the estimator of a_k is

$$a_k^{(m)} = \frac{\sum_{j=1}^{d_k} a_{kj}^{(m)}}{d_k} \tag{11}$$

  and the estimator of b_k is given by (9).
- FCLM$[a, b_k, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$: the estimator of a is

$$a^{(m)} = \frac{\sum_{k=1}^{K} \pi_k^{(m)} \sum_{j=1}^{d_k} a_{kj}^{(m)}}{\sum_{k=1}^{K} \pi_k^{(m)} d_k} \tag{12}$$

  and the estimator of b_k is given by (9).
- FCLM$[a_k, b, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$: the estimator of a_k is given by (11) and the estimator of b is given by (10).
- FCLM$[a, b, \boldsymbol{Q}_k, d_k, \alpha_k, \eta_k]$: the estimator of a is given by (12) and the estimator of b is given by (10).

In the second CM-step on the mth iteration we calculate the values $\eta_k^{(m)}$ with $\boldsymbol{\Psi}_1$ fixed at $\boldsymbol{\Psi}_1^{(m)}$ (Amovin-Assagba et al, 2022):

$$\eta_k^{(m)} = \max\left\{1, \frac{\sum_{i=1}^{n} t_{ik}^{(m)}(1-\nu_{ik}^{(m)})(\boldsymbol{c}_i - \boldsymbol{\mu}_k^{(m)})^\top \left(\boldsymbol{\Sigma}_k^{(m)}\right)^{-1} (\boldsymbol{c}_i - \boldsymbol{\mu}_k^{(m)})}{R\sum_{i=1}^{n} t_{ik}^{(m)}(1-\nu_{ik}^{(m)})}\right\},$$

$$\left(\boldsymbol{\Sigma}_k^{(m)}\right)^{-1} = \boldsymbol{W}^{1/2}\boldsymbol{Q}_k^{(m)}\left(\boldsymbol{D}_k^{(m)}\right)^{-1}\left(\boldsymbol{Q}_k^{(m)}\right)^\top \boldsymbol{W}^{1/2}.$$

3.2 Inference for the T-funHDDC models

To fit the models, we use the EM algorithm. We have two sources of missing data: the clusters' labels $\boldsymbol{Z}_i$ and the un-observed values H_i. Thus, the complete data are given by $\{\boldsymbol{c}_i, z_{ik}, h_i, i = 1, \ldots, n, k = 1, \ldots, K\}$.

Proposition 1 *The complete data log-likelihood of the observed curves under the FTLM$[a_{kj}, b_k, \boldsymbol{Q}_k, d_k]$ model can be written as*

$$l_c(\boldsymbol{\theta}) = l_{1c}(\pi) + l_{2c}(\nu) + l_{3c}(\boldsymbol{\vartheta}) \tag{13}$$

where

$$l_{1c}(\pi) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\log(\pi_k), \tag{14}$$

$$l_{2c}(\nu) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\left(-\log\left(\Gamma\left(\frac{\nu_k}{2}\right)\right) + \frac{\nu_k}{2}\log(\frac{\nu_k}{2}) + \frac{\nu_k}{2}\left(\log(h_i) - h_i\right) - \log(h_i)\right), \tag{15}$$

$$l_{3c}(\boldsymbol{\vartheta}) = -\frac{nR\log(2\pi)}{2} + \frac{n}{2}\log(\mid \boldsymbol{W} \mid) - \frac{1}{2}\sum_{k=1}^{K} n_k \sum_{l=1}^{d_k}\log(a_{kl}) - \frac{1}{2}\sum_{k=1}^{K} n_k$$
$$\sum_{l=d_k+1}^{R}\log(b_k) - \frac{1}{2}\sum_{k=1}^{K}\left(\sum_{l=1}^{d_k}\frac{\boldsymbol{q}_{kl}^{\top}\boldsymbol{W}^{1/2}\boldsymbol{S}_k\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{a_{kl}} + \sum_{l=d_k+1}^{R}\frac{\boldsymbol{q}_{kl}^{\top}\boldsymbol{W}^{1/2}\boldsymbol{S}_k\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{b_k}\right), \tag{16}$$

where $\boldsymbol{\vartheta} = \{\boldsymbol{\mu}_k, a_{kj}, b_k, \boldsymbol{q}_{kj}\}$, $k = 1, \ldots, K$, $j = 1, \ldots, d_k$, *with* $\boldsymbol{q}_{kj}$ *the* j*th column of* $\boldsymbol{Q}_k$, $n_k = \sum_{i=1}^{n} z_{ik}$, *and* $\boldsymbol{S}_k$ *is defined by*

$$\boldsymbol{S}_k := \sum_{i=1}^{n} z_{ik} h_i (\boldsymbol{c}_i - \boldsymbol{\mu}_k)(\boldsymbol{c}_i - \boldsymbol{\mu}_k)^{\top}. \tag{17}$$

Next we present the EM algorithm for the most general model FTLM$[a_{kj}, b_k, \boldsymbol{Q}_k, d_k, \nu_k]$.

3.2.1 The E-step

At the mth iteration of the EM algorithm we calculate $E[l_c(\boldsymbol{\theta}^{(m-1)}) \mid \boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{\theta}^{(m-1)}]$, given the current values of the parameters $\boldsymbol{\theta}^{(m-1)}$. As in Section 7.5 in McLachlan and Peel (2004), this reduces to the calculation of $E[Z_{ik} \mid \boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{\theta}^{(m-1)}]$, $E[H_i \mid Z_{ik} = 1, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{\theta}^{(m-1)}]$ and $E[\log(H_i) \mid Z_{ik} = 1, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{\theta}^{(m-1)}]$.

Proposition 2 *For the model FTLM*$[a_{kj}, b_k, \boldsymbol{Q}_k, d_k]$ *the density of the multivariate t distribution (7) for the kth cluster can be written as*

$$f_t(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k) = \frac{\Gamma(\frac{\nu_k+p}{2})\left(\prod_{l=1}^{d_k} a_{kl}\prod_{l=d_k+1}^{p} b_k\right)^{-1/2} \mid \boldsymbol{W} \mid^{1/2}}{(\pi\nu_k)^{R/2}\Gamma(\frac{\nu_k}{2})\left(1 + \frac{\delta(\boldsymbol{c}_i;\ \boldsymbol{\mu}_k, \boldsymbol{Q}_k, a, b, d_k)}{\nu_k}\right)^{\frac{\nu_k+R}{2}}} \tag{18}$$

where we denote

$$\delta(\boldsymbol{c}_i;\ \boldsymbol{\mu}_k, \boldsymbol{Q}_k, a, b, d_k) := \sum_{l=1}^{d_k}\frac{\boldsymbol{q}_{kl}^{\top}\boldsymbol{W}^{1/2}(\boldsymbol{c}_i - \boldsymbol{\mu}_k)(\boldsymbol{c}_i - \boldsymbol{\mu}_k)^{\top}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{a_{kl}}$$
$$+ \sum_{l=d_k+1}^{p}\frac{\boldsymbol{q}_{kl}^{\top}\boldsymbol{W}^{1/2}(\boldsymbol{c}_i - \boldsymbol{\mu}_k)(\boldsymbol{c}_i - \boldsymbol{\mu}_k)^{\top}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{b_k} \tag{19}$$

We also have

$$t_{ik}^{(m)} := E[Z_{ik} \mid \boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{\theta}^{(m-1)}] = \frac{\pi_k f_t\left(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k^{(m-1)}\right)}{\sum_{l=1}^{K}\pi_l f_t\left(\boldsymbol{c}_i;\ \boldsymbol{\theta}_l^{(m-1)}\right)} \tag{20}$$

$$h_{ik}^{(m)} := E[H_i \mid Z_{ik} = 1, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{\theta}^{(m-1)}]$$

$$= \frac{\nu_k^{(m-1)} + R}{\nu_k^{(m-1)} + \delta(\boldsymbol{c}_i;\ \boldsymbol{\mu}_k^{(m-1)}, \boldsymbol{Q}_k^{(m-1)}, a^{(m-1)}, b^{(m-1)}, d_k)} \tag{21}$$

$$\log(h_{ik})^{(m)} := E[\log(H_i) \mid Z_{ik} = 1, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{\theta}^{(m-1)}]$$

$$= \log(h_{ik}^{(m)}) + \Psi\left(\frac{\nu_k^{(m-1)} + R}{2}\right) - \log\left(\frac{\nu_k^{(m-1)} + R}{2}\right), \tag{22}$$

where $\Psi(\cdot)$ *is the digamma function:*

$$\Psi(s) = \frac{d(\log\Gamma(s))}{ds} = \frac{d\Gamma(s)/ds}{\Gamma(s)}. \tag{23}$$

Based on the current values of the parameters $\boldsymbol{\theta}^{(m-1)}$ the log-likelihood is given by

$$\log(L^{(m-1)}) = \log\left(\prod_{i=1}^{n} p(\boldsymbol{c}_i;\ \boldsymbol{\theta}^{(m-1)})\right) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \pi_k^{(m-1)} f_t(\boldsymbol{c}_i;\ \boldsymbol{\theta}_k^{(m-1)})\right)$$

3.2.2 The M-step

In the M-step at the mth iteration of the EM algorithm we estimate the parameters by maximizing the conditional expectation of the complete data log likelihood $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m-1)}) := E[\log(l_c(\boldsymbol{\theta}^{(m-1)})) \mid \boldsymbol{c}_1, \ldots, \boldsymbol{c}_n, \boldsymbol{\theta}^{(m-1)}]$.

