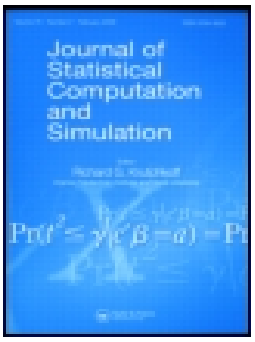


Taylor & Francis
Taylor & Francis Group

Journal of Statistical Computation and Simulation

ISSN: 0094-9655 (Print) 1563-5163 (Online) Journal homepage: http://www.tandfonline.com/loi/gscs20

Unsupervised learning of regression mixture models with unknown number of components

Faicel Chamroukhi

To cite this article: Faicel Chamroukhi (2015): Unsupervised learning of regression mixture models with unknown number of components, Journal of Statistical Computation and Simulation, DOI: 10.1080/00949655.2015.1109096

To link to this article: http://dx.doi.org/10.1080/00949655.2015.1109096

Published online: 05 Nov 2015.

JOURNAL OF STATISTICAL COMPUTATION AND SIMULATION, 2015
http://dx.doi.org/10.1080/00949655.2015.1109096

Taylor & Francis
Taylor & Francis Group

Unsupervised learning of regression mixture models with unknown number of components

Faicel Chamroukhi[a,b]

[a]Aix Marseille Université, CNRS, ENSAM, LSIS UMR 7296, Marseille, France; [b]Université de Toulon, CNRS, LSIS UMR 7296, La Garde, France

ABSTRACT

We propose a new unsupervised learning algorithm to fit regression mixture models with unknown number of components. The developed approach consists in a penalized maximum likelihood estimation carried out by a robust expectation–maximization (EM)-like algorithm. We derive it for polynomial, spline, and B-spline regression mixtures. The proposed learning approach is unsupervised: (i) it simultaneously infers the model parameters and the optimal number of the regression mixture components from the data as the learning proceeds, rather than in a two-fold scheme as in standard model-based clustering using afterward model selection criteria, and (ii) it does not require accurate initialization unlike the standard EM for regression mixtures. The developed approach is applied to curve clustering problems. Numerical experiments on simulated and real data show that the proposed algorithm performs well and provides accurate clustering results, and confirm its benefit for practical applications.

ARTICLE HISTORY
Received 20 March 2015
Accepted 13 October 2015

KEYWORDS
Unsupervised learning; regression mixtures; EM algorithm; robust EM-like algorithm; model selection; curve clustering

AMS SUBJECT CLASSIFICATION
62-XX; 62H30; 62H12; 62-07

1. Introduction

Mixture modelling [1] is one of the most popular and successful approaches in density estimation and cluster and discriminant analyses.[1–6] In this paper, we focus on the finite mixture model [1] and its use in clustering, that is, model-based clustering,[2,4,6] which is a widely used and a successful approach in cluster analysis. In the finite mixture approach for cluster analysis, the data probability density function is assumed to be a mixture density, each component density being associated with a cluster. The problem of clustering, therefore, becomes the one of estimating the parameters of the mixture model (e.g. estimating the mean vector and the covariance matrix for each component density in the case of Gaussian mixture models (GMMs)). Maximum likelihood estimation of the mixture density is often performed using the well-known expectation–maximization (EM) algorithm [7,8] thanks to its good desirable properties of stability and reliable convergence. One of the main model-based clustering approaches is the one based on the finite GMM and the EM algorithm or its extensions.[1,8] It concerns in general multivariate (vectorial) data. However, in many areas of application, such as electrical engineering,[9] railway monitoring,[10] speech or phoneme recognition,[11] radar waveform recognition,[12] etc, the individuals are curves or functions, which are more structured, so that a standard multivariate analysis that considers individuals as vectorial data, is not adapted. The adapted analysis approaches in this case relate the functional data analysis (FDA) framework [13,14] which concerns the paradigm of data analysis for which the individuals are curves or time series, or more generally functions, rather than vectors of reduced dimensions. In this case, the clustering can be

CONTACT Faicel Chamroukhi ✉ faicel.chamroukhi@univ-tln.fr

performed by using dedicated model-based curve clustering approaches, in particular the regression mixture model, including polynomial regression mixtures (PRM), spline regression mixtures (SRM), and B-spline regression mixtures (bSRM).[15–17] Non-parametric statistical approaches have also been proposed for functional data discrimination [11,18] and clustering.[11]

In this paper, we focus on regression mixtures and their use in model-based curve clustering. Modelling with regression mixtures is an important topic in the general family of mixture models. The regression mixture model [15,17,19–26] arises when we assume that the observed response $\mathbf{y}_i$ for the predictor variable $\mathbf{x}_i$ is generated from one of K possibly parametric regression functions $g(\mathbf{y}_i|\mathbf{x}_i;\boldsymbol{\theta}_k)$ of parameters $\boldsymbol{\theta}_k$ with prior probability π_k. This includes PRM, SRM, and bSRM. These three models are considered here. The use of regression mixtures for density estimation as well as for cluster and discriminant analyses, requires fitting the mixture parameters.

The problem of fitting regression mixture models is a widely studied problem in statistics, machine learning, and data analysis, particularly for cluster analyses. It is usually performed by maximum likelihood by using the EM algorithm.[7,8,17,21] However, it is well-known that the initialization is crucial for EM. If the initialization is not appropriately performed, the EM algorithm may lead to unsatisfactory results. The EM algorithm also requires the number of clusters to be given a priori. The problem of selecting the number of mixture components in this case can be addressed by using, in an afterward step, some model selection criteria to choose one from a set of pre-estimated candidate models. The problem of fitting regression mixtures is also related to the one of fitting Gaussian mixtures for multivariate data, for which some solutions have been provided particularly those in [27,28]. However, these approaches mainly concern GMMs for multivariate data, rather than functional data or curves.

In this paper, we consider regression mixtures and their use in model-based clustering for curves, rather than for vectorial data. We propose a new unsupervised learning algorithm to fit regression mixture models with unknown number of components. The developed unsupervised learning approach consists in a penalized maximum likelihood estimation carried out by a robust EM-like algorithm. We derive the proposed algorithm for fitting PRM, SRM, and bSRM. The proposed learning approach is unsupervised. It simultaneously infers the model parameters and the optimal number of the regression mixture components from the data as the learning proceeds, rather than in a two-fold scheme as in standard model-based clustering using afterward model selection criteria. Furthermore, it does not require accurate initialization unlike the standard EM for regression mixtures.

This paper is organized as follows. In Section 2, we give a background on regression mixtures and their use in model-based clustering. In Section 3, we present the proposed approach and the robust EM-like algorithm for fitting regression mixtures and model-based curve clustering. Then, in Section 4, we present experimental results on both simulated data and real-world data sets to apply and assess the proposed approach. Finally, in Section 5, we discuss the proposal and provide concluding remarks and future directions.

2. Regression mixtures

The finite regression mixture model provides a way to model data arising from a number of unknown classes of conditionally dependent observed data. Let us denote by $\mathcal{D} = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n))$ an observed independently and identically distributed (i.i.d) sample where each individual is a couple of a response $\mathbf{y}_i$ and its corresponding covariate $\mathbf{x}_i$. For example, in the case of curves, the response consists of m_i observations $\mathbf{y}_i = (y_{i1}, \ldots, y_{im_i})$ (regularly) observed at the inputs $\mathbf{x}_i = (x_{i1}, \ldots, x_{im_i})$ for all $i = 1, \ldots, n$ (e.g. x may represent the sampling time in a temporal context). The finite regression mixture model assumes that each individual $(\mathbf{x}_i, \mathbf{y}_i)$ is drawn from a mixture density of K (possibly unknown) components, whose mixing proportions are $(\pi_1, \ldots, \pi_K)$ where $\pi_k = p(z_i = k)$ is the prior probability of component k, $z_i \in \{1, \ldots, K\}$ being the hidden class label of the ith individual. A common way to model the conditional dependence in the observed data is to use regression functions. The regression mixture model assumes that each mixture component k is a conditional component density $f_k(\mathbf{y}_i|\mathbf{x}_i;\boldsymbol{\theta}_k)$ of a regression model with parameters $\boldsymbol{\theta}_k$. This includes PRM, SRM, and bSRM, see for example.[16,21,29] One can also use generative hidden process regression mixtures [30–32] which can be seen as a hierarchical dynamical regression mixture model which also performs curve segmentation, in addition to mixture density estimation and curve clustering.

The PRM assumes that each observation $(\mathbf{x}_i, \mathbf{y}_i)$ is drawn from one of K polynomial regression functions with coefficients $\boldsymbol{\beta}_k$ corrupted by a standard zero-mean Gaussian noise:

$$\mathbf{y}_i = \mathbf{X}_i\boldsymbol{\beta}_k + \boldsymbol{\epsilon}_{ik}, \qquad (1)$$

where $\mathbf{X}_i$ is the $m_i \times (p+1)$ regression (Vandermonde) matrix with rows $x_j = (1, x_{ij}, x_{ij}^2 \ldots, x_{ij}^p)$, p being the polynomial degree, $\boldsymbol{\beta}_k = (\beta_{k0}, \ldots, \beta_{kp})^{\mathrm{T}}$ is the $(p+1) \times 1$ vector of regression coefficients for class k, $\boldsymbol{\epsilon}_{ik} \sim \mathcal{N}(\mathbf{0}, \sigma_k^2\mathbf{I}_{m_i})$ is a multivariate standard zero-mean Gaussian noise with a covariance matrix $\sigma_k^2\mathbf{I}_{m_i}$, and $\mathbf{I}_{m_i}$ denotes the $m_i \times m_i$ identity matrix. Thus, the PRM is given by the following conditional mixture density:

$$f(\mathbf{y}_i|\mathbf{x}_i;\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k\, \mathcal{N}(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_k, \sigma_k^2\mathbf{I}_{m_i})\cdot \qquad (2)$$

The model parameters are given by the parameter vector $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_K, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$ where the π_k's are the non-negative mixing proportions that sum to 1 and $\boldsymbol{\theta}_k = (\boldsymbol{\beta}_k, \sigma_k^2)$ represents the regression parameters and the noise variance for cluster k. The unknown parameter vector $\boldsymbol{\theta}$ is generally estimated by maximizing the observed-data log-likelihood given an i.i.d data set $\mathcal{D} = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n))$:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k\, \mathcal{N}(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_k, \sigma_k^2\mathbf{I}_{m_i})\cdot \qquad (3)$$

This is usually performed iteratively via EM.[7,16,17]

2.1. Standard EM for fitting regression mixtures

The EM algorithm for regression mixtures starts with an initial parameter vector $\boldsymbol{\theta}^{(0)}$ and alternates between the two following steps until convergence:

2.1.1. E-step

This step computes the expected log-likelihood for the complete-data $(\mathcal{D}, \mathbf{z})$ where $\mathbf{z} = (z_1, \ldots, z_n)$ are the unknown cluster labels, given the observed data $\mathcal{D}$ and a current estimation $\boldsymbol{\theta}^{(q)}$ of the parameter vector $\boldsymbol{\theta}$, q being the current iteration number. It simply consists in computing the posterior probability that the ith observation is generated from cluster k:

$$\tau_{ik}^{(q)} = p(z_i = k|\mathbf{y}_i, \mathbf{x}_i; \boldsymbol{\theta}^{(q)}) = \frac{\pi_k^{(q)}\mathcal{N}\big(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_k^{(q)}, \sigma_k^{2(q)}\mathbf{I}_{m_i}\big)}{\sum_{h=1}^{K} \pi_h^{(q)}\mathcal{N}(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_h^{(q)}, \sigma_h^{2(q)}\mathbf{I}_{m_i})}\cdot \qquad (4)$$

2.1.2. M-step

This step updates the model parameters and provides the parameter vector $\boldsymbol{\theta}^{(q+1)}$ by maximizing the expected complete-data log-likelihood computed at the E-step, with respect to $\boldsymbol{\theta}$ and provides the following parameter updates [1,8,15,16]:

$$\pi_k^{(q+1)} = \frac{1}{n}\sum_{i=1}^{n} \tau_{ik}^{(q)} \quad (k = 1, \ldots, K), \qquad (5)$$

$$\boldsymbol{\beta}_k^{(q+1)} = \left[\sum_{i=1}^{n} \tau_{ik}^{(q)} \mathbf{X}^{\mathrm{T}}{}_i \mathbf{X}_i\right]^{-1} \sum_{i=1}^{n} \tau_{ik}^{(q)} \mathbf{X}_i{}^{\mathrm{T}} \mathbf{y}_i, \tag{6}$$

$$\sigma_k^{2(q+1)} = \frac{1}{\sum_{i=1}^{n} \tau_{ik}^{(q)} m_i} \sum_{i=1}^{n} \tau_{ik}^{(q)} \parallel \mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta}_k \parallel^2 . \tag{7}$$

The pseudo code 1 summarizes the standard EM algorithm for PRMs.

Algorithm 1 Pseudo code of the standard EM algorithm for regression mixtures.