Proposition 3 *For the model FTLM*$[a_{kj}, b_k, \boldsymbol{Q}_k, d_k, \nu_k]$ *we have the following updates for the parameters,* $k = 1, \ldots, K$

$$\pi_k^{(m)} = \frac{\sum_{i=1}^{n} t_{ik}^{(m)}}{n} = \frac{n_k^{(m)}}{n}, \quad n_k^{(m)} = \sum_{i=1}^{n} t_{ik}^{(m)} \tag{24}$$

$$\boldsymbol{\mu}_k^{(m)} = \frac{\sum_{i=1}^{n} t_{ik}^{(m)} h_{ik}^{(m)} \boldsymbol{c}_i}{\sum_{i=1}^{n} t_{ik}^{(m)} h_{ik}^{(m)}} \tag{25}$$

$\nu_k^{(m)}$ *is a solution of the equation*

$$1 - \Psi\left(\frac{\nu_k^{(m)}}{2}\right) + \frac{1}{n_k^{(m)}} \sum_{i=1}^{n} t_{ik}^{(m)} (\log(h_{ik}^{(m)}) - h_{ik}^{(m)})$$

$$+ \log\left(\frac{\nu_k^{(m)}}{2}\right) + \Psi\left(\frac{\nu_k^{(m-1)} + R}{2}\right) - \log\left(\frac{\nu_k^{(m-1)} + R}{2}\right) = 0 \tag{26}$$

If we consider the degrees of freedom to be the same for all groups, then an update for $\nu^{(m)}$ *can be found by solving numerically the equation*

$$1 - \Psi\left(\frac{\nu^{(m)}}{2}\right) + \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik}^{(m)} (\log(h_{ik}^{(m)}) - h_{ik}^{(m)})$$

$$+ \log\left(\frac{\nu^{(m)}}{2}\right) + \Psi\left(\frac{\nu^{(m-1)} + R}{2}\right) - \log\left(\frac{\nu^{(m-1)} + R}{2}\right) = 0 \tag{27}$$

Let

$$\boldsymbol{S}_k^{(m)} = \frac{\sum_{i=1}^{n} t_{ik}^{(m)} h_{ik}^{(m)} (\boldsymbol{c}_i - \boldsymbol{\mu}_k^{(m)})(\boldsymbol{c}_i - \boldsymbol{\mu}_k^{(m)})^\top}{n_k^{(m)}}. \tag{28}$$

Then

- $\boldsymbol{q}_{kj}^{(m)}$, $k = 1, \ldots, K, j = 1, \ldots, d_k$ *are updated as the eigenfunctions associated with the* d_k *largest eigenvalues of* $\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}$;
- $a_{kj}^{(m)}$, $k = 1, \ldots, K, j = 1, \ldots, d_k$ *are updated by the* d_k *largest eigenvalues of* $\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}$;
- $b_k^{(m)}$, $k = 1, \ldots, K$ *are updated by*

$$b_k^{(m)} = \frac{1}{R - d_k}\left(trace\left(\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}\right) - \sum_{j=1}^{d_k} a_{kj}^{(m)}\right) \tag{29}$$

For the simplified models we have:

- *FTLM*$[a_{kj}, b, \boldsymbol{Q}_k, d_k, \nu_k]$*: the estimator of* b *is*

$$b^{(m)} = \frac{trace\left(\sum_{k=1}^{K} \pi_k^{(m)} \boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}\right) - \sum_{k=1}^{K} \pi_k^{(m)} \sum_{j=1}^{d_k} a_{kj}^{(m)}}{R - \sum_{k=1}^{K} \pi_k^{(m)} d_k} \tag{30}$$

- *FTLM*$[a_k, b_k, \boldsymbol{Q}_k, d_k, \nu_k]$*: the estimator of* a_k *is*

$$a_k^{(m)} = \frac{\sum_{j=1}^{d_k} a_{kj}^{(m)}}{d_k} \tag{31}$$

  and the estimator of b_k *is given by* (29).
- *FTLM*$[a, b_k, \boldsymbol{Q}_k, d_k, \nu_k]$*: the estimator of* a *is*

$$a^{(m)} = \frac{\sum_{k=1}^{K} \pi_k^{(m)} \sum_{j=1}^{d_k} a_{kj}^{(m)}}{\sum_{k=1}^{K} \pi_k^{(m)} d_k} \tag{32}$$

  and the estimator of b_k *is given by* (29).
- *FTLM*$[a_k, b, \boldsymbol{Q}_k, d_k, \nu_k]$*: the estimator of* a_k *is given by* (31) *and the estimator of* b *is given by* (30).
- *FTLM*$[a, b, \boldsymbol{Q}_k, d_k, \nu_k]$*: the estimator of* a *is given by* (32) *and the estimator of* b *is given by* (30).

In the implementation we use the *uniroot* function in the *stats* package in R to solve for updates of ν_k numerically, and we restrict these values between 2 and 200.

3.2.3 Initialization

To start the EM algorithm, we need initial values $t_{ik}^{(0)}$. As for funHDDC (Schmutz et al, 2020), we have implemented an initialization with the *kmeans* method available in the *stats* package in R. We also consider a second approach, based on the function *tkmeans* from the R package *tclust*, which is an implementation of the trimmed k-means method in Cuesta-Albertos et al (1997). This robust initialization method requires the proportion of data to be trimmed. As in Amovin-Assagba et al (2022), to ensure that we are not contaminated by the presence of outliers, we consider a large proportion, and we fix this parameter to 0.2.

Initialization is critical for preventing the convergence of the EM algorithm to a local maximum, so we execute the algorithm several times with different initialization values for $t_{ik}^{(0)}$, and we keep the best result given by the EM algorithm using the Bayesian information criterion (BIC; Schwarz, 1978) defined below in (33). In the applications we consider the number of initializations to be at least 20.

The degrees of freedom are initialized as $\nu_k^{(0)} = 50$ (Andrews and McNicholas, 2011), and the rest of the parameters and $h_{ik}^{(0)}$ are initialized as per the updates in Propositions 2 and 3.

3.2.4 Estimation of the hyper-parameters

The hyper-parameters K and d_k, $k = 1, \ldots, K$ are not estimated by the EM algorithm. As in Bouveyron and Jacques (2011), the group specific dimension d_k is selected through the Cattell scree-test by comparison of the difference between eigenvalues with a given threshold. Alternatively, we can do a grid search and choose d_k as the positive integer from the grid that corresponds to the maximum value of the BIC (Amovin-Assagba et al, 2022). We have tried both methods for the applications presented here, and we have not obtained a major improvement using the grid search, but the computational time was substantially larger. In the next section we present the results obtained with the Cattell scree-test.

The number of clusters K as well as the parsimonious model are selected using the BIC criterion: we maximize the BIC defined as

$$BIC = L^{(m_f)} - \frac{\tau}{2} \log n, \tag{33}$$

where τ is the overall number of the free parameters, n is the number of observations, $L^{(m_f)}$ is the maximum log-likelihood value, and m_f is the last iteration of the algorithm before convergence.

3.2.5 Convergence criterion and the classification step

After initialization, the E and M steps are alternated until convergence. The EM algorithm is considered to have converged if a maximum number of iterations is reached or $L_\infty^{(m+2)} - L^{(m+1)} < \epsilon$, provided that this difference is positive (McNicholas et al, 2010). We choose 200 for the maximum number of iterations and $\epsilon = 10^{-6}$. Here $L^{(m+1)}$ is the log-likelihood value from iteration $m+1$ and $L_\infty^{(m+2)}$ is the asymptotic estimate of log-likelihood at iteration $m+2$ (Andrews et al, 2011) given by

$$L_\infty^{(m+2)} = L^{(m+1)} + \frac{L^{(m+2)} - L^{(m+1)}}{1 - a^{(m+1)}},$$

where $a^{(m+1)}$ is the Aitken acceleration (Aitken, 1927) at iteration $m+1$:

$$a^{(m+1)} = \frac{L^{(m+2)} - L^{(m+1)}}{L^{(m+1)} - L^{(m)}}$$

At the end of the EM algorithm, we do a classification step to provide the expected clustering. We determine the cluster using the maximum *a posteriori* (MAP) rule: an observation $\mathbf{c}_i$ is assigned to the cluster $k \in \{1, \ldots, K\}$ with the largest $t_{ik}^{(m_f)}$, where m_f is the last iteration of the algorithm before convergence.

4 Applications

We first apply T-funHDDC, C-funHDDC, and funHDDC methods to clustering simulated functional data with outliers. We consider one- and two-dimensional curves with various outlier contamination scenarios. Next we compare T-funHDDC with competing algorithms for clustering the NOx data representing daily curves of Nitrogen Oxides (NOx) emissions in the neighborhood of the industrial area of Poblenou, Barcelona (Spain). Finally, we use T-funHDDC for clustering traffic flow data in Edmonton, Canada.

Although analyses for the simulated data are conducted as clustering examples, the true classifications are actually known. In these examples, the Adjusted Rand Index (ARI; Hubert and Arabie, 1985) is used to measure the accuracy of the classification. The expected value of the adjusted Rand index is 0, and for a perfect classification its value is 1.

4.1 Simulation Study-Univariate curves

Similarly with Anton and Smith (2023) we simulate 1000 curves based on the model $FCLM[a_k, b_k, \mathbf{Q}_k, d_k, \alpha_k, \eta_k]$. The number of clusters is fixed to $K = 3$ and the mixing proportions are equal $\pi_1 = \pi_2 = \pi_3 = 1/3$. We consider the following values of the parameters

Group 1: $d = 5$, $a = 150$, $b = 5$, $\mu = (1, 0, 50, 100, 0, \ldots, 0)$

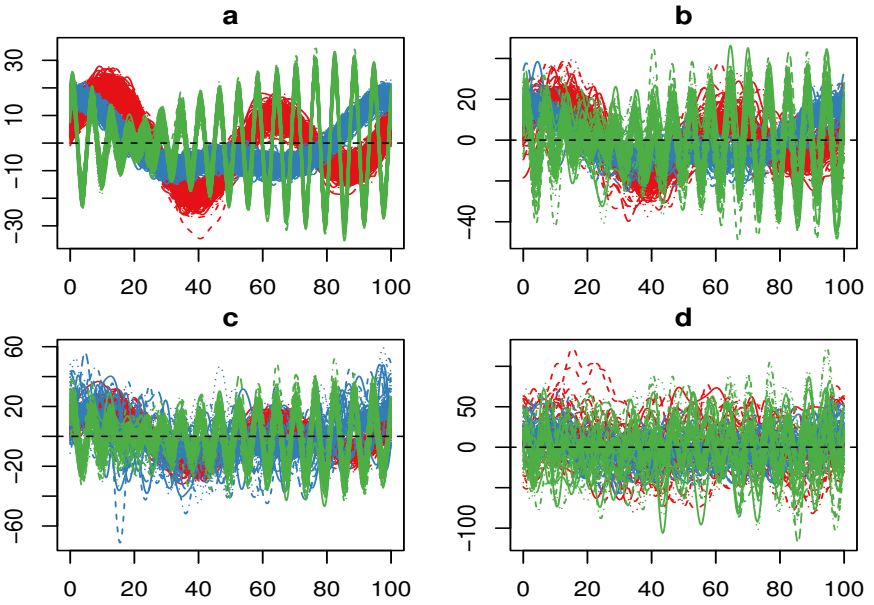

Fig. 2 Smooth data simulated without oultiers (a) according to scenario A (b), scenario B (c), and scenario C (d), colored by group for one simulation.

Group 2: $d = 20$, $a = 15$, $b = 8$, $\mu = (0, 0, 80, 0, 40, 2, 0, \ldots, 0)$
Group 3: $d = 10$, $a = 30$, $b = 10$, $\mu = (0, \ldots, 0, 20, 0, 80, 0, 0, 100)$,

where d is the intrinsic dimension of the subgroups, μ is the mean vector of size 70, a is the values of the d-first diagonal elements of $\mathbf{D}$, and b the value of the last $70-d$- elements. Curves as smoothed using 35 Fourier basis functions. We repeat the simulation setting 100 times. A sample of these data is plotted in Fig. 2 a.

We consider the following contamination schemes.

Scenario A: Very little contamination. Scores are simulated from contaminated normal distributions with the previous parameters, $\alpha_i = 0.9$, $i = 1, \ldots, 3$, and $\eta_1 = 10$, $\eta_2 = 7$, $\eta_3 = 17$.
Scenario B: Medium contamination. Scores are simulated from contaminated normal distributions with the previous parameters, $\alpha_i = 0.9$, $i = 1, \ldots, 3$, and $\eta_1 = 5$, $\eta_2 = 50$, $\eta_3 = 15$.
Scenario C: High contamination. Scores are simulated from contaminated normal distributions with the previous parameters, $\alpha_i = 0.9$, $i = 1, \ldots, 3$, and $\eta_1 = 100$, $\eta_2 = 70$, $\eta_3 = 170$.

Samples for data generated according to scenarios A, B, C are plotted in Figs. 2 b, c, d, respectively. We notice that the clustering problem becomes more difficult when we increase the values of η, such as scenario C where we have a lot of overlapping between the 3 groups due to outliers.

Table 1 Mean (and standard deviation) of ARI for BIC best model on 100 simulations. Bold values indicate the highest value for each method.

Scenario	Method	ϵ	ARI	ARI Outliers
A	FunHDDC	0.05	**0.519 (0.11)**	-
A	FunHDDC	0.1	0.499(0.05)	-
A	FunHDDC	0.2	0.494 (0.01)	-
A	C-funHDDC	0.05	0.769 (0.23)	0.959(0.04)
A	C-funHDDC	0.1	0.986(0.08)	0.998(0.01)
A	C-funHDDC	0.2	**0.9995 (0.001)**	**1 (0)**
A	tfunHDDC	0.05	0.964 (0.120)	-
A	tfunHDDC	0.1	0.995(0.0438)	-
A	tfunHDDC	0.2	**0.9996(0.001)**	-
B	FunHDDC	0.05	**0.861 (0.23)**	-
B	FunHDDC	0.1	0.754(0.25)	-
B	FunHDDC	0.2	0.52 (0.09)	-
B	C-funHDDC	0.05	0.807 (0.22)	0.961(0.05)
B	C-funHDDC	0.1	0.948 (0.14)	0.99(0.03)
B	C-funHDDC	0.2	**0.990 (0.062)**	**0.971 (0.149)**
B	tfunHDDC	0.05	0.892 (0.188)	-
B	tfunHDDC	0.1	0.97(0.111)	-
B	tfunHDDC	0.2	**0.991 (0.06)**	-
C	FunHDDC	0.05	0.490 (0.02)	-
C	FunHDDC	0.1	0.491(0.02)	-
C	FunHDDC	0.2	**0.494 (0.01)**	-
C	C-funHDDC	0.05	0.736 (0.23)	0.928(0.10)
C	C-funHDDC	0.1	0.911 (0.18)	0.958(0.15)
C	C-funHDDC	0.2	**0.965 (0.11)**	**0.994 (0.03)**
C	tfunHDDC	0.05	0.588 (0.137)	-
C	tfunHDDC	0.1	0.678(0.205)	-
C	tfunHDDC	0.2	**0.892 (0.18)**	-

The quality of the estimated partitions obtained using funHDDC, C-funHDDC, and T-funHDDC is evaluated using the ARI, and the results are included in Table 1. For funHDDC we use the package *funHDDC* in R. We run the three algorithms for $K = 3$ with all 6 sub-models, and the best solution in terms of the highest BIC value for all those sub-models is returned. The initialization is done with the *kmeans* strategy with 50 repetitions, and the maximum number of iterations is 200 for the stopping criterion. We use $\epsilon \in \{0.05, 0.1, 0.2\}$ in the Cattell test. For C-funHDDC we use $\alpha^* = 0.75$, and for T-funHDDC we do not constrain the degrees of freedom for different groups to be equal.

We notice that C-funHDDC and T-funHDDC give similar results, and they give excellent results even for Scenario C. They both outperform funHDDC. C-funHDDC also precisely identifies the outliers, and in Table 2 we report the means, medians, and standard deviations for estimations of η_i and α_i, $i = 1, 2, 3$ for the 100 tests done with $\epsilon = 0.2$ for each scenario. The accuracy of these estimates decreases when the accuracy of the classification decreases, but we get good results for the medians even for Scenario C.

Table 2 Medians, means, and standard deviations for estimations for η_i and α_i, $i = 1, 2, 3$ for 100 simulations for BIC best model with $\epsilon = 0.2$ in the Cattell test.

Scenario A	$\eta_1 = 10$	$\eta_2 = 7$	$\eta_3 = 17$	$\alpha_1 = 0.9$	$\alpha_2 = 0.9$	$\alpha_3 = 0.9$
median	10.11	7.01	17.07	0.90	0.90	0.90
mean	10.09	7.00	17.05	0.90	0.90	0.90
st.dev	0.50	0.35	0.78	0.0003	0.0003	0.0003
Scenario B	$\eta_1 = 5$	$\eta_2 = 50$	$\eta_3 = 15$	$\alpha_1 = 0.9$	$\alpha_2 = 0.9$	$\alpha_3 = 0.9$
median	5.03	49.99	15.09	0.90	0.90	0.90
mean	5.06	68.70	15.23	0.90	0.90	0.90
st.dev	0.29	133.74	1.64	0.001	0.0005	0.0003
Scenario C	$\eta_1 = 100$	$\eta_2 = 70$	$\eta_3 = 170$	$\alpha_1 = 0.9$	$\alpha_2 = 0.9$	$\alpha_3 = 0.9$
median	98.26	69.14	175.76	0.90	0.90	0.90
mean	97.21	69.42	233.49	0.90	0.90	0.91
st.dev	20.79	18.32	195.95	0.0007	0.0007	0.0009

4.2 Simulation Study: Bivariate Curves

Using a triangle model inspired by Bouveyron and Jacques (2011), we simulate 400 bivariate curves according to the following model:

Group 1:

$X_1(t) = U + (0.6 - U)H_1(t) + \epsilon_1(t)$

$X_2(t) = U + (0.5 - U)H_1(t) + \epsilon_1(t)$

Contaminated $X_1(t) = \sin(t) + (0.6 - U)H_1(t) + \epsilon_2(t)$

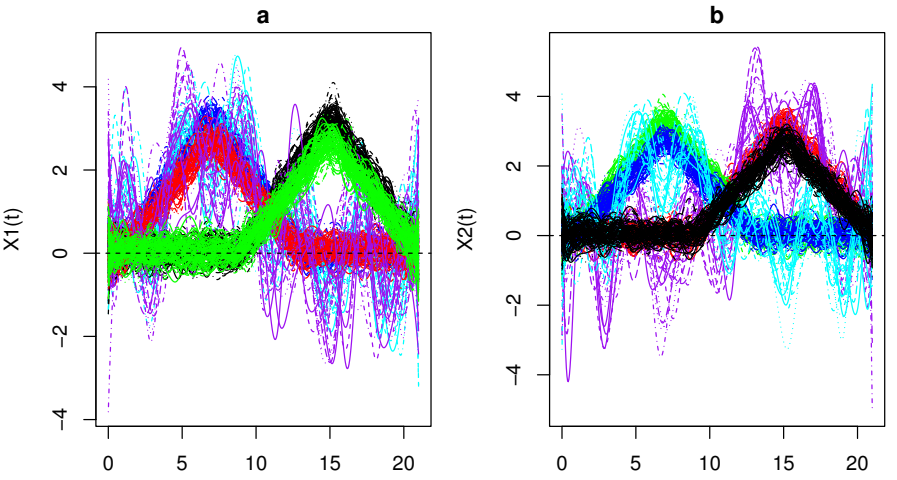

Fig. 3 Smooth data simulated for X_1 (a) and X_2 (b) . Group 1 (blue) and outliers (cyan), group 2 (black), group 3 (red) and outliers (purple), and group 4 (green).