Inputs: Data $\mathcal{D} = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n))$, number of clusters K, and polynomial degree p

```
fix a threshold $\upsilon > 0$ ; set $q \leftarrow 0$ (iteration)
**Initialization:** $\boldsymbol{\theta}^{(0)} = (\pi_1^{(0)}, \ldots, \pi_K^{(0)}, \boldsymbol{\theta}_1^{(0)}, \ldots, \boldsymbol{\theta}_K^{(0)})$:
Initialize the partition randomly or by running $K$-means and initialize the $\pi_k$'s
Fit a regression model with parameters $\boldsymbol{\theta}_k^{(0)} = (\boldsymbol{\beta}_k^{(0)}, \sigma_k^{2(0)})$ to each cluster
while increment in log-likelihood $> \epsilon$ do
  //E-step:
  for $k = 1, \ldots, K$ do
    Compute $\tau_{ik}^{(q)}$ for $i = 1, \ldots, n$ using Equation (4)
  end for
  //M-step:
  for $k = 1, \ldots, K$ do
    Compute $\pi_k^{(q+1)}$ using Equation (5)
    Compute $\boldsymbol{\beta}_k^{(q+1)}$ using Equation (6)
    Compute $\sigma_k^{2(q+1)}$ using Equation (7)
  end for
  $q \leftarrow q + 1$
end while
```

Outputs: $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(q)}, \quad \hat{\tau}_{ik} = \tau_{ik}^{(q)}$

2.2. SRM and bSRM

The SRM (respectively, bSRM) is an extension of the previously described PRM to a semiparametric modelling by relying on splines (respectively, B-splines).

2.2.1. Spline and B-spline regression

Splines [33] are widely used for function approximation based on constrained piecewise polynomials. Let $\boldsymbol{\xi} = \xi_0 < \xi_1, \ldots, < \xi_L < \xi_{L+1}$ be ordered knots, including L internal knots, ξ_0 and ξ_{L+1} being the two boundary knots. An order-M spline with knots $\boldsymbol{\xi}$ is a piecewise-polynomial of degree $p = M - 1$ with continuous derivatives at the interior knots up to order $M - 2$. For example, an order-2 spline is a continuous piecewise linear function. The spline regression function can be written as:

$$y_{ij} = \sum_{\ell=0}^{p} \beta_\ell x_{ij}^{\ell} + \sum_{\ell=1}^{L} \beta_{\ell+p} (x_{ij} - \xi_\ell)_+^p + \epsilon_{ij}, \tag{8}$$

where $(x_{ij} - \xi_\ell)_+ = x_{ij} - \xi_\ell$ if $x_{ij} \geq \xi_k$ and $(x_{ij} - \xi_\ell)_+ = 0$ otherwise, $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_{L+p})^{\mathrm{T}}$ in $\mathbb{R}^{L+M}$ is the vector of spline coefficients, and ϵ is an additive Gaussian noise. This spline regression model can be written in a vectorial form as

$$\mathbf{y}_i = \mathbf{S}_i\boldsymbol{\beta} + \boldsymbol{\epsilon}_i, \tag{9}$$

where $\mathbf{S}_i$ is the $m_i \times (L + M)$ spline regression matrix with rows $\mathbf{s}_j = (1, x_{ij}, x_{ij}^2 \ldots, x_{ij}^p, (x_{ij} - \xi_1)_+^p, \ldots, (x_{ij} - \xi_L)_+^p)$.

For splines, the columns of the regression matrix $\mathbf{X}$ tend to be highly correlated since each column is a transformed version of x. This collinearity may result in a nearly singular matrix and imprecision in the spline fit.[34] B-splines allows efficient computations thanks to the block matrix form of the regression matrix. An order-M B-spline function is defined as a sum of linear combination of specific basis functions $B_{\ell,M}$ as:

$$y_{ij} = \sum_{\ell=1}^{L+M} \beta_\ell B_{\ell,M}(x_{ij}),\ x_{ij} \in [\zeta_\ell, \zeta_{\ell+M}], \tag{10}$$

where each Mth order B-spline $B_{\ell,M}$ is a piecewise polynomial of degree $p = M - 1$ that has finite support over $[\zeta_l, \zeta_{\ell+M}]$ and is zero everywhere else. Each of the basis functions $B_{\ell,M}(x_{ij})$ can be computed as in [35] (see also Appendix 1). The vectorial form for the B-spline regression model can be written as:

$$\mathbf{y}_i = \mathbf{B}_i\boldsymbol{\beta} + \boldsymbol{\epsilon}_i, \tag{11}$$

where each row of the $m_i \times (L + M)$ B-spline regression matrix $\mathbf{B}_i$ for the ith curve is given by: $\mathbf{b}_j = (B_{1,M}(x_{ij}),\ B_{2,M}(x_{ij}), \ldots,\ B_{L+M,M}(x_{ij}))$.

2.2.2. SRM and bSRM and the EM algorithm

The SRM (respectively, B-spline) is similarly defined as the PRM described previously. The mixture density in this case is given by Equation (2) where the regression matrix $\mathbf{X}_i$ is replaced by the spline regression matrix $\mathbf{S}_i$ (respectively, the B-spline regression matrix $\mathbf{B}_i$).

The parameter estimation procedure for the SRM and bSRM is the same as the one used for the PRM, that is maximum likelihood estimation via the EM algorithm. The E- and M- steps are still the same, as well as as the initialization procedure and the convergence conditions.

2.3. Regression mixtures for model-based clustering

From a clustering prospective, the estimated mixture components can be interpreted as K clusters, where each cluster is associated to a mixture component. Thus, once the model parameters have been estimated, the posterior cluster probabilities given by Equation (4) can be used as a fuzzy partition of the data into K clusters. They indeed represent the a posteriori uncertainty, given the observed data and the estimated model, about which cluster k each observed data $(\mathbf{x}_i, \mathbf{y}_i)$ is originated from. Furthermore, a hard partition of the data can then be obtained by assigning each observation to the cluster with the highest posterior probability (MAP rule), that is, by estimating the cluster label as:

$$\hat{z}_i = \arg\max_{k=1}^{K} \hat{\tau}_{ik}. \tag{12}$$

However, it can be noticed that, the standard EM algorithm for all theses regression mixture models is sensitive to initialization. It might yield poor estimations if the mixture parameters are not initialized properly. The EM initialization in general can be performed from a randomly chosen partition of the data or by computing a partition from another clustering algorithm such as K-means, Classification EM,[36] Stochastic EM,[37] etc or by initializing EM with a few number of iterations of EM itself. Several works have been proposed in the literature in order to overcome this drawback and making the EM algorithm for Gaussian mixtures robust with regard initialization.[28,38,39] Further details about choosing starting values for the EM algorithm for Gaussian mixtures can be found for example in [38]. In addition, the EM algorithm requires the number of mixture components (clusters) to be known. While the number of clusters can be chosen by some model selection criteria such as the Bayesian information criterion (BIC),[40] the Akaike information criterion (AIC) [41] or the integrated classification likelihood criterion,[42] or resampling methods such as bootstrap,[43,44] this requires performing an afterward model selection procedure. Some authors have considered this issue in order to estimate the unknown number of mixture components in GMMs, for example in [28,45].

In general, these two issues have been considered each separately. Among the approaches that consider the problem of robustness with regard to initial values and the one of estimating the number of mixture components, in the same algorithm, one can cite the EM algorithm proposed in [27]. This EM algorithm is capable of selecting the number of components and attempts to be not sensitive with regard to initial values. It optimizes a minimum message length criterion, which is a penalized log-likelihood, rather than the observed-data log-likelihood. The penalization term allows to control the model complexity. It starts by fitting a mixture model with a large number of clusters and discards invalid clusters as the learning proceeds. The degree of validity of each cluster is measured through the penalization term which includes the mixing proportions to know if the cluster is small or not to be discarded, and therefore to reduce the number of clusters. More recently, in [28], the authors developed a robust EM-like algorithm for model-based clustering of multivariate data using GMMs that simultaneously addresses the problem of initialization and the one of estimation of the number of mixture components. This algorithm overcomes some initialization drawback of the EM algorithm proposed in [27]. As shown in [28], this problem regarding initialization can become more serious especially for a data set with a large number of clusters. However, these presented model-based clustering approaches, including those in [27,28], are concerned with vectorial data where the observations are assumed to be vectors of reduced dimension. When the data are rather curves or functions, they are not adapted. Indeed, when the data are functional where the individuals are presented as curves or surfaces rather than vectors, they are in general very structured. Relying on standard multivariate mixture analysis may therefore lead to unsatisfactory results in terms of modelling and classification accuracy.[10,30,31,46] However, addressing the problem from an FDA prospective, that is formulating 'functional' mixture models, allows to overcome these limitations, for example, as in [10,16,30,46]. In this case of model-based functional data clustering, one can rely on the regression mixture approaches [15–17] or generative hidden process regression [15,30,31] which are adapted for curves with regime changes. In this paper, we attempt to overcome the limitations of the EM algorithm in the case of regression mixtures and model-based curve clustering by proposing an EM-like algorithm which is robust with regard initialization and automatically estimates the optimal number of clusters as the learning proceeds.

In the next section we derive our robust EM-like algorithm for fitting regression mixtures including PRM, SRM, and bSRM.

3. Penalized maximum likelihood via a robust EM-like algorithm for fitting regression mixtures

In this section we present the proposed EM-like algorithm for model-based curve clustering using regression mixtures. The present work is in the same spirit of the EM-like algorithm presented in [28], but by extending the idea to the case of functional data (curve) clustering, rather than multivariate data clustering. It is therefore concerned with regression mixture models rather than multivariate GMMs. Indeed, the data here are assumed to be curves rather than vectors of reduced dimensions. This leads to fitting regression mixture models (including splines or B-splines), rather than fitting standard Gaussian mixtures. We start by describing the maximized objective function and then we derive the proposed EM-like algorithm to estimate the regression mixture model parameters. The proposed approach was initially described in part in [47] where the proposed algorithm was derived for the PRM and first results on simulated curves have been provided. Here, we derive the proposed approach for PRM, SRM, and bSRM. We provide additional technical details including on model selection, an extensive experiments on additional benchmark and several real-world data.

3.1. Penalized maximum likelihood estimation

For estimating the regression mixture model (2), we attempt to maximize a penalized log-likelihood function rather than the standard observed-data log-likelihood (3). This criterion consists in penalizing the observed-data log-likelihood (3) by a term accounting for the model complexity. As the model complexity is related to particularly the number of clusters and therefore the structure of the hidden variables z_i (recall that z_i represents the class label of the ith curve), we chose to use the entropy of the hidden variable z_i as penalty. The penalized log-likelihood criterion is therefore derived as follows. The (differential) entropy of z_i is defined by:

$$H(z_i) = -\sum_{k=1}^{K} p(z_i = k) \log p(z_i = k) = -\sum_{k=1}^{K} \pi_k \log \pi_k. \tag{13}$$

By assuming that the variables $\mathbf{z} = (z_1, \ldots, z_n)$ are i.i.d, which is in general the assumption in clustering using mixtures where the cluster labels are assumed to be distributed according to a Multinomial distribution, the whole entropy for $\mathbf{z}$ is therefore additive and we have

$$H(\mathbf{z}) = -\sum_{i=1}^{n} \sum_{k=1}^{K} \pi_k \log \pi_k. \tag{14}$$

The penalized log-likelihood function we propose to maximize is thus constructed by penalizing the observed-data log-likelihood (3) by the entropy term (14), that is

$$\mathcal{J}(\lambda, \boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) - \lambda H(\mathbf{z}), \quad \lambda \geq 0, \tag{15}$$

which leads to the following penalized log-likelihood criterion:

$$\mathcal{J}(\lambda, \boldsymbol{\theta}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{y}_i; \mathbf{X}_i \boldsymbol{\beta}_k, \sigma_k^2 \mathbf{I}_{m_i}) + \lambda n \sum_{k=1}^{K} \pi_k \log \pi_k, \tag{16}$$

where $\mathcal{L}(\boldsymbol{\theta})$ is the observed-data log-likelihood maximized by the standard EM algorithm for regression mixtures (see Equation (3)) and $\lambda \geq 0$ is a parameter that controls the complexity of the fitted model. This penalized log-likelihood function (16) we attempt to optimize allows to control the complexity of the model fit through the roughness penalty $\lambda n \sum_{k=1}^{K} \pi_k \log \pi_k$. The entropy term $-n \sum_{k=1}^{K} \pi_k \log \pi_k$ in the penalty measures the complexity of a fitted model for K clusters. When the entropy is large, the fitted model is rougher, and when it is small, the fitted model is smoother. The non-negative smoothing parameter λ is for establishing a trade-off between closeness of fit to the data and a smooth fit. As λ decreases, the fitted model tends to be less complex, and we get a smoother fit. However, when λ increases, the result is a rougher fit. In the next section, we discuss how to set this regularization coefficient in an adaptive way.

The next section presents the proposed robust EM-like algorithm to maximize the penalized observed-data log-likelihood $\mathcal{J}(\lambda, \boldsymbol{\theta})$ for regression mixture density estimation and model-based curve clustering.