Contaminated $X_2(t) = \sin(t) + (0.5 - U)H_1(t) + \epsilon_2(t)$

Group 2:

$X_1(t) = U + (0.6 - U)H_2(t) + \epsilon_1(t)$

$X_2(t) = U + (0.5 - U)H_2(t) + \epsilon_1(t)$

Group 3:

$X_1(t) = U + (0.5 - U)H_1(t) + \epsilon_1(t)$

$X_2(t) = U + (0.6 - U)H_2(t) + \epsilon_1(t)$

Contaminated $X_1(t) = \sin(t) + (0.5 - U)H_1(t) + \epsilon_3(t)$

Contaminated $X_2(t) = \sin(t) + (0.6 - U)H_2(t) + \epsilon_3(t)$

Group 4:

$X_1(t) = U + (0.5 - U)H_2(t) + \epsilon_1(t)$

$X_2(t) = U + (0.6 - U)H_1(t) + \epsilon_1(t)$.

Here $t \in [1, 21]$, $H_1(t) = (6 - |t - 7|)_+$, and $H_2(t) = (6 - |t - 15|)_+$, with $(\cdot)_+$ representing the positive part. $U \sim \mathcal{U}(0, 0.1)$, and $\epsilon_1(t) \sim N(0, 0.5)$, $\epsilon_2(t) \sim N(0, 2)$, $\epsilon_3(t) \sim Cauchy(0, 4)$ are mutually independent white noises and independent of U. We use $\sin(t)$ as a behavioral change that retains group membership, and $\epsilon_2(t)$ and $\epsilon_3(t)$ represent noises that have larger variances in contaminated groups. We simulate 100 curves for each group, groups 1 and 3 consisting of 80 ordinary curves and 20 contaminated curves. Curves are smoothed using a 25 cubic B-spline basis, and we repeat the simulation 100 times. A single simulation of these data is plotted in Fig. 3. We can notice

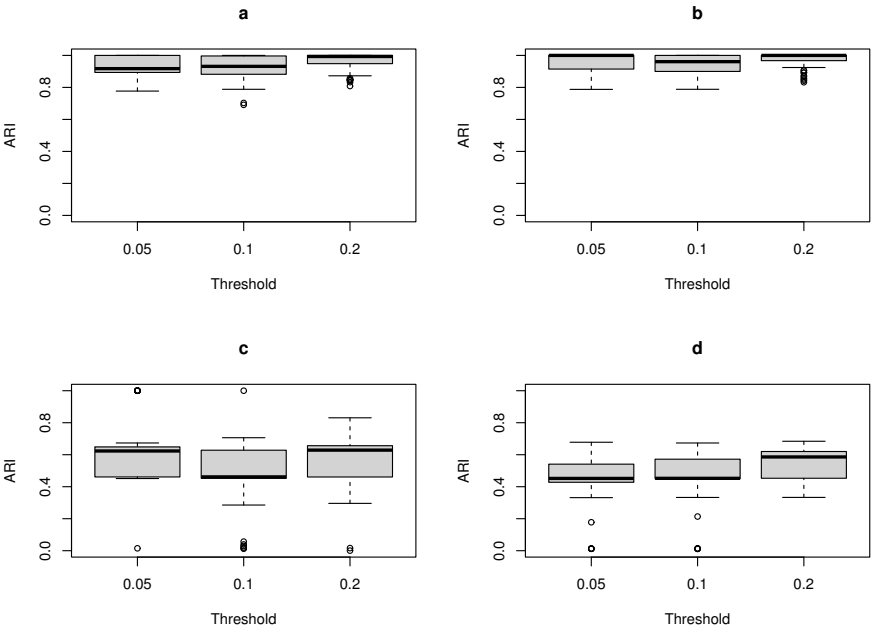

Fig. 4 ARI for T-funHDDC using *kmeans* (a) or *tkmeans* (b) initialization, and ARI for C-funHDDC using *kmeans* (c) or *tkmeans* (d) initialization

that any clustering method applied only to $X_1(t)$ should fail because groups 1 (blue) and 3 (red) are similar, and group 2 (black) is similar to group 4 (green). For $X_2(t)$, groups 1 (blue) and 4 (green) are similar, and group 2 (black) is similar to group 3 (red), so no univariate clustering method would succeed. The results in Table 3 show that there is a clear improvement in the clustering

Table 3 Mean (and standard deviation) of ARI of all methods applied to simulated triangles bivariate data for $K = 4$ and $K = 5$ clusters. Bold values indicate the highest value for each method.

Method	Initialization	ϵ	ARI for $K = 4$	ARI for $K = 5$
FunHDDC	kmeans	0.05	0.690(0.044)	0.828 (0.099)
FunHDDC	kmeans	0.1	**0.691(0.044)**	0.830 (0.083)
FunHDDC	kmeans	0.2	0.682(0.051)	**0.831 (0.117)**
T-funHDDC	kmeans	0.05	0.935 (0.068)	0.805 (0.058)
T-funHDDC	kmeans	0.1	0.915 (0.073)	0.842 (0.063)
T-funHDDC	kmeans	0.2	0.981 (0.043)	0.878 (0.037)
T-funHDDC	tkmeans	0.05	0.971 (0.042)	0.840 (0.051)
T-funHDDC	tkmeans	0.1	0.929 (0.064)	**0.887 (0.055)**
T-funHDDC	tkmeans	0.2	**0.987 (0.031)**	0.886 (0.032)
C-funHDDC	kmeans	0.05	**0.616 (0.182)**	0.971 (0.000)
C-funHDDC	kmeans	0.1	0.504 (0.192)	0.941 (0.083)
C-funHDDC	kmeans	0.2	0.556 (0.129)	0.735 (0.117)
C-funHDDC	tkmeans	0.05	0.476 (0.128)	**0.998 (0.015)**
C-funHDDC	tkmeans	0.1	0.466 (0.111)	0.977 (0.084)
C-funHDDC	tkmeans	0.2	0.548 (0.083)	0.801 (0.158)

results of T-funHDDC over funHDDC. Consistently funHDDC miss-classifies the outliers, mixing the separate groups of outliers into one group and clustering groups 1 and 3 together. We run the three algorithms for $K = 4$ with all 6 sub-models and the best solution in terms of the highest BIC value for all those sub-models is returned. For C-funHDDC we use $\alpha^* = 0.85$, and for T-funHDDC we do not constrain the degrees of freedom for different groups to be equal. The initialization is done using *kmeans* with 20 repetitions, and for T-funHDDC and C-funHDDC we also do the initialization using *tkmeans*. From the boxplots in Fig. 4, we can see that the trimming used by *tkmeans* improves T-funHDDC and C-funHDDC's consistency. Although both are robust methods T-funHDDC clearly clusters the triangles correctly and more consistently than C-funHDDC.

From Amovin-Assagba et al (2022) we know that when the proportion of outliers is large, C-funHDDC tends to group all the outliers into an additional cluster. To study the behavior of T-funHDDC, we consider that all the outliers form a fifth cluster and we also run the three algorithms for $K = 5$ clusters (the rest of the settings are the same as for $K = 4$). From the results in the last column of Table 3 we can see that T-funHDDC has a better performance for $K = 4$ clusters, so the heavy tailed multivariate t distribution is able to

handle the outliers and we get a very accurate separation into four clusters. In contrast, funHDDC and C-funHDDC put the outliers together in a separate cluster and perform much better for $K = 5$ than for $K = 4$. C-funHDDC gives excellent results for $K = 5$, better than funHDDC and T-funHDDC which have similar mean values for ARI.

4.3 Benchmark study-NOx levels data

We consider the NOx data available in the *fda.usc* library in R (Febrero-Bande and de la Fuente, 2012). The measurements of NOx (in $\mu g/m^3$) were taken hourly resulting in 76 curves for "working days" and 39 curves for "non-working days" (see Fig. 5 a). Since NOx is a contaminant agent, the detection of outlying emission is useful for environmental protection. This data set has been used for testing methods for the detection of outliers in functional data (Febrero-Bande et al, 2008, Sawant et al, 2012, Sguera et al, 2015) and to illustrate robust clustering based on trimming for functional data (Rivera-García et al, 2019).

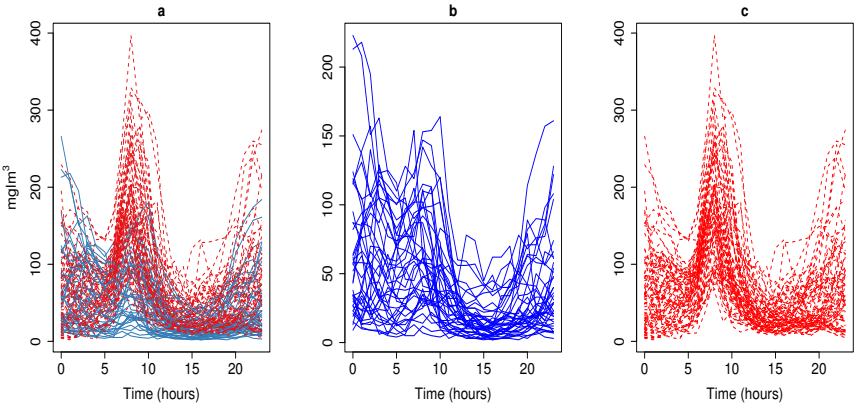

Fig. 5 a.DailyNOx curves for 115 days; b. c. Clustering obtained with T-funHDDC, $\epsilon = 0.6$, Non-working days (blue, plain line), working days (red, dashed line)

We apply T-funHDDC, C-funHDDC, funHDDC, CNmixt (Punzo and McNicholas, 2016), and teigen (Andrews and McNicholas, 2012) to the NOx data. Curves are smoothed using a B-spline basis of functions of order 3 with 15 basis elements, and we run the algorithms for $K = 2$ clusters. For T-funHDDC, C-funHDDC, and funHDDC we use $\epsilon \in \{0.05, 0.2, 0.4, 0.6\}$ in the Cattell test and we consider all 6 sub-models. The best solution in terms of the highest BIC value for all those sub-models is returned. The initialization is done with the *kmeans* strategy with 20 repetitions, and the maximum number of iterations is 200 for the stopping criterion. We run the CNmixt function

from the *ContaminatedMixt* R package for all 14 models , based on the coefficients in the B-spline basis of functions of order 3. Initialization is done with the *kmeans* function. We also run the teigen function from the *teigen* package in R based on the coefficients in the B-spline basis of functions of order 3, with models having constrained (CN) and unconstrained (UN) degrees of freedom. Initialization for teigen is done with the *kmeans* function. Assuming, as in Rivera-García et al (2019), that the correct groups were determined by working and non-working days, the correct classification rates (CCR) obtained with these five methods are reported in Table 4.

Table 4 Correct classification rates for each method for the NOx data

Method	ϵ	α^*	CCR	Method	ϵ	CN/UN	CCR
funHDDC	0.05	-	0.51	T-funHDDC	0.05	CN	0.52
funHDDC	0.2	-	0.71	T-funHDDC	0.2	CN	0.73
funHDDC	0.4	-	**0.76**	T-funHDDC	0.4	CN	0.78
funHDDC	0.6	-	0.70	T-funHDDC	0.6	CN	0.91
C-funHDDC	0.05	0.85	0.77	T-funHDDC	0.05	UN	0.52
C-funHDDC	0.2	0.85	**0.86**	T-funHDDC	0.2	UN	0.73
C-funHDDC	0.4	0.85	0.84	T-funHDDC	0.4	UN	0.78
C-funHDDC	0.6	0.85	0.84	T-funHDDC	0.6	UN	**0.91**
CNmixt	-	0.5	0.64	teigen	-	CN	0.56
CNmixt	-	0.85	0.64	teigen	-	UN	0.57

The CCRs for T-funHDDC and C-funHDDC are better than the ones for funHDDC, and the best CCR is 0.91 obtained with T-funHDDC. These results are comparable with the ones reported in Table 1 in Rivera-García et al (2019) for Funclust, RFC, and TrimK, with the best CCR equal with 0.84 and obtained with Funclust and RFC. In Figs. 5 b, c, we present the clusters obtained using T-funHDDC with $\epsilon = 0.6$ and unconstrained degrees of freedom.

We can notice that for the NOx data the two-steps methods CNmixt and teigen applied on the coefficients in the B-spline basis of functions of order 3 give the lowest CCRs. We have also applied them directly to the sampled NOx data, but the CCRs were even lower and there were convergence problems for several models.

4.4 Real data example-Traffic Speeding and Weather Conditions

We analyze traffic data from the City of Edmonton available from the Edmonton open data portal at https://data.edmonton.ca/stories/s/Speed-Surveys/kd7n-5iq3/, joined with historical weather data from the Edmonton International Airport available at https://climate.weather.gc.ca/climate_data/. We extract records for traffic counts going 5-10 km/h under the speed limit, and records for traffic counts going 0-5km/h over the speed limit, such that we have

a speeding differential of 5-15 km/h for analysis. We join temperature and visibility data by date as factors that affect the number of cars in each speeding category. Thus, we work with multivariate functional data with four components. The speeding data are organized in 15 minute intervals, so we have 96 time points for each curve in the two speeding components. The weather data is recorded in hour intervals, so we have 24 time points for each curve in the temperature and visibility components. Different trends in the amount of travel and speeds can be distinguished based on the time of day and the direction of the road (to or away from work areas). The data also have potential outliers from holidays, special events, and weather, all of which may cause more or less cars to appear in each speeding interval.

A thousand records are randomly sampled using the *sample* function from R and used for clustering with the T-funHDDC method. We use an hourly descriptive weather column (clouds, rain, fog, etc.) for the final analysis of clusters. The objective of this study is to find trends and explain how weather conditions affect speeding differentials among road sections in Edmonton, with clustering being the first step in a larger data mining analysis.

Data are fit with a Fourier basis with 12 basis functions. We use all the models in the T-funHDDC method and consider 2 to 10 clusters. A model with 10 clusters, unconstrained degrees of freedom, and the threshold for the Cattell test $\epsilon = 0.05$ was chosen based on BIC.

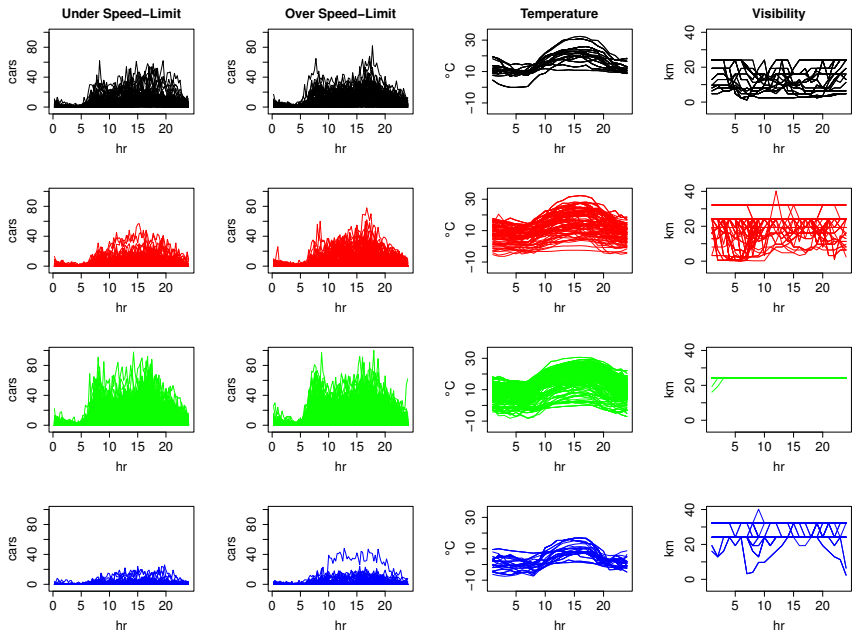

Fig. 6 Group 1 (first row, black) and group 2 (second row, red) have similar traffic speeding flow. Group 3 (third row, green) and group 9 (fourth row, blue) are unique groups affected by other conditions.

Table 5 Month distributions of the traffic/weather data clusters

Month	Cluster 1	Cluster 2	Cluster 3	Cluster 9
April	0	2	5	0
May	0	11	60	1
June	0	23	119	0
July	8	22	90	0
August	60	27	57	0
September	2	25	57	6
October	0	13	34	37
November	0	0	0	2

Four clusters, shown in Fig. 6, are of particular interest. We found that groups 1 and 2 reflect similar trends with the only difference being that group 1 has many low visibility "smoky" days in August (see Table 5), when smoke from wildfires moves through the region. The clustering indicates that while visibility is lowered by smoke, there is no change in the speeding behavior. Group 3 represents a different behavior with cloudy weather conditions, where speeding spikes in the morning and afternoon (to and from work). Group 9 is composed of snowy days in September, October, and November when snow starts to fall in the region.

The robust method T-funHDDC allowed for distinct patterns of visibility and temperature to be extracted from the data which otherwise would have low differences in comparison to speeding. These groups have low speeding variation which brings into question how drivers react to changes in visibility.

5 Conclusions

In this paper we propose a model-based clustering method, called T-funHDDC, for functional data with outliers. The method is based on a multivariate functional principal component analysis and a functional latent multivariate t distribution mixture model. This is an extension of the funHDDC (Schmutz et al, 2020) algorithm for functional data with outliers, following the ideas used for the teigen method in (Andrews and McNicholas, 2012). Parameter estimation is carried out within the EM algorithm framework. Numerical optimization is only used for fitting the degrees of freedom for the multivariate t distributions, all the other parameters are available in closed form. Similar with funHDDC, we consider a family of parsimonious models, including models with constrained degrees of freedom for the t distributions.

We also extend the C-funHDDC method (Amovin-Assagba et al, 2022) by adding five more parsimonious models. Similarly with the CNmixt algorithm (Punzo and McNicholas, 2016), C-funHDDC is based on a contaminated Gaussian mixture model.

We compare the performance of funHDDC, C-funHDDC, and T-funHDDC for simulated univariate and bivariate curves with outliers. T-funHDDC always outperforms funHDDC, for uni-variate curves it has a similar performance

with C-funHDDC, and for bivariate curves it outperforms C-funHDDC. An advantage of the C-funHDDC method is that it can also be used for outlier detection.

For the NOx data, the C-funHDDC and T-funHDDC methods outperform funHDDC, and they have a similar performance with Funclust and robust functional clustering methods based on trimming, such as RFC and TrimK (Rivera-García et al, 2019). They have better performance than two-step methods based on CNmixt or teigen. Although there are several model-based methods for multivariate data with outliers that can be used to construct two-step methods for functional data, as observed in Bouveyron and Jacques (2011), these two-step methods always suffer from the difficulty to choose the best discretization.

For the real data application, the T-funHDDC method classifies the speeding data into subsets that are interesting for a follow up analysis of traffic patterns. Using a robust method, we were able to identify groups that show how drivers adapt their speed to reduced visibility. Speeding differential is a leading cause in traffic accidents, so finding conditions that increase speed variance can help diagnose when weather conditions pose the greatest risk to drivers and where the risk is greatest.

Both the contaminated Gaussian and the t distributions are well suited to work with mild outliers, and, due to polynomial tails, a mixture of t distributions can deal even with very large or small values. To handle highly atypical observations, as future work these methods can be extended as in Farcomeni and Punzo (2020) such that a proportion of the observations is trimmed.