3.2. Robust EM-like algorithm for model-based curve clustering using regression mixtures

Given an i.i.d training data set of n curves $\mathcal{D} = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n))$, the penalized log-likelihood (16) is iteratively maximized by using the following robust EM-like algorithm for model-based curve clustering. Before giving the EM steps, we give the penalized complete-data log-likelihood, on which the EM formulation is based. The complete-data log-likelihood, in this penalized case, is given by:

$$\mathcal{J}_c(\lambda, \boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\log[\pi_k \mathcal{N}(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_k, \sigma_k^2\mathbf{I}_{m_i})] + \lambda n \sum_{k=1}^{K}\pi_k \log \pi_k, \tag{17}$$

where z_{ik} is an indicator binary-valued variable such that $z_{ik} = 1$ if $z_i = k$ (i.e. if the ith observation $(\mathbf{x}_i, \mathbf{y}_i)$ is generated from the kth regression mixture component) and $z_{ik} = 0$ otherwise. After starting with an initial solution (see Section 3.3 for the initialization strategy and stopping rule), the proposed algorithm alternates between the two following steps until convergence.

3.2.1. E-step

This step computes the expectation of the penalized complete-data log-likelihood (17) over the hidden data $\mathbf{z}$, given the observed data $\mathcal{D}$ and a current parameter estimation $\boldsymbol{\theta}^{(q)}$, q being the current iteration number:

$$\begin{aligned}
Q(\lambda, \boldsymbol{\theta}; \boldsymbol{\theta}^{(q)}) &= \mathbb{E}\big[\mathcal{J}_c(\lambda, \boldsymbol{\theta})|\mathcal{D}; \boldsymbol{\theta}^{(q)}\big] \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}\big[z_{ik}|\mathcal{D}; \boldsymbol{\theta}^{(q)}\big]\log[\pi_k \mathcal{N}(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_k, \sigma_k^2\mathbf{I}_{m_i})] + \lambda n \sum_{k=1}^{K}\pi_k \log \pi_k \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(q)}\log[\pi_k \mathcal{N}(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_k, \sigma_k^2\mathbf{I}_{m_i})] + \lambda n \sum_{k=1}^{K}\pi_k \log \pi_k,
\end{aligned} \tag{18}$$

where

$$\tau_{ik}^{(q)} = p(z_i = k|\mathbf{y}_i, \mathbf{x}_i; \boldsymbol{\theta}^{(q)}) = \frac{\pi_k^{(q)}\mathcal{N}\big(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_k^{T(q)}, \sigma_k^{2(q)}\mathbf{I}_{m_i}\big)}{\sum_{h=1}^{K}\pi_h^{(q)}\mathcal{N}(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_h^{(q)}, \sigma_h^{2(q)}\mathbf{I}_{m_i})} \tag{19}$$

is the posterior probability that the curve $(\mathbf{x}_i, \mathbf{y}_i)$ is generated by the kth cluster. This step therefore only requires the computation of the posterior cluster probabilities $\tau_{ik}^{(q)}$ $(i = 1, \ldots, n)$ for each of the K clusters.

3.2.2. M-step

This step updates the value of the parameter vector $\boldsymbol{\theta}$ by maximizing the Q-function (20) with respect to $\boldsymbol{\theta}$, that is by computing the parameter vector update $\boldsymbol{\theta}^{(q+1)} = \arg\max_{\boldsymbol{\theta}} Q(\lambda, \boldsymbol{\theta}; \boldsymbol{\theta}^{(q)})$. By decomposing Equation (20) as:

$$Q(\lambda, \boldsymbol{\theta}; \boldsymbol{\theta}^{(q)}) = Q_\pi(\lambda, \pi_1, \ldots, \pi_K; \boldsymbol{\theta}^{(q)}) + \sum_{k=1}^{K} Q_{\boldsymbol{\theta}_k}(\lambda, \boldsymbol{\beta}_k, \sigma_k^2; \boldsymbol{\theta}^{(q)}), \tag{20}$$

where

$$Q_\pi(\lambda, \pi_1, \ldots, \pi_K; \boldsymbol{\theta}^{(q)}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(q)}\log \pi_k + \lambda n \sum_{k=1}^{K}\pi_k \log \pi_k \tag{21}$$

and

$$Q_{\boldsymbol{\theta}_k}(\lambda, \boldsymbol{\beta}_k, \sigma_k^2; \boldsymbol{\theta}^{(q)}) = \sum_{i=1}^{n} \tau_{ik}^{(q)} \log \mathcal{N}(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_k, \sigma_k^2 \mathbf{I}_{m_i})$$

$$= \sum_{i=1}^{n} \tau_{ik}^{(q)} \left(-\frac{m_i}{2} \log 2\pi - \frac{m_i}{2} \log \sigma_k^2 - \frac{1}{2\sigma_k^2} \parallel \mathbf{y}_i - \mathbf{X}_i\boldsymbol{\beta}_k \parallel^2 \right), \quad (22)$$

it follows the maximization of the Q-function can be performed by maximizing separately $Q_\pi(\lambda, \pi_1, \ldots, \pi_K; \boldsymbol{\theta}^{(q)})$ with respect to the mixing proportions $(\pi_1, \ldots, \pi_K)$ and, for each component k, $Q_{\boldsymbol{\theta}_k}(\lambda, \boldsymbol{\beta}_k, \sigma_k^2; \boldsymbol{\theta}^{(q)})$ with respect to the regression parameters $\{\boldsymbol{\beta}_k, \sigma_k^2\}$.

The mixing proportions updates are obtained by maximizing Equation (21) with respect to the mixing proportions $(\pi_1, \ldots, \pi_K)$ subject to the constraint $\sum_{k=1}^{K} \pi_k = 1$. This can be solved using Lagrange multipliers and the obtained updating formula is given by (details are given in Appendix 2):

$$\pi_k^{(q+1)} = \frac{1}{n} \sum_{i=1}^{n} \tau_{ik}^{(q)} + \lambda \pi_k^{(q)} \Big(\log \pi_k^{(q)} - \sum_{h=1}^{K} \pi_h^{(q)} \log \pi_h^{(q)} \Big) \cdot \quad (23)$$

We notice here that the update of the mixing proportions (23) is close to the standard EM update $((1/n) \sum_{i=1}^{n} \tau_{ik}^{(q)}$ see Equation (5)) for very small value of λ. However, for a large value of λ, the penalization term will play its role in order to make clusters competitive and thus allows for discarding invalid clusters and enhancing actual clusters. Indeed, in the updating formula (23), we can see that for cluster k if

$$\log \pi_k^{(q)} - \sum_{h=1}^{K} \pi_h^{(q)} \log \pi_h^{(q)} > 0, \quad (24)$$

that is, for the (logarithm of the) current proportion $\log \pi_k^{(q)}$, the entropy of the hidden variables is decreasing, and therefore the model complexity tends to be stable, the cluster k has therefore to be enhanced. This explicitly results in the fact that the update of the kth mixing proportion $\pi_k^{(q+1)}$ in Equation (23) will increase. On the other hand, if Equation (24) is less than 0, the cluster proportion will therefore decrease as it is not very informative in the sense of the entropy.

The regularization parameter λ is updated so that it is large when the difference between the current and the previous values of the mixing proportions π_k is small, in order to enhance the competition, and it is small when that difference is large, in order to keep the partition stable. A simple updating formula to take this into account is the exponential of minus the magnitude the difference between the current and the previous values of the mixing proportions, that is, $\lambda^{(q+1)} = \mathrm{e}^{(-\kappa|\pi_k^{(q+1)} - \pi_k^{(q)}|)}$, where κ is a chosen positive constant which characterizes the variation speed of the parameter. Hence, for all the K components, the resulting updating formula is the normalized sum over the K components, that is

$$\lambda^{(q+1)} = \frac{\sum_{k=1}^{K} \mathrm{e}^{(-\kappa|\pi_k^{(q+1)} - \pi_k^{(q)}|)}}{K}.$$

The constant κ can be fixed by taking into account the amount of the data, that is, mainly the sample size, and also the number of observations per individual. For example it can be set to ηn, n being the sample size and η a positive constant set which can be set to $\min(1, 0.5^{\lfloor m/2-1 \rfloor})$ as in [28], m being the average number of observations per curve and $\lfloor x \rfloor$ denotes the largest integer that is no more than x. Then, since the updated mixing proportions in Equation (23) directly depend on the new value of the regularization parameter λ, we must also take into account the constraint that the updated mixing proportions are non-negative and sum to 1, that is, $0 \leq \pi_k \leq 1$, $\sum_{k=1}^{K} \pi_k = 1$. Based on the study of the function $\pi_k \log \pi_k$ in Equation (23), it can be easily shown as described in [28], that the updated value of λ must not exceed $1 - \max_{k=1}^{K}((\sum_{i=1}^{n} \tau_{ik}^{(q)})/n)/ - \max_{k=1}^{K} \pi_k^{(q)} \sum_{k=1}^{K} \pi_k^{(q)} \log \pi_k^{(q)}$ to namely prevent the mixing proportions exceeding one. Finally, it follows that the adaptive formula for updating the penalization coefficient λ is given as in [28] by:

$$\lambda^{(q+1)} = \min \left\{ \frac{\sum_{k=1}^{K} \mathrm{e}^{(-\eta n|\pi_k^{(q+1)} - \pi_k^{(q)}|)}}{K}, \frac{1 - \max_{k=1}^{K}(\sum_{i=1}^{n} \tau_{ik}^{(q)}/n)}{-\max_{k=1}^{K} \pi_k^{(q)} \sum_{k=1}^{K} \pi_k^{(q)} \log \pi_k^{(q)}} \right\}. \tag{25}$$

Thus, the penalization coefficient λ is set as described previously in such a way to be large for enhancing competition when the proportions are not increasing enough from one iteration to another. In this case, the robust algorithm plays its role for estimating the number of clusters (which is decreasing in this case by discarding small invalid clusters). We note that here a cluster k can be discarded if its proportion is less than $1/n$, that is $\pi_k^{(q)} < 1/n$. On the other hand, λ has to become small when the proportions are sufficiently increasing as the learning proceeds and the partition can therefore be considered as stable. In this case, the robust EM-like algorithm tends to have the same behaviour as the standard EM described in Section 2.1. The regularization coefficient λ is also set in $[0, 1]$ to prevent very large values.

The regression parameters $(\boldsymbol{\beta}_k, \sigma_k^2)$ are updated by separately maximizing for each component k function (22). This maximization consists in analytically solving a weighted least-squares problem where the weights are the posterior cluster probabilities $\tau_{ik}^{(q)}$ and provides the following parameter updates:

$$\boldsymbol{\beta}_k^{(q+1)} = \left[\sum_{i=1}^{n} \tau_{ik}^{(q)} \mathbf{X}_i^T \mathbf{X}_i \right]^{-1} \sum_{i=1}^{n} \tau_{ik}^{(q)} \mathbf{X}_i^T \mathbf{y}_i, \tag{26}$$

$$\sigma_k^{2(q+1)} = \frac{1}{\sum_{i=1}^{n} \tau_{ik}^{(q)} m_i} \sum_{i=1}^{n} \tau_{ik}^{(q)} \parallel \mathbf{y}_i - \mathbf{X}_i \boldsymbol{\beta}_k^{(q+1)} \parallel^2, \tag{27}$$

where the posterior cluster probabilities $\tau_{ik}^{(q)}$ given by Equation (19) are computed using the mixing proportions derived in Equation (23).

Then, once the model parameters have been estimated, a fuzzy partition of the data into K clusters, represented by the estimated posterior cluster probabilities $\hat{\tau}_{ik}$, is obtained. A hard partition can also be computed according to the MAP principle by maximizing the posterior cluster probabilities according to Equation (12).

3.3. Initialization strategy and stopping rule

The initial number of clusters is $K^{(0)} = n$, n being the total number of curves and the initial mixing proportions are $\pi_k^{(0)} = 1/K^{(0)}$, $(k = 1, \ldots, K^{(0)})$. Then, to initialize the regression parameters $\boldsymbol{\beta}_k$ and the noise variances σ_k^2 $(k = 1, \ldots, K^{(0)})$, we fitted a polynomial regression model on each curve k, $(k = 1, \ldots, K^{(0)})$; The initial values of the regression parameters are thus given by $\boldsymbol{\beta}_k^{(0)} = (\mathbf{X}_k^{\mathrm{T}} \mathbf{X}_k)^{-1} \mathbf{X}_k \mathbf{y}_k$ and the noise variance can be deduced as $\sigma_k^{2(0)} = 1/m_k \parallel \mathbf{y}_k - \mathbf{X}_k \boldsymbol{\beta}_k^{(0)} \parallel^2$. To avoid singularities at the starting point, we set $\sigma_k^{2(0)}$ as a middle value in the following sorted range $\parallel \mathbf{y}_k - \mathbf{X}_k \boldsymbol{\beta}_k^{(0)} \parallel^2$ for $k = 1, \ldots, n$. The algorithm is stopped when the maximum variation of the estimated regression parameters between two iterations $\max_{1 \leq k \leq K^{(q)}} \parallel \boldsymbol{\beta}_k^{(q+1)} - \boldsymbol{\beta}_k^{(q)} \parallel$ was less than a fixed threshold υ (e.g. 10^{-6}).

The pseudo code 2 summarizes the proposed robust EM-like algorithm for model-based curve clustering using regression mixtures.

Algorithm 2 Pseudo code of the proposed robust EM-like algorithm for regression mixtures.

```
Inputs: Data $\mathcal{D} = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n))$ and polynomial degree $p$
$\upsilon \leftarrow 10^{-6}$; $q \leftarrow 0$; $converge \leftarrow 0$
//Initialization:
$K^{(0)} = n$
$\lambda^{(0)} = 1$
$\boldsymbol{\theta}^{(0)} = (\pi_1^{(0)}, \ldots, \pi_K^{(0)}, \boldsymbol{\theta}_1^{(0)}, \ldots, \boldsymbol{\theta}_K^{(0)})$
for $k = 1, \ldots, K^{(q)}$ do
    Compute $\tau_{ik}^{(q)}$ for $i = 1, \ldots, n$ using Equation (19)
end for
//main loop:
while (! $converge$) do
    for $k = 1, \ldots, K^{(q)}$ do
        Compute $\tau_{ik}^{(q)}$ for $i = 1, \ldots, n$ using Equation (19)
    end for
    for $k = 1, \ldots, K^{(q)}$ do
        Compute $\pi_k^{(q+1)}$ using Equation (23)
    end for
    Compute $\lambda^{(q+1)}$ using Equation (25)
    Discard invalid clusters with small proportions $\pi_k^{(q)} < \frac{1}{n}$ ; Set $K^{(q+1)} = K^{(q)} - \#\{\pi_k^{(q)} | \pi_k^{(q)} < \frac{1}{n}\}$ ; normalize $\tau_{ik}^{(q+1)}$ and $\pi_k^{(q+1)}$ so that they sum to one
    if  the partition is stabilized : if $K^{(q+1)} - K^{(q+1-nIter)} = 0$ (e.g. nIter = 50) then
        set $\lambda^{(q)} = 0$
    end if
    for $k = 1, \ldots, K^{(q)}$ do
        Compute $\boldsymbol{\beta}_k^{(q+1)}$ using Equation (26)
        Compute $\sigma_k^{2(q+1)}$ using Equation (27)
    end for
    for $k = 1, \ldots, K^{(q)}$ do
        Compute $\tau_{ik}^{(q)}$ for $i = 1, \ldots, n$ using Equation (19)
    end for
    for $k = 1, \ldots, K^{(q)}$ do
        Compute $\boldsymbol{\beta}_k^{(q+1)}$ using Equation (26)
    end for
    if $\max_{1 \leq k \leq K^{(q)}} \parallel \boldsymbol{\beta}_k^{(q+1)} - \boldsymbol{\beta}_k^{(q)} \parallel < \epsilon$ then
        $converge = 1$
    end if
    $q \leftarrow q + 1$
end while
Outputs: $\hat{K} = K^{(q)}$,   $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(q)}$,   $\hat{\tau}_{ik} = \tau_{ik}^{(q)}$
```

3.4. Choosing the order of regression and spline knots number and locations

For a general use of the proposed algorithm for the PRM, the order of regression can be chosen by cross-validation techniques as in [16]. In our experiments, we report the results corresponding to the degree for which the PRM provides the best fit. However, in some situations, the PRM model may be too simple to capture the full structure of the data, in particular for curves with high nonlinearity or with regime changes, even if they can be seen as providing a useful first-order approximation of the data structure. The (B-)spline regression models in this case are more adapted. For these models, one may need to choose the spline order as well as the number of knots and their locations. For the order of regression in (B-)splines, we notice that, in practice, the most widely used orders are $M = 1, 2,$ and 4.[35] For smooth function approximation, cubic (B-)splines, which correspond to a (B-)spline of order 4 and thus with twice continuous derivatives, are sufficient to approximate smooth functions. When the data present irregularity, such as a kind of piecewise non-continuous functions, a linear spline (of order 2) should be more adapted. This was namely used for the satellite data set. The order 1 can be chosen for piecewise constant data. Concerning the choice of the number of knots and their locations, a common choice is to place a number of knots uniformly spaced across the range of x. In general more knots are needed for functions with high nonlinearity or regime changes. One can also use automatic techniques for the selection of the number of knots and their locations as reported in [16]. For example, this can be performed by using cross validation as in [34]. In [48], the knots are placed at selected order statistics of the sample data and the number of knots is determined either by a simple rule or by minimizing a variant of AIC. The general goal is to use a sufficient number of knots to fit the data while at the same time to avoid over-fitting and to not make the computation excessive. The current algorithm can be easily extended to handle this type of automatic selection of spline knots placement, but as the unsupervised clustering problem itself requires much attention and is difficult, it is wise to fix the number and location of knots. In this paper, we will use knot sequences which are uniformly spaced across the range of x. The studied problems are not very sensitive to the number and location of knots. Few number of equispaced knots (less than ten for the data studied in this paper) are sufficient to fit the data.

4. Experimental study

This section is dedicated to the evaluation of the proposed approach on simulated data and real-world data. The algorithms have been implemented in MATLAB[1]. We evaluate the proposed robust algorithm for the three regression mixture models, that is, PRM, SRM, and bSRM, respectively, abbreviated as PRM, SRM, and bSRM. The evaluation is performed in terms of estimating the actual partition by considering the estimated number of clusters and the clustering accuracy (misclassification error). First results on simulated arbitrary nonlinear curves as well as curves from a mixture of linear regressions have been presented in [47] and show the potential benefit of the proposed algorithm. Here we perform several additional experiments and consider the three regression mixture models. We first consider simulated data and the waveforms benchmark of Breiman et al.[49] Then, we consider three real-world data sets covering three different application area: phoneme recognition, clustering gene expression time course data for bio-informatics and clustering radar waveform data.

4.1. Simulation study

This section is dedicated to the evaluation of the proposed approach on simulated data. We consider the waveform curves of Brieman and data simulated according to two different simulation scenarios.

In the first experiment, we consider the waveform data introduced in [49] and studied in [50] and elsewhere. The waveform data consist in a three-class problem where each curve is generated as follows:

- $\mathbf{y}_1(t) = uh_1(t) + (1-u)h_2(t) + \epsilon_t$ for the class 1;
- $\mathbf{y}_2(t) = uh_2(t) + (1-u)h_3(t) + \epsilon_t$ for the class 2;
- $\mathbf{y}_3(t) = uh_1(t) + (1-u)h_3(t) + \epsilon_t$ for the class 3;

where u is a uniform random variable on $(0,1)$, $h_1(t)=\max(6-|t-11|,0)$; $h_2(t)=h_1(t-4)$; $h_3(t)=h_1(t+4)$ and ϵ_t is a zero-mean Gaussian noise with unit standard deviation. The temporal interval considered for each curve is $[1;21]$ with a constant period of sampling of 1 s. Figure 1 shows the waveform data mean functions from the generative model before the Gaussian noise is added and a sample of 150 waveforms. Figures 2–4, respectively, show the corresponding obtained clustering results for the waveform data for the PRM, SRM, and bSRM. Each sub-figure corresponds to a cluster. The solid line corresponds to the estimated mean curve and the dashed lines correspond to the confidence region computed as plus and minus twice the estimated standard deviation. The number of clusters is correctly estimated by the proposed algorithm for three models. For this data, the spline regression models provide slightly better results in terms of clusters approximation than the PRM (here $p=4$). This can be seen for the third cluster. Table 1 presents the clustering results averaged over 20 different sample of 500 curves. It includes the estimated number of clusters, the misclassification error rate, and the absolute error between the true clusters proportions and variances and the estimated ones. We compared the algorithm for the proposed models to two standard clustering algorithms: K-means clustering, and the EM clustering using GMMs. The GMM density of the observations was assumed $f(\mathbf{y}_i;\boldsymbol{\mu}_k,\sigma_k^2)=\sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{y}_i;\boldsymbol{\mu}_k;\sigma_k^2\mathbf{I}_{m_i})$. We note that, for these two algorithms, the number of clusters was fixed to the actual one. The number of clusters in this case

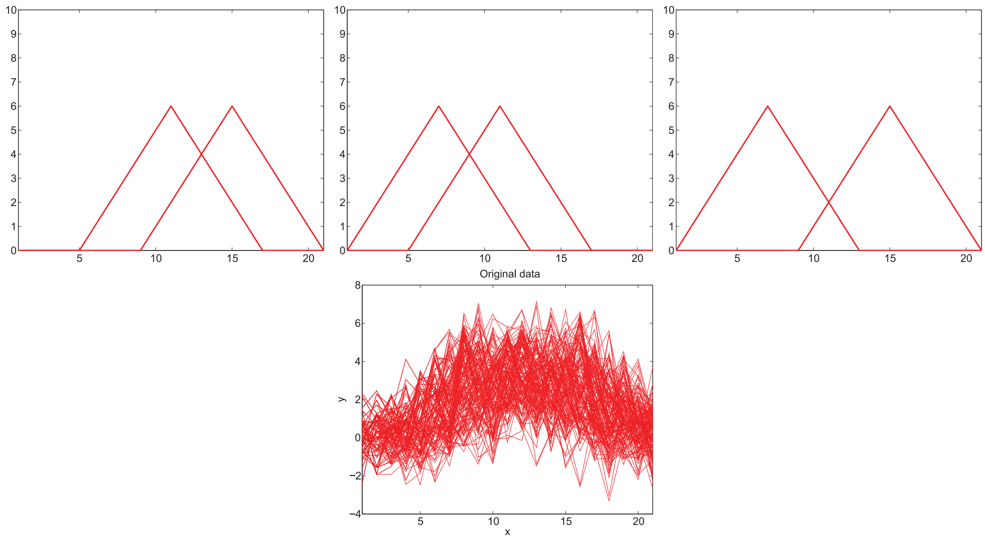


Figure 1. The waveform mean functions from the generative model before the Gaussian noise is added (up), and a sample of simulated waveform data (bottom).

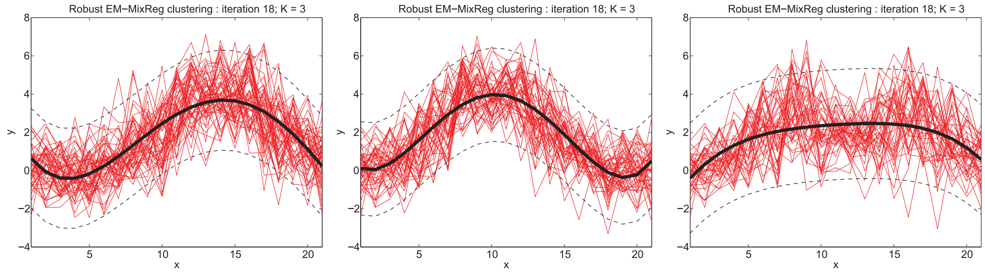


Figure 2. Clustering results obtained by the proposed robust EM-like algorithm and the PRM model (polynomial degree $p=4$) for the waveform data. Each sub-figure corresponds to a cluster.

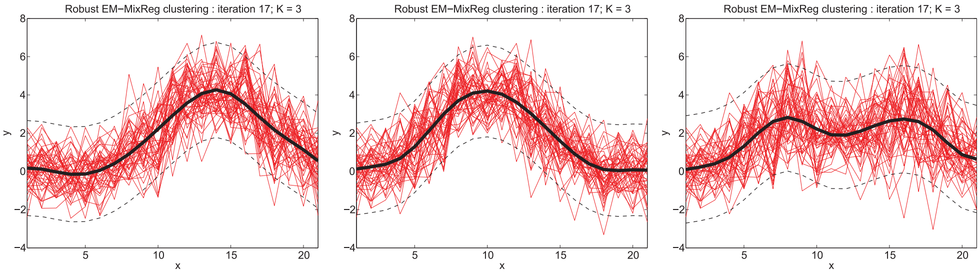


Figure 3. Clustering results obtained by the proposed robust EM-like algorithm and the SRM model with a cubic-spline of three knots for the waveform data. Each sub-figure corresponds to a cluster.

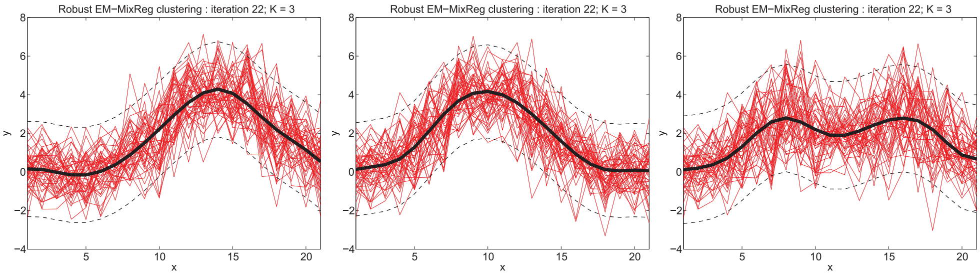


Figure 4. Clustering results obtained by the proposed robust EM-like algorithm and the bSRM model with a cubic b-spline of three knots for the waveform data. Each sub-figure corresponds to a cluster.