6 Appendix

Proof of Proposition 1 The complete-data likelihood can be written as the product of the conditional densities of $\boldsymbol{c}_i$ given that $\boldsymbol{Z}_i = \boldsymbol{z}_i$ and $H_i = h_i$, the conditional densities of H_i given that $\boldsymbol{Z}_i = \boldsymbol{z}_i$, and the marginal densities of the $\boldsymbol{Z}_i$:

$$L_c(\boldsymbol{\theta}) = \prod_{i=1}^{n}\prod_{k=1}^{K}\left\{\phi(\boldsymbol{c}_i;\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k/h_i)g(h_i;\ \nu_k/2, \nu_k/2)\pi_k\right\}^{z_{ik}},$$

where $z_{ik} = 1$ if $\boldsymbol{c}_i$ belongs to the cluster k and $z_{ik} = 0$ otherwise. Thus, the complete-data log-likelihood can be written as

$$l_c(\boldsymbol{\theta}) = l_{1c}(\pi) + l_{2c}(\nu) + l_{3c}(\boldsymbol{\vartheta})$$

where

$$l_{1c}(\pi) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\log(\pi_k)$$

$$l_{2c}(\nu) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\log\left(g(h_i;\ \nu_k/2, \nu_k/2)\right)$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\left(-\log\left(\Gamma\left(\frac{\nu_k}{2}\right)\right) + \frac{\nu_k}{2}\log\left(\frac{\nu_k}{2}\right) + \frac{\nu_k}{2}\left(\log(h_i) - h_i\right) - \log(h_i)\right)$$

$$l_{3c}(\boldsymbol{\vartheta}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\Big(R\log(2\pi) + \log \mid \boldsymbol{\Sigma}_k \mid + h_i(\boldsymbol{c}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{c}_i - \boldsymbol{\mu}_k)\Big). \tag{34}$$

From (6) we have

$$\boldsymbol{\Sigma}_k^{-1} = \boldsymbol{W}^{1/2}\boldsymbol{Q}_k\boldsymbol{D}_k^{-1}\boldsymbol{Q}_k^\top\boldsymbol{W}^{1/2},$$

and

$$\mid \boldsymbol{\Sigma}_k \mid = \mid \boldsymbol{D}_k \mid\mid \boldsymbol{W} \mid^{-1} \mid\mid \boldsymbol{Q}_k^\top\boldsymbol{Q}_k \mid = \mid \boldsymbol{D}_k \mid\mid \boldsymbol{W} \mid^{-1} = \mid \boldsymbol{W} \mid^{-1} \prod_{l=1}^{d_k} a_{kl} \prod_{l=d_k+1}^{R} b_k. \tag{35}$$

Moreover, since $(\boldsymbol{c}_i - \boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{c}_i - \boldsymbol{\mu}_k)$ is a scalar, we get

$$\begin{aligned}
&(\boldsymbol{c}_i - \boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{c}_i - \boldsymbol{\mu}_k) = \operatorname{trace}((\boldsymbol{c}_i - \boldsymbol{\mu}_k)^\top\boldsymbol{W}^{1/2}\boldsymbol{Q}_k\boldsymbol{D}_k^{-1}\boldsymbol{Q}_k^\top\boldsymbol{W}^{1/2}(\boldsymbol{c}_i - \boldsymbol{\mu}_k)) \\
&= \operatorname{trace}\Big(\Big((\boldsymbol{c}_i - \boldsymbol{\mu}_k)^\top\boldsymbol{W}^{1/2}\boldsymbol{Q}_k\Big)\Big(\boldsymbol{D}_k^{-1}\boldsymbol{Q}_k^\top\boldsymbol{W}^{1/2}(\boldsymbol{c}_i - \boldsymbol{\mu}_k)\Big)\Big) \\
&= \operatorname{trace}\Big(\Big(\boldsymbol{D}_k^{-1}\boldsymbol{Q}_k^\top\boldsymbol{W}^{1/2}(\boldsymbol{c}_i - \boldsymbol{\mu}_k)\Big)\Big((\boldsymbol{c}_i - \boldsymbol{\mu}_k)^\top\boldsymbol{W}^{1/2}\boldsymbol{Q}_k\Big)\Big)
\end{aligned} \tag{36}$$

Replacing in (34) we obtain

$$\begin{aligned}
l_{3c}(\boldsymbol{\vartheta}) = &-\frac{nR\log(2\pi)}{2} + \frac{n}{2}\log(\mid \boldsymbol{W} \mid) - \frac{1}{2}\sum_{k=1}^{K} n_k \sum_{l=1}^{d_k}\log(a_{kl}) - \frac{1}{2}\sum_{k=1}^{K} n_k \sum_{l=d_k+1}^{R}\log(b_k) \\
&- \frac{1}{2}\sum_{k=1}^{K}\operatorname{trace}\Big(\Big(\boldsymbol{D}_k^{-1}\boldsymbol{Q}_k^\top\boldsymbol{W}^{1/2}\Big)\Big(\sum_{i=1}^{n} z_{ik}h_i(\boldsymbol{c}_i - \boldsymbol{\mu}_k)(\boldsymbol{c}_i - \boldsymbol{\mu}_k)^\top\Big)\Big(\boldsymbol{W}^{1/2}\boldsymbol{Q}_k\Big)\Big).
\end{aligned}$$

We can rewrite $l_{3c}(\boldsymbol{\vartheta})$ as

$$\begin{aligned}
l_{3c}(\boldsymbol{\vartheta}) &= -\frac{nR\log(2\pi)}{2} + \frac{n}{2}\log(\mid \boldsymbol{W} \mid) - \frac{1}{2}\sum_{k=1}^{K} n_k \sum_{l=1}^{d_k}\log(a_{kl}) \\
&- \frac{1}{2}\sum_{k=1}^{K} n_k \sum_{l=d_k+1}^{R}\log(b_k) - \frac{1}{2}\sum_{k=1}^{K}\operatorname{trace}\Big(\boldsymbol{D}_k^{-1}\boldsymbol{Q}_k^\top\boldsymbol{W}^{1/2}\boldsymbol{S}_k\boldsymbol{W}^{1/2}\boldsymbol{Q}_k\Big) \\
&= -\frac{nR\log(2\pi)}{2} + \frac{n}{2}\log(\mid \boldsymbol{W} \mid) - \frac{1}{2}\sum_{k=1}^{K} n_k \sum_{l=1}^{d_k}\log(a_{kl}) - \frac{1}{2}\sum_{k=1}^{K} n_k \sum_{l=d_k+1}^{R}\log(b_k) \\
&- \frac{1}{2}\sum_{k=1}^{K}\Big(\sum_{l=1}^{d_k}\frac{\boldsymbol{q}_{kl}^\top\boldsymbol{W}^{1/2}\boldsymbol{S}_k\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{a_{kl}} + \sum_{l=d_k+1}^{R}\frac{\boldsymbol{q}_{kl}^\top\boldsymbol{W}^{1/2}\boldsymbol{S}_k\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{b_k}\Big),
\end{aligned}$$

where $\boldsymbol{q}_{kl}$ is the lth column of $\boldsymbol{Q}_k$, $\boldsymbol{S}_k$ is defined in (17), and $\boldsymbol{\vartheta} = \{\boldsymbol{\mu}_k, a_{kj}, b_k, \boldsymbol{q}_{kj}\}$, $k = 1, \ldots, k$, $j = 1, \ldots, d_k$. □

Proof of Proposition 2 From (36) we obtain

$$(\boldsymbol{c}_i - \boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{c}_i - \boldsymbol{\mu}_k) = \Big(\sum_{l=1}^{d_k}\frac{\boldsymbol{q}_{kl}^\top\boldsymbol{W}^{1/2}(\boldsymbol{c}_i - \boldsymbol{\mu}_k)(\boldsymbol{c}_i - \boldsymbol{\mu}_k)^\top\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{a_{kl}}$$

$$+\sum_{l=d_k+1}^{R}\frac{\boldsymbol{q}_{kl}^{\top}\boldsymbol{W}^{1/2}(\boldsymbol{c}_i-\boldsymbol{\mu}_k)(\boldsymbol{c}_i-\boldsymbol{\mu}_k)^{\top}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{b_k}\Bigg)$$
$$=\delta(\boldsymbol{c}_i;\ \boldsymbol{\mu}_k,\boldsymbol{Q}_k,a,b,d_k). \tag{37}$$

Replacing in (7) and using also (35) we obtain (18).

From Section 7.5 in McLachlan and Peel (2004) we know that the distribution of H_i given $Z_{ik}=1$ and $\boldsymbol{c}_1,\ldots,\boldsymbol{c}_n$ is Gamma (m_{1k},m_{2k}) where

$$m_{1k}:=\frac{\nu_k^{(m-1)}+R}{2},\quad m_{2k}:=\frac{\nu_k^{(m-1)}+\delta(\boldsymbol{c}_i;\ \boldsymbol{\mu}_k^{(m-1)},\boldsymbol{Q}_k^{(m-1)},a^{(m-1)},b^{(m-1)},d_k)}{2}.$$

This implies (21) and

$$E[\log(H_i)\mid Z_{ik}=1,\boldsymbol{c}_1,\ldots,\boldsymbol{c}_n,\boldsymbol{\theta}^{(m-1)}]=\Psi(m_{1k})-\log(m_{2k}),$$

where $\Psi(\cdot)$ is defined in (23) (McLachlan and Peel, 2004, Section 7.5). Replacing the formulas for m_{1k}, m_{2k} and using (21) we get (22). □

Proof of Proposition 3 Using (13)-(16) we have that $Q(\boldsymbol{\theta}\mid\boldsymbol{\theta}^{(m-1)})$ is given by

$$Q(\boldsymbol{\theta}\mid\boldsymbol{\theta}^{(m-1)})=Q_1(\pi\mid\boldsymbol{\theta}^{(m-1)})+Q_2(\nu\mid\boldsymbol{\theta}^{(m-1)})+Q_3(\boldsymbol{\vartheta}\mid\boldsymbol{\theta}^{(m-1)}),$$

were

$$Q_1(\pi\mid\boldsymbol{\theta}^{(m-1)})=\sum_{i=1}^{n}\sum_{k=1}^{K}t_{ik}^{(m)}\log(\pi_k)$$

$$Q_2(\nu\mid\boldsymbol{\theta}^{(m-1)})=\sum_{i=1}^{n}\sum_{k=1}^{K}t_{ik}^{(m)}\Bigg(-\log\left(\Gamma\left(\frac{\nu_k}{2}\right)\right)+\frac{\nu_k}{2}\log(\frac{\nu_k}{2})$$
$$+\frac{\nu_k}{2}\left(\log(h_{ik})^{(m)}-h_{ik}^{(m)}\right)-\log(h_{ik})^{(m)}\Bigg)$$

$$Q_3(\boldsymbol{\vartheta}\mid\boldsymbol{\theta}^{(m-1)})=-\frac{nR\log(2\pi)}{2}+\frac{n}{2}\log(\mid\boldsymbol{W}\mid)$$
$$-\frac{1}{2}\sum_{k=1}^{K}n_k^{(m)}\sum_{l=1}^{d_k}\log(a_{kl})-\frac{1}{2}\sum_{k=1}^{K}n_k^{(m)}\sum_{l=d_k+1}^{R}\log(b_k)$$
$$-\frac{1}{2}\sum_{k=1}^{K}n_k^{(m)}\Bigg(\sum_{l=1}^{d_k}\frac{\boldsymbol{q}_{kl}^{\top}\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{a_{kl}}+\sum_{l=d_k+1}^{R}\frac{\boldsymbol{q}_{kl}^{\top}\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{b_k}\Bigg),$$

where $\boldsymbol{S}_k^{(m)}$ is defined in (28).

For the estimation of π_k, $k=1,\ldots,K$ we introduce the Lagrange multiplier λ and we maximize $Q_1=Q_1(\pi\mid\boldsymbol{\theta}^{(m-1)})-\lambda(\sum_{k=1}^{K}\pi_k-1)$. We get (24) solving the system

$$\frac{\partial Q_1}{\partial\pi_k}=\sum_{i=1}^{n}\frac{t_{ik}^{(m)}}{\pi_k}-\lambda=0,k=1,\ldots,K\quad\frac{\partial Q_1}{\partial\lambda}=\sum_{k=1}^{K}\pi_k-1=0.$$

Replacing $\log(h_{ik})^{(m)}$ from (22) we get

$$Q_2(\nu\mid\boldsymbol{\theta}^{(m-1)})=\sum_{i=1}^{n}\sum_{k=1}^{K}t_{ik}^{(m)}\Bigg(-\log\left(\Gamma\left(\frac{\nu_k}{2}\right)\right)+\frac{\nu_k}{2}\log(\frac{\nu_k}{2})$$

$$+\frac{\nu_k}{2}\left(\log(h_{ik}^{(m)})+\Psi\left(\frac{\nu_k^{(m-1)}+R}{2}\right)-\log\left(\frac{\nu_k^{(m-1)}+R}{2}\right)-h_{ik}^{(m)}\right)$$
$$-\log(h_{ik}^{(m)})-\Psi\left(\frac{\nu_k^{(m-1)}+R}{2}\right)+\log\left(\frac{\nu_k^{(m-1)}+R}{2}\right)\Bigg) \tag{38}$$

From the equation

$$\frac{\partial Q_2(\nu \mid \boldsymbol{\theta}^{(m-1)})}{\partial \nu_k}=0,$$

we get that $\nu_k^{(m)}$ is a solution of the equation (26).

If we consider the degrees of freedom to be the same for all groups, (38) becomes

$$Q_2(\nu \mid \boldsymbol{\theta}^{(m-1)})=n\Bigg(-\log\left(\Gamma\left(\frac{\nu}{2}\right)\right)+\frac{\nu}{2}\log(\frac{\nu}{2})+\frac{\nu}{2}\Bigg(\Psi\left(\frac{\nu^{(m-1)}+R}{2}\right)$$
$$-\log\left(\frac{\nu^{(m-1)}+R}{2}\right)\Bigg)-\Psi\left(\frac{\nu^{(m-1)}+R}{2}\right)+\log\left(\frac{\nu^{(m-1)}+R}{2}\right)\Bigg)$$
$$+\frac{\nu}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}t_{ik}^{(m)}\left(\log(h_{ik}^{(m)})-h_{ik}^{(m)}\right)-\sum_{k=1}^{K}t_{ik}^{(m)}\log(h_{ik}^{(m)}),$$

and an update for $\nu^{(m)}$ can be found by solving numerically the equation (27).

To get an update for $\boldsymbol{\mu}_k^{(m)}$ we calculate $Q_3(\boldsymbol{\vartheta} \mid \boldsymbol{\theta}^{(m-1)})$ starting from the formula (34) and we obtain

$$Q_3(\boldsymbol{\vartheta} \mid \boldsymbol{\theta}^{(m-1)})=-\frac{n}{2}R\log(2\pi)-\frac{1}{2}\sum_{k=1}^{K}n_k^{(m)}\log\mid\boldsymbol{\Sigma}_k\mid$$
$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}t_{ik}^{(m)}h_{ik}^{(m)}(\boldsymbol{c}_i-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{c}_i-\boldsymbol{\mu}_k).$$

The gradient of Q_3 with respect to $\boldsymbol{\mu}_k$ is

$$\nabla_{\boldsymbol{\mu}_k}Q_3(\boldsymbol{\vartheta} \mid \boldsymbol{\theta}^{(m-1)})=-\sum_{i=1}^{n}t_{ik}^{(m)}h_{ik}^{(m)}(\boldsymbol{c}_i-\boldsymbol{\mu}_k)\boldsymbol{\Sigma}_k^{-1}$$
$$=\left(-\sum_{i=1}^{n}t_{ik}^{(m)}h_{ik}^{(m)}\boldsymbol{c}_i+\boldsymbol{\mu}_k\sum_{i=1}^{n}t_{ik}^{(m)}h_{ik}^{(m)}\right)\boldsymbol{\Sigma}_k^{-1}.$$

Thus, we can easily get (25) solving $\nabla_{\boldsymbol{\mu}_k}Q_3(\boldsymbol{\vartheta} \mid \boldsymbol{\theta}^{(m-1)})=\mathbf{0}$.

To estimate $\boldsymbol{Q}_k$ we have to maximize $Q_3(\boldsymbol{\vartheta} \mid \boldsymbol{\theta}^{(m-1)})$ with respect to $\boldsymbol{q}_{kl}$ under the constraint $\boldsymbol{q}_{kl}^\top\boldsymbol{q}_{kl}=1$. This is equivalent with minimizing $-2Q_3(\boldsymbol{\vartheta} \mid \boldsymbol{\theta}^{(m-1)})$ with respect to $\boldsymbol{q}_{kl}$ under this constraint, so we consider the function $Q_{3c}=-2Q_3(\boldsymbol{\vartheta} \mid \boldsymbol{\theta}^{(m-1)})-\sum_{l=1}^{R}\omega_{kl}(\boldsymbol{q}_{kl}^\top\boldsymbol{q}_{kl}-1)$, where ω_{kl} are Lagrange multipliers. The gradient of Q_{3c} with respect to $\boldsymbol{q}_{kl}$ is

$$\nabla_{\boldsymbol{q}_{kl}}Q_{3c}=2n_k^{(m)}\frac{\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{\Sigma_{kl}}-2\omega_{kl}\boldsymbol{q}_{kl},$$
$$\Sigma_{kl}=\begin{cases}a_{kl}\text{ if }l=1,\ldots,d_k\\ b_k\text{ if }l=d_k+1,\ldots,R.\end{cases}$$

From $\nabla_{\boldsymbol{q}_{kl}} Q_{3c} = 0$ we get $\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl} = \frac{\omega_{kl}\Sigma_{kl}}{n_k^{(m)}}\boldsymbol{q}_{kl}$, so $\boldsymbol{q}_{kl}$ is an eigenfunction of $\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}$ and the associated eigenvalue is $\lambda_{kl}^{(m)} = \frac{\omega_{kl}\Sigma_{kl}}{n_k^{(m)}}$. Notice that we also have $\boldsymbol{q}_{kl}^\top\boldsymbol{q}_{kj} = 0$ if $l \neq j$, and $\lambda_{kl}^{(m)} = \boldsymbol{q}_{kl}^\top\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}$ so we can write

$$
\begin{aligned}
&-2Q_3(\boldsymbol{\vartheta} \mid \boldsymbol{\theta}^{(m-1)}) = nR\log(2\pi) - n\log(|\,\boldsymbol{W}\,|) + \sum_{k=1}^{K} n_k^{(m)}\left(\sum_{l=1}^{d_k}\log(a_{kl})\right.\\
&\left.+\sum_{l=d_k+1}^{R}\log(b_k)\right) + \sum_{k=1}^{K} n_k^{(m)}\left(\sum_{l=1}^{d_k}\frac{\lambda_{kl}^{(m)}}{a_{kl}} + \sum_{l=d_k+1}^{R}\frac{\lambda_{kl}^{(m)}}{b_k}\right)\\
&= nR\log(2\pi) - n\log(|\,\boldsymbol{W}\,|) + \sum_{k=1}^{K} n_k^{(m)}\left(\sum_{l=1}^{d_k}\log(a_{kl}) + \sum_{l=d_k+1}^{R}\log(b_k)\right)\\
&+\sum_{k=1}^{K} n_k^{(m)}\left(\sum_{l=1}^{d_k}\lambda_{kl}^{(m)}\left(\frac{1}{a_{kl}} - \frac{1}{b_k}\right) + \frac{1}{b_k}\mathrm{trace}(\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2})\right).
\end{aligned}
$$

Here we have also used

$$
\mathrm{trace}(\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}) = \sum_{l=1}^{R}\lambda_{kl}^{(m)} = \sum_{l=1}^{d_k}\lambda_{kl}^{(m)} + \sum_{l=d_k+1}^{R}\lambda_{kl}^{(m)}. \tag{39}
$$

Since for any $l = 1,\ldots,d_k$ we have $a_{kl} \geq b_k$, we get $\frac{1}{a_{kl}} - \frac{1}{b_k} \leq 0$, so $\sum_{l=1}^{d_k}\lambda_{kl}^{(m)}\left(\frac{1}{a_{kl}} - \frac{1}{b_k}\right)$ is a decreasing function of λ_{kl}. Thus, we estimate $\boldsymbol{q}_{kl}$ by the eigenfunction associated with the lth highest eigenvalue of $\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}$.