Table 1. Clustering results for the waveform data.

	Actual	*K*-means	Stand. EM–GMM	EM–PRM	EM–SRM	EM–bSRM
K	3	–	–	3	3	3
misc. error	–	6.2 ± (0.24)%	5.90 ± (0.23)%	4.31 ± (0.42)%	2.94 ± (0.88)%	2.53 ± (0.70)
σ_1	1	0.164 ± (0.013)	0.14 ± (0.018)	0.128 ± (0.015)	0.108 ± (0.015)	0.103 ± (0.012)
σ_2	1	0.131 ± (0.014)	0.124 ± (0.015)	0.102 ± (0.015)	0.090 ± (0.011)	0.079 ± (0.010)
σ_3	1	0.264 ± (0.027)	0.245 ± (0.025)	0.223 ± (0.021)	0.180 ± (0.014)	0.141 ± (0.013)
π_1	$\frac{1}{3}$	0.0043 ± (0.002)	0.0041 ± (0.002)	0.0037 ± (0.0018)	0.0035 ± (0.0015)	0.0034 ± (0.0015)
π_2	$\frac{1}{3}$	0.0036 ± (0.0028)	0.0034 ± (0.0019)	0.0029 ± (0.0023)	0.0018 ± (0.0015)	0.0012 ± (0.0011)
π_3	$\frac{1}{3}$	0.0047 ± (0.0049)	0.0043 ± (0.0058)	0.0040 ± (0.0062)	0.0037 ± (0.0015)	0.0035 ± (0.0014)

can be chosen by using model selection criteria such as the BIC. This requires however an afterward step which consists in selecting a model from pre-estimated models with different number of components. One can observe that for all the models, the actual number of clusters is correctly retrieved. The misclassification error rate as well as the parameter estimation errors are slightly better for the spline regression models, in particular the bSRM. On the other hand, it can be seen that the regression mixture models with the proposed EM-like algorithm outperform the standard *K*-means and EM–GMM clustering algorithms. In addition, notice that for the standard EM algorithm, when the number of mixture components is not fixed by the user, its estimation is in general performed in a two-fold procedure, that is, the estimation of several mixture models with varying number of components, followed by a model selection step using selection criteria. however, the proposed algorithm simultaneously infers the model and its optimal number of components. In Figure 5, one can see the variation of the estimated number of clusters as well as the value of the objective function from one iteration to another for the three models. These results highlight the capability of the proposed algorithm to provide an accurate partition with an optimal number of clusters.

To assess the behaviour of the proposed approach in terms of the number of observations, the dimension of each observation, as well as the number of clusters in the data, the cluster and the

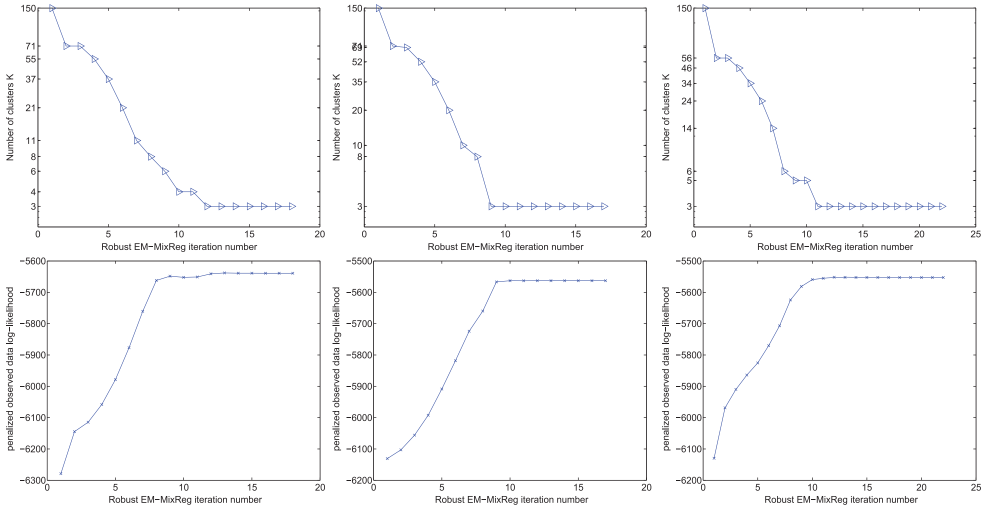


Figure 5. Variation of the number of clusters and the value of the objective function during the iterations of the algorithm for the PRM (left), SRM (middle), and bSRM (right) for the waveform data.

cluster proportions, we considered the following simulation scenarios. Simulations $S1$ were designed to assess the capacity of the proposed approach to retrieve partitions with a small number of clusters while simulations $S2$ were designed to retrieve partitions with a large number of clusters. In the first scenario $S1$, we simulated data from a small number of clusters ($K = 3$), including one well separated cluster and two poorly separated clusters. The data were generated according to the PRM model (2) as follows:

- $\mathbf{y}_i = -0.1\mathbf{x}_i + 0.7 + 0.05\boldsymbol{\epsilon}_i$ if $z_i = 1$;
- $\mathbf{y}_i = \mathbf{x}_i + 0.05\boldsymbol{\epsilon}_i$ if $z_i = 2$;
- $\mathbf{y}_i = \mathbf{x}_i + 0.1\boldsymbol{\epsilon}_i$ if $z_i = 3$.

The input $\mathbf{x}_i$ is composed of m ordered equi-spaced points in the range $(0, 1)$, $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$ is a standard multivariate Gaussian noise and the cluster label z_i is simulated according to the multinomial distribution $\mathcal{M}(1; \pi_1, \pi_2, \pi_3)$ with non-uniform mixing proportions which are given by $\pi_1 = 0.2, \pi_2 = \pi_3 = 0.4$. It can be seen from the generative model and in Figure 6 that the first cluster cluster is designed to be well separated from the two others, while the second cluster is completely incorporated within the third cluster. These two clusters have the same mean and only differ according to the variance.

For this scenario, we considered different situations regarding the number of observations and the dimension of each observation. We considered a small sample size $n = 50$ and a large sample size $n = 500$. For each sample size, we considered a small observation dimension $m = 50$ and a large observation dimension $m = 500$. We however note that for curve clustering, the number of observations m per curve is in general expected to be more than 50. Figure 6 shows the obtained results for data simulated according to the first scenario. It can be seen that, for the four situations with different sample size and curve dimension, the true partition is correctly estimated. The merged clusters are also retrieved with success. The model indeed takes into account mixture components with different noise variances (heteroskedastic model).

Table 2 shows the actual and estimated model parameters for the data shown in Figure 6. One can observe that the estimated parameters are very close to the true ones, for each of the considered situations.

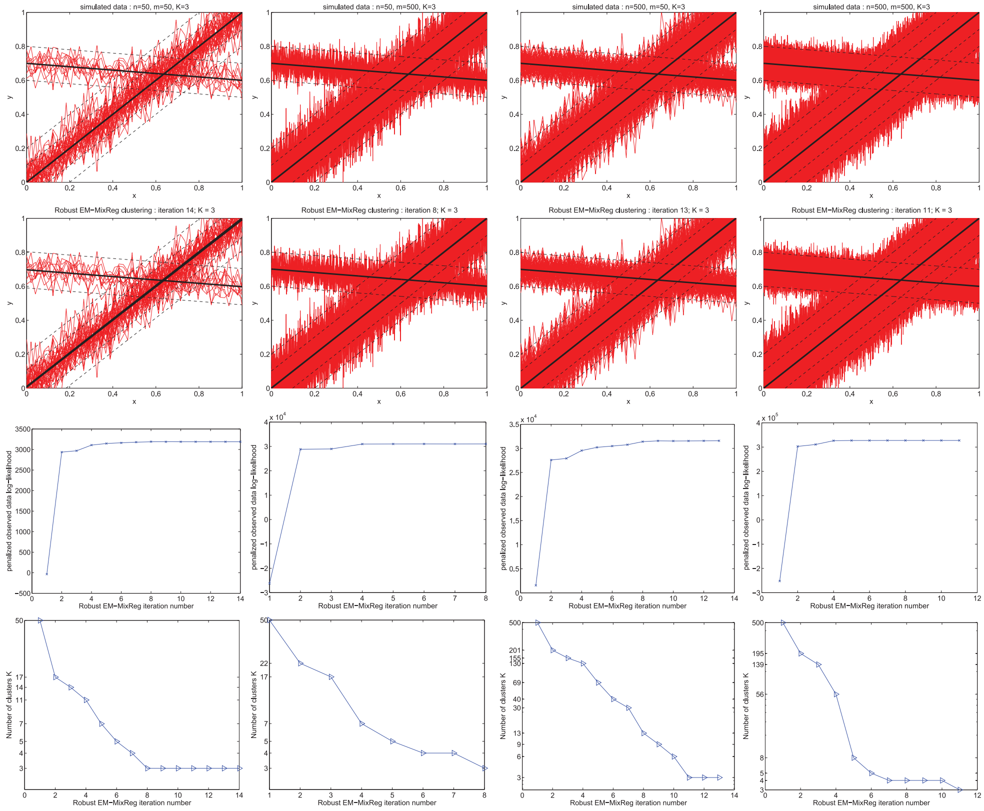


Figure 6. Clustering results obtained by the proposed algorithm and the PRM model for data simulated according to the scenario *S*1 with, from top to bottom, the simulated data with the true partition and the true parameters, their estimated counterparts, the value of the objective function, and the number of clusters during the iterations of the algorithm.

Table 2. Estimated parameters.

Parameter	$\boldsymbol{\beta}_{11}$	$\boldsymbol{\beta}_{10}$	$\boldsymbol{\beta}_{21}$	$\boldsymbol{\beta}_{20}$	$\boldsymbol{\beta}_{31}$	$\boldsymbol{\beta}_{30}$	σ_1	σ_2	σ_3	π_1	π_2	π_3
Actual	-0.1	0.7	1	0	1	0	0.05	0.05	0.1	0.2	0.4	0.4
$n = 50, m = 50$	-0.0994	0.6969	0.9862	0.0057	0.9913	0.0104	0.0533	0.0517	0.0958	0.2	0.44	0.36
$n = 50, m = 500$	-0.1007	0.7012	0.9978	0.0013	1.0011	-0.0000	0.0500	0.0499	0.1001	0.18	0.34	0.48
$n = 500, m = 50$	-0.0988	0.6994	1.0002	-0.0003	1.0036	-0.0016	0.0498	0.0503	0.1000	0.218	0.3919	0.3901
$n = 500, m =$ 500	-0.1001	0.6999	1.0003	0.0000	0.9992	-0.0002	0.0499	0.0501	0.1000	0.1960	0.4240	0.3800

In the second scenario *S*2, we simulated data from a large number of clusters ($K = 12$), including poorly separated clusters an some well separated clusters. We considered data from curves containing $m = 500$ observations with a sample size $n = 300$ and $n = 500$. The data for this scenario were generated as follows:

- $\mathbf{y}_i = \mathbf{x}_i + 0.1\boldsymbol{\epsilon}_i$ if $z_i = 1$; $\mathbf{y}_i = 0.8\mathbf{x}_i + 0.1 + 0.01\boldsymbol{\epsilon}_i$ if $z_i = 2$; $\mathbf{y}_i = 1.2\mathbf{x}_i - 0.1 + 0.01\boldsymbol{\epsilon}_i$ if $z_i = 3$;
- $\mathbf{y}_i = 0.5 + 0.05\boldsymbol{\epsilon}_i$ if $z_i = 4$; $\mathbf{y}_i = 0.2\mathbf{x}_i + 0.4 + 0.01\boldsymbol{\epsilon}_i$ if $z_i = 5$; $\mathbf{y}_i = -0.2\mathbf{x}_i + 0.6 + 0.01\boldsymbol{\epsilon}_i$ if $z_i = 6$;
- $\mathbf{y}_i = -\mathbf{x}_i + 1 + 0.05\boldsymbol{\epsilon}_i$ if $z_i = 7$; $\mathbf{y}_i = -0.8\mathbf{x}_i + 0.9 + 0.01\boldsymbol{\epsilon}_i$ if $z_i = 8$; $\mathbf{y}_i = -1.2\mathbf{x}_i + 1.1 + 0.01\boldsymbol{\epsilon}_i$ if $z_i = 9$;
- $\mathbf{y}_i = -5\mathbf{x}_i + 3 + 0.1\boldsymbol{\epsilon}_i$ if $z_i = 10$; $\mathbf{y}_i = 5\mathbf{x}_i - 2 + 0.1\boldsymbol{\epsilon}_i$ if $z_i = 11$; $\mathbf{y}_i = -40\mathbf{x}_i + 20.5 + 0.03\boldsymbol{\epsilon}_i$ if $z_i = 12$.