To update a_{kl} we solve

$$
\frac{\partial Q_3(\boldsymbol{\vartheta} \mid \boldsymbol{\theta}^{(m-1)})}{\partial a_{kl}} = -\frac{n_k^{(m)}}{2a_{kl}} + \frac{n_k^{(m)}\boldsymbol{q}_{kl}^\top\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{2a_{kl}^2} = 0,
$$

and we get $a_{kl}^{(m)} = \boldsymbol{q}_{kl}^\top\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl} = \lambda_{kl}^{(m)}$, the lth highest eigenvalue of $\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}$.

From

$$
\frac{\partial Q_3(\boldsymbol{\vartheta} \mid \boldsymbol{\theta}^{(m-1)})}{\partial b_k} = -\frac{n_k^{(m)}}{2}\sum_{l=d_k+1}^{R}\frac{1}{b_k} + \frac{n_k^{(m)}}{2}\sum_{l=d_k+1}^{R}\frac{\boldsymbol{q}_{kl}^\top\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl}}{b_k^2} = 0,
$$

we obtain

$$
b_k^{(m)} = \frac{1}{R-d_k}\sum_{l=d_k+1}^{R}\boldsymbol{q}_{kl}^\top\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}\boldsymbol{q}_{kl} = \frac{1}{R-d_k}\sum_{l=d_k+1}^{R}\lambda_{kl}^{(m)}
$$

Thus, using (39) we get

$$
b_k^{(m)} = \frac{1}{R-d_k}\left(\mathrm{trace}(\boldsymbol{W}^{1/2}\boldsymbol{S}_k^{(m)}\boldsymbol{W}^{1/2}) - \sum_{l=1}^{d_k}\lambda_{kl}^{(m)}\right).
$$

□

References

Aitken AC (1927) On Bernoulli's numerical solution of algebraic equations. Proceedings of the Royal Society of Edinburgh 46:289–305. https://doi.org/10.1017/S0370164600022070

Amovin-Assagba M, Gannaz I, Jacques J (2022) Outlier detection in multivariate functional data through a contaminated mixture model. Comput Stat Data Anal 174

Andrews JL, McNicholas PD (2011) Extending mixtures of multivariate *t*-distributions. Stat Comput 21:361–373. https://doi.org/10.1007/s11222-010-9175-2

Andrews JL, McNicholas PD (2012) Model-based clustering, classification, and discriminant analysis via mixtures of multivariate *t*-distributions: The teigen family. Stat Comput 22:1021–1029. https://doi.org/10.1007/s11222-011-9272-x

Andrews JL, McNicholas PD, Subedi S (2011) Model-based classification via mixtures of multivariate *t*-distributions. Comput Stat Data Anal 55(1):520–529. https://doi.org/https://doi.org/10.1016/j.csda.2010.05.019

Andrews JL, Wickins JR, Boers NM, et al (2018) An R package for model-based clustering and classification via the multivariate *t* distribution. Journal of Statistical Software 83(7):1–32

Anton C, Smith I (2023) Model based clustering of functional data with mild outliers. In: Brito P, Dias J, Lausen B, et al (eds) Classification and Data Science in the Digital Age. Studies in Classification, Data Analysis, and Knowledge Organization, Springer International Publishing, to appear

Bagnato L, Punzo A, Zoia MG (2017) The multivariate leptokurtic-normal distribution and its application in model-based clustering. Canadian Journal of Statistics 45(1):95–119

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

Bouveyron C, Girard S, Schmid C (2007) High-dimensional data clustering. Comput Stat Data Anal 52(1):502–519

Celeux G, Govaert G (1995) Gaussian parsimonious clustering models. Pattern Recognition 28(5):781–793. https://doi.org/https://doi.org/10.1016/0031-3203(94)00125-6

Cuesta-Albertos JA, Gordaliza A, Matrán C (1997) Trimmed k-means: an attempt to robustify quantizers. Ann Stat 25(2):553 – 576. https://doi.org/10.1214/aos/1031833664

Cuevas A, Febrero M, Fraiman R (2007) Robust estimation and classification for functional data via projection-based depth notions. Comput Stat 22(3):481–496. https://doi.org/10.1007/s00180-007-0053-0

Dang UJ, Browne RP, McNicholas PD (2015) Mixtures of multivariate power exponential distributions. Biometrics 71(4):1081–1089

Delaigle A, Hall P (2010) Defining probability density for a distribution of random functions. Ann Stat 38(2):1171–1193. URL http://www.jstor.org/stable/25662272

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Series B Stat Methodol 39(1):1–38

Farcomeni A, Punzo A (2020) Robust model-based clustering with mild and gross outliers. TEST: An Official Journal of the Spanish Society of Statistics and Operations Research 29(4):989–1007. https://doi.org/10.1007/s11749-019-00693-

Febrero-Bande M, de la Fuente MO (2012) Statistical computing in functional data analysis: The R package fda.usc. J Stat Softw 51(4):1–28. https://doi.org/10.18637/jss.v051.i04

Febrero-Bande M, Galeano P, Gonzãlez-Manteiga W (2008) Outlier detection in functional data by depth measures, with application to identify abnormal nox levels. Environmetrics 19:331 – 345. https://doi.org/10.1002/env.878

Fraiman R, Muniz G (2001) Trimmed means for functional data. TEST: An Official Journal of the Spanish Society of Statistics and Operations Research 10:419–440. https://doi.org/10.1007/BF02595706

García-Escudero L, Gordaliza A (2005) A proposal for robust curve clustering. J Classif 22:185–201. https://doi.org/10.1007/s00357-005-0013-8

Holzmann H, Munk A, Gneitting T (2006) Identifiability of finite mixtures of elliptical distributions. Scandinavian Journal of Statistics 33(4):753–763. https://doi.org/https://doi.org/10.1111/j.1467-9469.2006.00505.x

Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218

Jacques J, Preda C (2013) Funclust: A curves clustering method using functional random variables density approximation. Neurocomputing 112:164–171. https://doi.org/https://doi.org/10.1016/j.neucom.2012.11.042

Jacques J, Preda C (2014a) Functional data clustering: A survey. Adv Data Anal Classif 8(3):231–255. https://doi.org/10.1007/s11634-013-0158-y

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

McLachlan G, Peel D (2004) Finite Mixture Models. Wiley Series in Probability and Statistics, Wiley

McNicholas PD, Murphy TB, McDaid AF, et al (2010) Serial and parallel implementations of model-based clustering via parsimonious Gaussian mixture models. Comput Stat Data Anal 54(3):711–723

Meng XL, Rubin DB (1993) Maximum likelihood estimation via the ECM algorithm: A general framework. Biometrika 80(2):267–278. https://doi.org/10.1093/biomet/80.2.267

Peel D, McLachlan GJ (2000) Robust mixture modelling using the t distribution. Stat Comput 10(4):339–348

Punzo A, McNicholas PD (2016) Parsimonious mixtures of multivariate contaminated normal distributions. Biometrical Journal 58(6):1506–1537. https://doi.org/https://doi.org/10.1002/bimj.201500144

Punzo A, Mazza A, McNicholas PD (2018) Contaminatedmixt: An R package for fitting parsimonious mixtures of multivariate contaminated normal distributions. Journal of Statistical Software 85(10):1–25

Punzo A, Blostein M, McNicholas PD (2020) High-dimensional unsupervised classification via parsimonious contaminated mixtures. Pattern Recognition 98:107031. https://doi.org/10.1016/j.patcog.2019.107031

Ramsay J, Silverman B (2006) Functional Data Analysis. Springer Series in Statistics, Springer New York

Ritter G (2014) Robust Cluster Analysis and Variable Selection, Monographs on Statistics & Applied Probability, vol 37. Chapman and Hall/CRC

Rivera-García D, García-Escudero LA, Mayo-Iscar A, et al (2019) Robust clustering for functional data based on trimming and constraints. Adv Data Anal Classif 13(1):201–225. https://doi.org/10.1007/s11634-018-0312-7

Sawant P, Billor N, Shin H (2012) Functional outlier detection with robust functional principal component analysis. Comput Stat 27(1):83–102. https:

//doi.org/10.1007/s00180-011-0239-3

Schmutz A, Jacques J, Bouveyron C, et al (2020) Clustering multivariate functional data in group-specific functional subspaces. Comput Stat 35:1101–1131

Schwarz G (1978) Estimating the dimension of a model. Ann Stat pp 461–464

Sguera C, Galeano P, Lillo RE (2015) Functional outlier detection by a local depth with application to nox levels. Stoch Environ Res Risk Assess 30:1115–1130

Tomarchio SD, Bagnato L, A. P (2022) Model-based clustering via new parsimonious mixtures of heavy tailed distributions. AStA Advances in Statistical Analysis 106(2):315–347