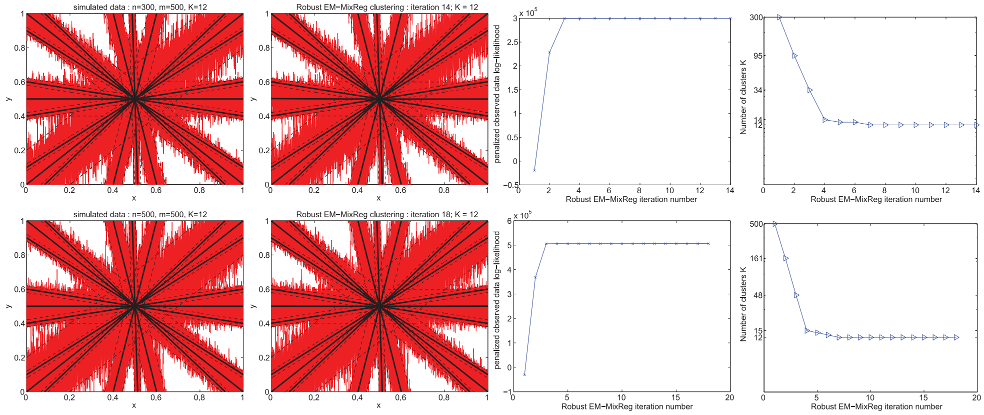


Figure 7. Clustering results obtained by the proposed algorithm and the PRM model for data simulated according to the scenario *S*2 with, from left to right, the simulated data with the true partition and the true parameters, their estimated counterparts, the value of the objective function, and the number of clusters during the iterations of the algorithm.

The clusters in this case have uniform mixing-proportions $\pi_k = 1/K$ for $k = 1, \ldots, 12$. As it can be seen from the generative model and in Figure 7, the clusters 1, 2, and 3 (respectively , the clusters 4, 5, and 6, and the clusters 7, 8, and 9) are poorly separated and are designed to be seen as almost merged into one cluster. The clusters 10, 11, and 12 are quite well separated.

Figure 7 shows the obtained results for data simulated according to the second scenario. It can be seen that the actual partition with 12 clusters is correctly retrieved. The clusters which are not well separated are retrieved with success. We note that for the two scenarios, we used the proposed EM-like algorithm for the PRM model with linear mean functions, as well as the SRM and the bSRM models with linear mean functions and only two boundary knots, which is equivalent to linear fitting. The obtained results for the three models are quasi-identical.

4.2. Experiments on real data

In this section, we consider real data sets to apply and evaluate the proposed approach. The considered data are curves issued from three different application domains: the phonemes data, the yeast cell cycle data, and the Topex/Poseidon satellite data. The curves of each data set are shown in Figure 8.

4.2.1. Phonemes data

In this section, we use the phonemes data set used in [18][2] which is a part of the original one available at http://www-stat.stanford.edu/ElemStatLearn and was described and used namely in [51]. The application context related to this data set is a phoneme classification problem. The phonemes data

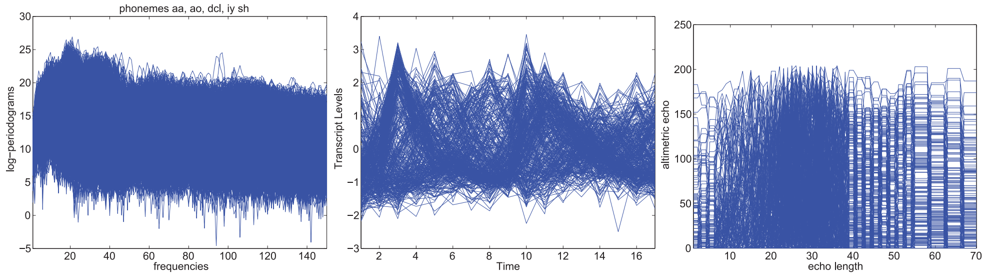


Figure 8. Real data: Phonemes of the classes ‘ao’, ‘aa’, ‘iy’, ‘dcl’, ‘sh’ (left), the yeast cell cycle data (middle) and the Topex/Poseidon satellite data (right).

correspond to log-periodograms y constructed from recordings available at different equispaced frequencies x for different phonemes. The data set contains five classes corresponding to the following five phonemes: 'sh' as in 'she', 'dcl' as in 'dark', 'iy' as in 'she', 'aa' as in 'dark', and 'ao' as in 'water'. For each phoneme we have 400 log-periodograms at a 16-kHz sampling rate. We only retain the first 150 frequencies from each subject as in [18]. This data set has been considered in a phoneme discrimination problem as in [18,51] where the aim is to predict the phoneme class for a new log-periodogram. Here we reformulate the problem into a clustering problem where the aim is to automatically group the phonemes data into classes. We therefore assume that the cluster labels are missing. We also assume that the number of clusters is unknown. Thus, the proposed algorithm will be assessed in terms of estimating both the actual partition and the optimal number of clusters from the data. Figure 8 (left) shows the used 1000 log-periodograms (200 per cluster) and Figure 9 shows the curves of the actual five phoneme classes, class by class.

Figures 10 and 11 show the clustering results for the phonemes log-periodograms obtained by, respectively, the PRM and the bSRM. The SRM results are closely similar to those provided by the bSRM model. The number of phoneme classes (five) is correctly estimated by the three models. The spline regression models provide better results in terms of clusters approximation than the PRM (here $p = 7$). Notice that the value of $p = 7$ corresponds to the PRM model with the best error rate for p varying from 4 to 8. The corresponding misclassification error rate is 14.29 %. The values of the estimated number of clusters and the misclassification error rate corresponding to each of the three models are given in Table 3. One can see that the spline regression mixtures perform better than the PRM. In a general use of functional data modelling, splines are indeed more adapted than simple polynomial modelling. In a similar way as previously, in Figure 12, one can see the variation of the estimated number of clusters as well as the value of the objective function as the learning proceeds. It can be observed that the number of clusters decreases very rapidly from 1000 to 51 for the PRM model, and to 44 for the SRM and bSRM models. The grand majority of invalid clusters is discarded at the beginning of the learning process. Then, the number of clusters gradually decreases from one iteration to another for the three models and the algorithm converges toward a partition with the actual number of clusters for the three models after at most 43 iterations. We can also see from the curve of the number of clusters and the objectives functions that the algorithm for the SRM and

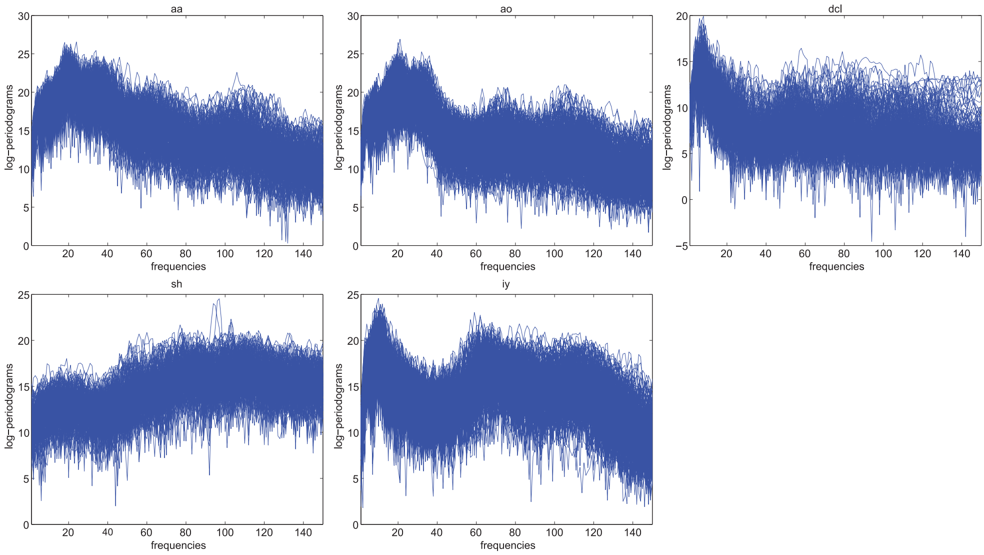


Figure 9. Phonemes data classes: 'ao', 'aa', 'yi', 'dcl', 'sh'.

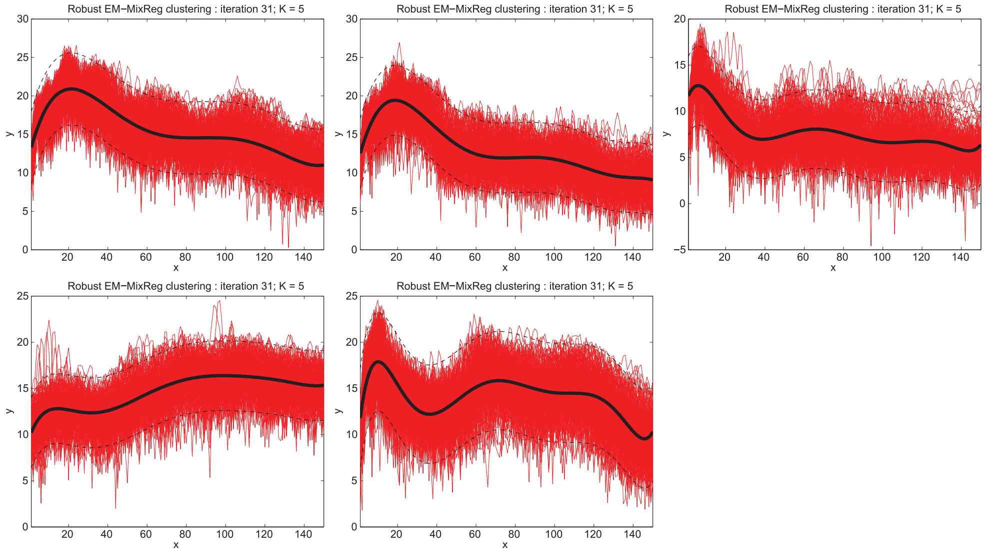


Figure 10. Clustering results obtained by the proposed robust EM-like algorithm and the PRM model (polynomial degree $p = 7$) for the phonemes data. Each sub-figure corresponds to a cluster.

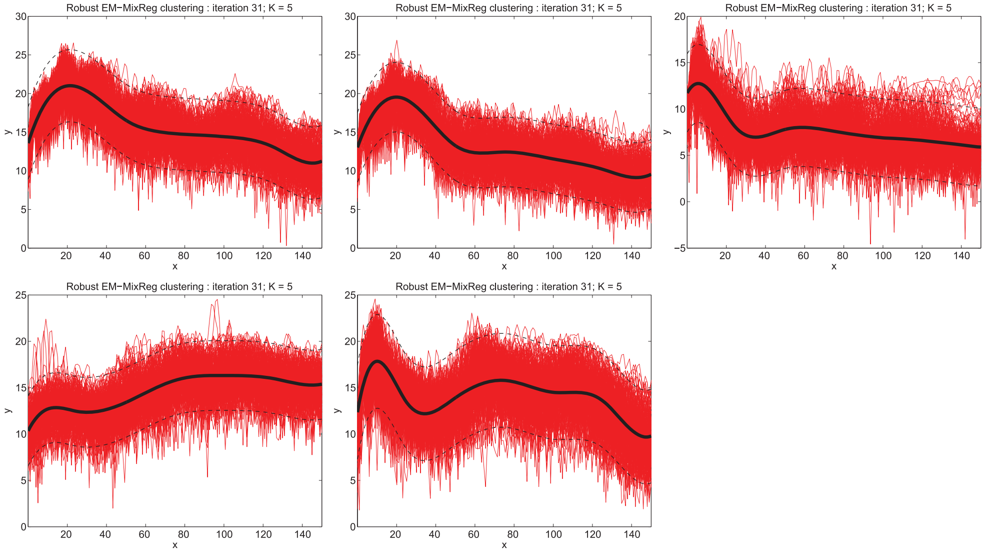


Figure 11. Clustering results obtained by the proposed robust EM-like algorithm and the bSRM model with a cubic B-spline of seven knots for the phonemes data. Each sub-figure corresponds to a cluster.

Table 3. Clustering results for the phonemes data.

	EM–PRM	EM–SRM	EM–bSRM
$\hat{K}$	5	5	5
Misc. error rate	14.29 %	14.09 %	14.2 %

bSRM models behaves in a very similar way. We can also notice that the objective function becomes horizontal once the number of clusters is stabilized.

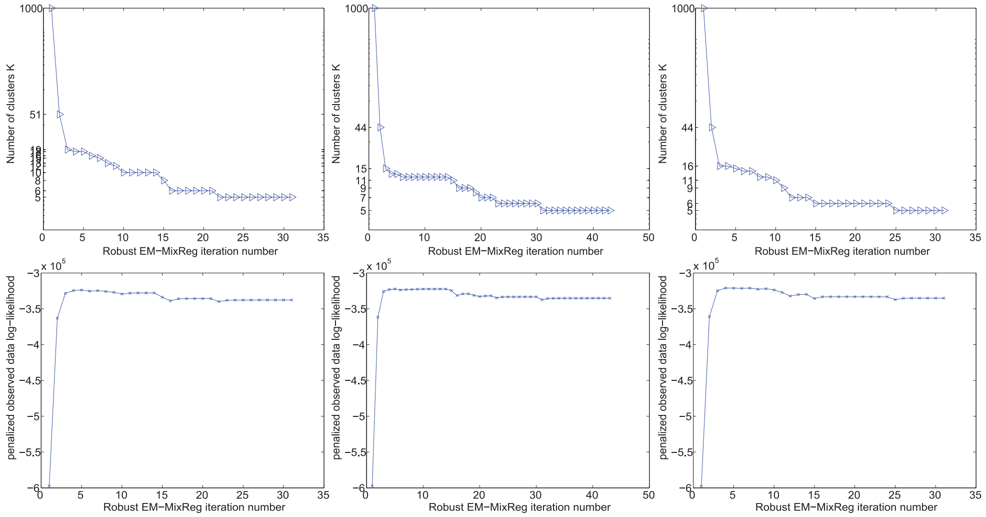


Figure 12. Variation of the number of clusters and the value of the objective function during the iterations of the algorithm for the PRM model (left), the SRM model (middle), and the bSRM model (right) for the phonemes data.

4.2.2. Yeast cell cycle data

In this experiment, we consider the yeast cell cycle data set.[52] The original yeast cell cycle data represent the fluctuation of expression levels of approximately 6000 genes over 17 time points corresponding to two cell cycles.[52] This data set has been used to demonstrate the effectiveness of clustering techniques for time course Gene expression data in bio-informatics such as model-based clustering as in [53]. We used the standardized subset constructed by Yeung et al. [53] available in http://faculty.washington.edu/kayee/model/.[3] This data set referred to as the subset of the 5-phase criterion in [53] contains 384 gene expression levels over 17 time points. The usefulness of the cluster analysis in this case is therefore to automatically reconstruct this five class partition. Figure 8 (middle) shows the 384 curves of the yeast cell cycle data and Figure 13 shows the curves of each of the five clusters. The clustering results are shown in Figures 14 and 15, respectively, for the SRM and bSRM models. Both the PRM and the SRM models provide similar partitions with four clusters. The second and third clusters in Figure 13 (from left to right) look to be merged into the second cluster in Figure 14 (from left to right). Note that some model selection criteria in [53] also provide four clusters in some situations. However, the bSRM model (Figure 15) correctly infers the actual number of clusters. The Rand index (RI)[4] for the obtained partition is 0.7914 which indicates that the partition is quite well defined. Figure 16 shows the variation of the number of clusters and the value of the objective function during the iterations of the algorithm for three models. We can see that the number of clusters starts with $n = 384$ clusters and more than half is discarded after one iteration. Then, it gradually decreases and is stabilized until convergence. The shape of the objective function also becomes horizontal when the partition is converged.

4.2.3. Topex/Poseidon satellite data

The last considered real data set is the Topex/Poseidon radar satellite data set[5] namely used in [9,12]. This data set was registered by the satellite Topex/Poseidon around an area of 25 km upon the Amazon River. The data contain $n = 472$ waveforms of the measured echoes, sampled at $m = 70$ number of echoes. The curves of this data set are shown in Figure 8 (right). The actual number of clusters and the actual partition are unknown for this data set.

Figure 17 shows the obtained clustering results for the bSRM model. The provided solution for the PRM is rather an overall rough approximation and provides three clusters. The polynomial fitting for this type of curves is not adapted. This is because the curves present in particular peaks and transitions. The solutions provided by the SRM and the bSRM are very close and are more informative about the underlying structure of this data set. We used a linear (B-)spline for this data set in order to allow piecewise linear function approximation and thus to better recover the possible peaks and transitions in the curves. As a result, both the SRM and the bSRM provide a five class partition. The partitions are quasi-identical and contain clearly informative clusters. We can see different shapes of waves that summarize the general underlying structure governing this data. We can observe that the first and the second clusters in Figure 17 contain curves presenting one narrow peak. The two clusters however differ with the peak location in x. The third cluster contains curves with one less narrow peak. Then, the fourth cluster contains curves that look to have two large peaks. Finally, the

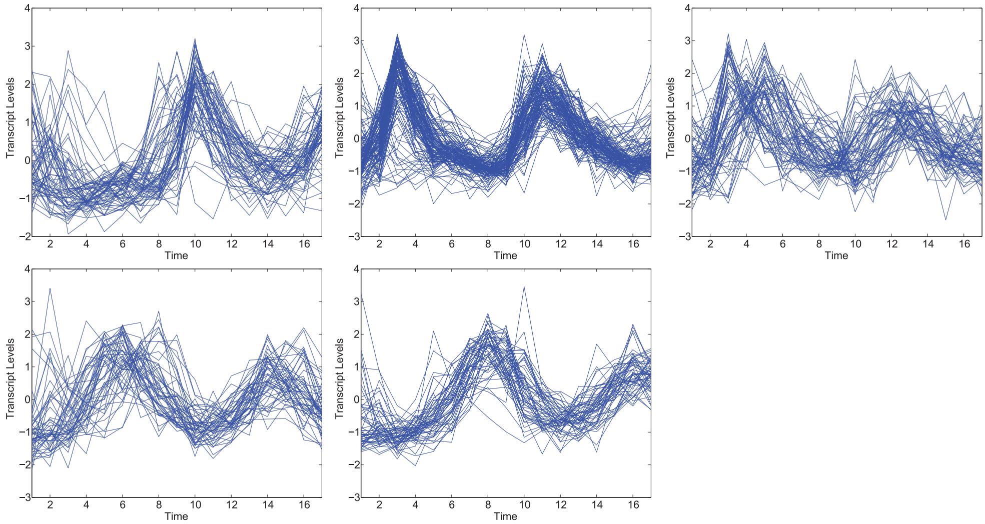
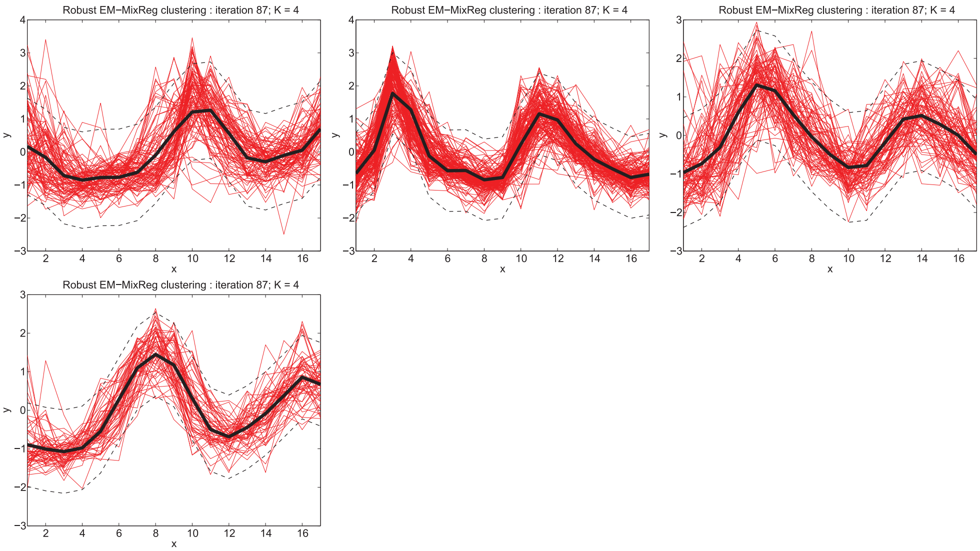


Figure 13. The five actual clusters of the used yeast cell cycle data.


Robust EM−MixReg clustering : iteration 87; K = 4

Robust EM−MixReg clustering : iteration 87; K = 4

Robust EM−MixReg clustering : iteration 87; K = 4

Robust EM−MixReg clustering : iteration 87; K = 4

Figure 14. Clustering results obtained by the proposed robust EM-like algorithm and the SRM model with a cubic spline of seven knots for the yeast cell cycle data. Each sub-figure corresponds to a cluster.

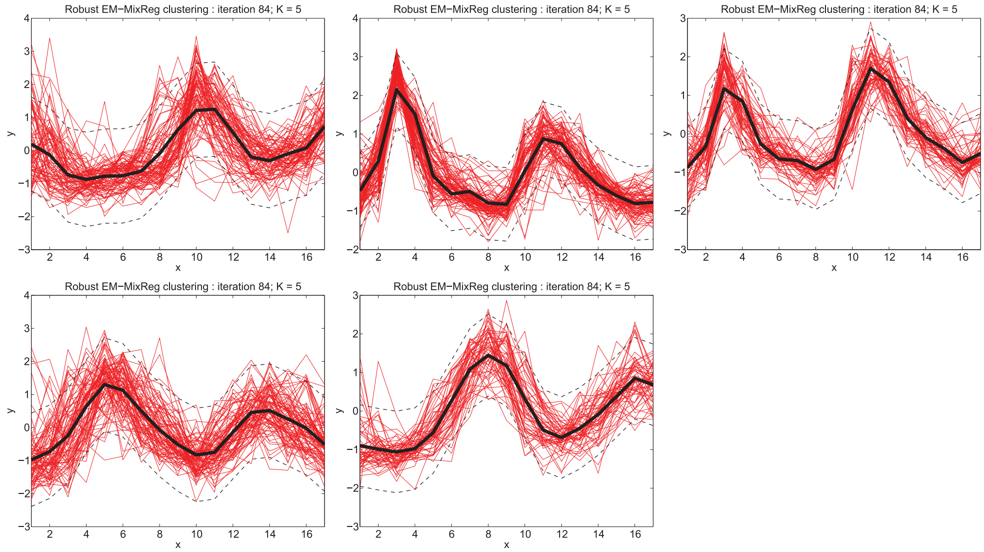


Figure 15. Clustering results obtained by the proposed robust EM-like algorithm and the bSRM model with a cubic B-spline of seven knots for the yeast cell cycle data. Each sub-figure corresponds to a cluster.

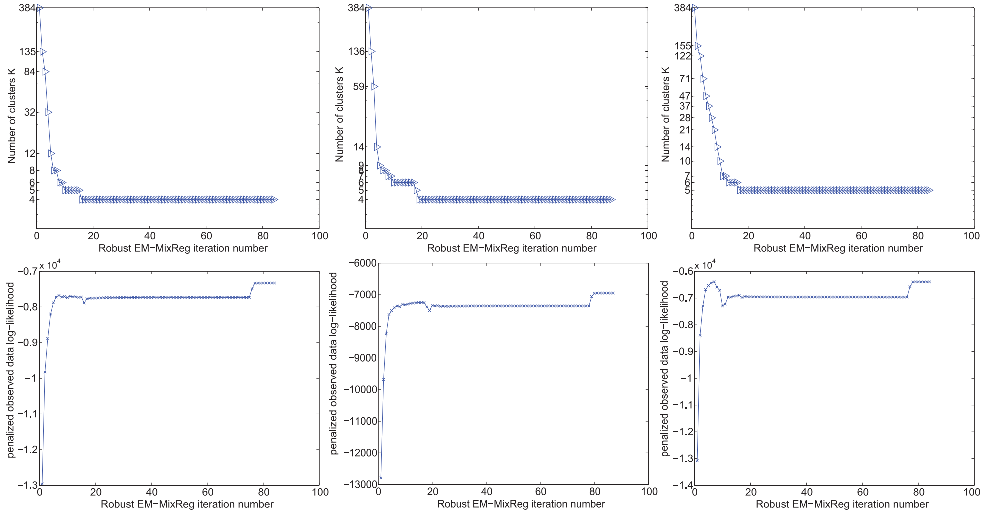


Figure 16. Variation of the number of clusters and the value of the objective function during the iterations of the algorithm for the PRM model (left), the SRM model (middle), and the bSRM model (right) for the yeast data.

fifth cluster looks to contain curves without peaks and with a part rather flat. Furthermore, we can see that the structure is more clear with the cluster mean (prototypes) than with the raw curves. The SRM models thus help to better understand the underlying structure of the data as well as to recover a plausible number of clusters from the data. In addition, the found number of clusters (five) also equals the one found by Dabo-Niang et al. [12] by using another hierarchical non-parametric kernel-based unsupervised classification technique. The mean curves for the five terminal groups reflecting the hidden structure provided by the proposed approach for both the SRM and the bSRM are similar to those in [12]. On the other hand, one can also see that this result is similar to the one found in [9]; Most of the profiles are indeed present in the two results. The slight difference can be attributed to the fact that the results in [9] are provided from a two-stage scheme which includes an additional

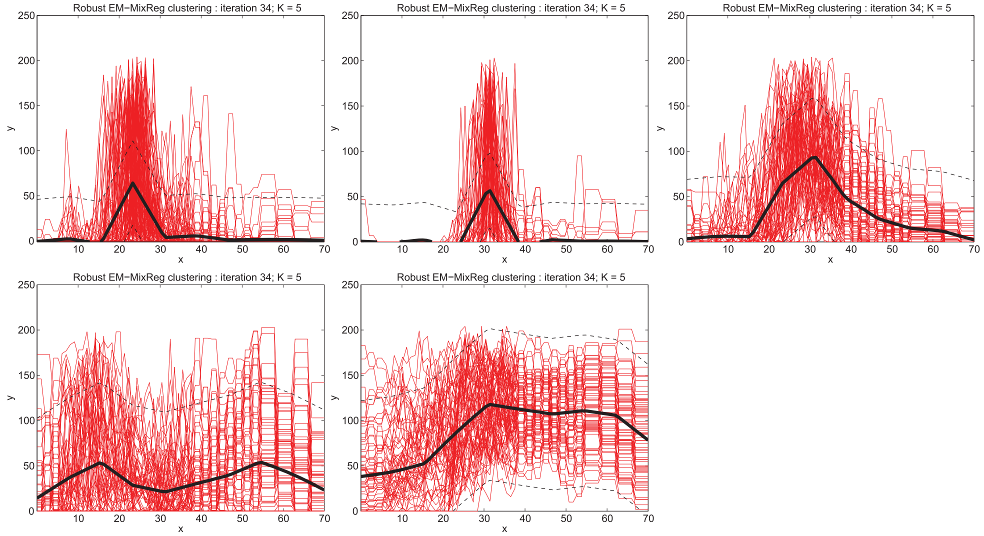


Figure 17. Clustering results obtained by the proposed robust EM-like algorithm and the bSRM model with a linear B-spline of 8 knots for the satellite data. Each sub-figure corresponds to a cluster.

pre-clustering step using the self organizing map (SOM), rather by directly applying the piecewise regression model to the raw data. We also notice that, in the study of Hébrail et al. [9], the number of clusters was set to 20 and the clustering procedure was two-fold. The authors first performed a topographic clustering step using the SOM, and then applied a K-means-like approach to the results of the SOM. However, in our approach, we directly apply the proposed algorithm to the raw satellite data without a preprocessing step. In addition, the number of clusters is automatically inferred from the data. We also can observe that, the found five clusters here do summarize the general behaviour of the 20 clusters in [9] which can be summarized in clusters with one narrow shifted peak, less narrow peak, two large peaks, and finally a cluster containing curves with sharp increase followed by a slow decrease.

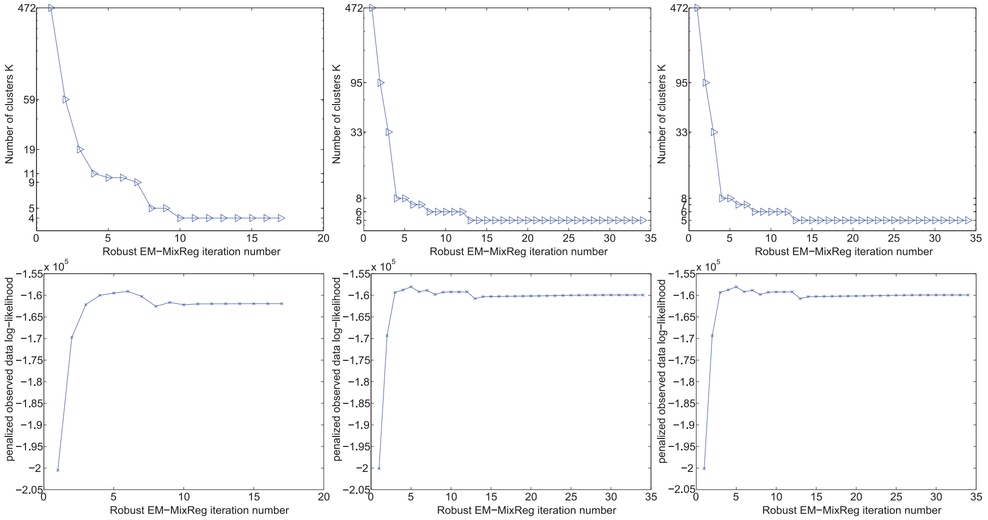


Figure 18. Variation of the number of clusters and the value of the objective function during the iterations of the algorithm for the PRM model (left), the SRM model (middle), and the bSRM model (right) for the satellite data.

Figure 18 shows that the algorithms converge after at most 35 iterations. The variation of the number of clusters during the iterations of the algorithm shows that after starting with $n = 472$ clusters, the number of clusters rapidly decreases to 59 for the PRM and to 95 for both the SRM and the bSRM models. Then it gradually decreases until the number of clusters is stabilized. The variation of the value of the objective function during the iterations of the algorithm also shows that it becomes horizontal at convergence which corresponds to the stabilization of the partition.

5. Conclusions and discussion

We presented a new robust EM-like algorithm for model-based clustering using regression mixtures. It optimizes a penalized observed-data log-likelihood and overcomes both the problem of sensitivity to initialization and determining the optimal number of clusters for standard EM for regression mixtures. We note that the proposed algorithm, as it proceeds to the estimation of the number of components, does not guarantee the ascent property of the objective function, and, thus, is not a true EM algorithm. The experimental results on simulated data and real-world data demonstrate the benefit of the proposed approach for applications in curve clustering. For the PRM, the choice of the polynomial degree can be performed in such a way to obtain the best partition. In practice, we varied the polynomial degree from 3 to 7 for the simulated data and from 4 to 7 for the waveform data. The obtained clustering results are closely similar and the number of clusters was always correctly selected. For the phonemes data and the yeast cell cycle data, the polynomial degree with the best solution was retained. However, for a more general use in functional data clustering and approximation, the splines are clearly more adapted. In practice, for the SRM and bSRM, we used cubic (B-)splines because cubic splines, which correspond to a spline of order 4 which are are sufficient to approximate smooth functions. However, when the data present irregularity, such as a kind of piecewise non-continuous functions, which is the case of the the Topex/Poseidon satellite data, we use a linear (B-)spline approximation. We also note that the algorithm is fast for the three models. It converged after a few number of iterations, and took at most less than 45 seconds for the phonemes data. For the other data, it took only few seconds. This makes it useful for real practical situations.

In this paper, we considered the problem of unsupervised fitting of regression mixtures with unknown number of components. The regression mixture models are similar to the mixture of experts (MEs) model.[54] Although similar, MEs differ from curve clustering models in many respects. One of the main differences is that the ME model consists in a fully conditional mixture while in the regression mixture, only the component densities are conditional. Indeed, the mixing proportions are constant for the regression mixture, while in ME, where they are known as the gating functions, they are modelled as a function of the inputs, generally as a logistic or a softmax function. One interesting future direction is to extend the proposed approach to the problem of fitting MEs [54] and hierarchical MEs [55] with unknown number of experts.

Notes

1. The MATLAB codes are available upon request from the author.
2. Data from http://www.math.univ-toulouse.fr/staph/npfda/
3. The complete data are from http://genome-www.stanford.edu/cellcycle/.
4. The RI measures the similarity between two data clusterings. It has a value between 0 and 1, with 0 indicating that the two partitions do not agree on any pair of observations and 1 indicating that the data clusters are exactly the same. For more details on the RI, see [56].
5. Available at http://www.lsp.ups-tlse.fr/staph/npfda/npfda-datasets.html.

Acknowledgments

I would like to thank the anonymous reviewers for their helpful comments which led to many improvements in the paper.

Disclosure statement

No potential conflict of interest was reported by the author.

References

[1] McLachlan GJ, Peel D. Finite mixture models. New York: Wiley; 2000.
[2] Banfield JD, Raftery AE. Model-based gaussian and non-gaussian clustering. Biometrics. 1993;49(3):803–821.
[3] Fraley C, Raftery AE. How many clusters? Which clustering method? Answers via model-based cluster analysis. Comput J. 1998;41(8):578–588.
[4] Fraley C, Raftery AE. Model-based clustering, discriminant analysis, and density estimation. J Amer Statist Assoc. 2002;97:611–631.
[5] Fraley C, Raftery AE. Bayesian regularization for normal mixture estimation and model-based clustering. J Classif. 2007;24(2):155–181.
[6] McLachlan GJ, Basford KE. Mixture models: inference and applications to clustering. New York: Marcel Dekker; 1988.
[7] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc B. 1977;39(1):1–38.
[8] McLachlan GJ, Krishnan T. The EM algorithm and extensions. 2nd ed. New York: Wiley; 2008.
[9] Hébrail G, Hugueney B, Lechevallier Y, Rossi F. Exploratory analysis of functional data via clustering and optimal segmentation. Neurocomputing. 2010;73(7–9):1125–1141.
[10] Chamroukhi F, Samé A, Govaert G, Aknin P. Time series modeling by a regression approach based on a latent process. Neural Netw. 2009;22(5–6):593–602.
[11] Delaigle A, Hall P, Bathia N. Componentwise classification and clustering of functional data. Biometrika. 2012;99(2):299–313.
[12] Dabo-Niang S, Ferraty F, Vieu P. On the using of modal curves for radar waveforms classification. Comput Statist Data Anal. 2007;51(10):4878–4890.
[13] Ferraty F, Vieu P. Nonparametric functional data analysis: theory and practice. Springer series in statistics. New York: Springer; 2006.
[14] Ramsay JO, Silverman BW. Functional data analysis. Springer series in statistics. New York: Springer; 2005.
[15] Chamroukhi F. Hidden process regression for curve modeling, classification and tracking [PhD thesis]. Compiègne, France: Université de Technologie de Compiègne; 2010.
[16] Gaffney SJ. Probabilistic curve-aligned clustering and prediction with regression mixture models [PhD thesis]. Irvine: Department of Computer Science, University of California; 2004.
[17] Gaffney S, Smyth P. Trajectory clustering with mixtures of regression models. Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining; ACM Press; 1999. p. 63–72.
[18] Ferraty F, Vieu P. Curves discrimination: a nonparametric functional approach. Comput Statist Data Anal. 2003;44(1–2):161–173.
[19] Faria S, Soromenho G. Fitting mixtures of linear regressions. J Stat Comput Simul. 2010;80(2):201–225.
[20] Hunter DR, Young DS. Semiparametric mixtures of regressions. J Nonparametr Stat. 2012;24(1):19–38.
[21] Jones PN, McLachlan GJ. Fitting finite mixture models in a regression context. Aust J Stat. 1992;34(2):233–240.
[22] Quandt RE. A new approach to estimating switching regressions. J Amer Statist Assoc. 1972;67(338):306–310.
[23] Quandt RE, Ramsey JB. Esimating mixtures of normal distributions and switching regressions. J Amer Statist Assoc. 1978;73(364):730–738.
[24] De Veaux RD. Mixtures of linear regressions. Comput Statist Data Anal. 1989;8(3):227–245.
[25] Viele K, Tong B. Modeling with mixtures of linear regressions. Stat Comput. 2002;12:315–330.
[26] Young DS, Hunter DR. Mixtures of regressions with predictor-dependent mixing proportions. Comput Statist Data Anal. 2010;55(10):2253–2266.
[27] Figueiredo MAT, Jain AK. Unsupervised learning of finite mixture models. IEEE Trans Pattern Anal Mach Intell. 2000;24:381–396.
[28] Yang M-S, Lai C-Y, Lin C-Y. A robust em clustering algorithm for gaussian mixture models. Pattern Recognit. 2012;45(11):3950–3961.
[29] DeSarbo WS, Cron WL. A maximum likelihood methodology for clusterwise linear regression. J Classification. 1988;5(2):249–282.
[30] Chamroukhi F, Samé A, Govaert G, Aknin P. A hidden process regression model for functional data description. Application to curve discrimination. Neurocomputing. 2010;73(7–9):1210–1221.
[31] Chamroukhi F, Hervé G, Samé A. Model-based functional mixture discriminant analysis with hidden process regression for curve classification. Neurocomputing. 2013;112:153–163.
[32] Samé A, Chamroukhi F, Govaert G, Aknin P. Model-based clustering and segmentation of time series with changes in regime. Adv Data Anal Classif. 2011;5(4):1–21.
[33] Deboor C. A practical guide to splines. New York: Springer-Verlag; 1978.

[34] Ruppert MP, Wand D, Carroll RJ. Semiparametric regression. Cambridge: Cambridge University Press; 2003.

[35] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. Springer series in statistics. 2nd ed. New York: Springer-Verlag; 2010.

[36] Celeux G, Diebolt J. The SEM algorithm a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. Comput Stat Quart. 1985;2(1):73–82.

[37] Celeux G, Govaert. G. A classification EM algorithm for clustering and two stochastic versions. Comput Statist Data Anal. 1992;14:315–332.

[38] Biernacki C, Celeux G, Govaert G. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models. Comput Statist Data Anal. 2003;41:561–575.

[39] Reddy CK, Chiang H-D, Rajaratnam B. Trust-tech-based expectation maximization for learning finite mixture models. IEEE Trans Pattern Anal Mach Intell. 2008;30(7):1146–1157.

[40] Schwarz G. Estimating the dimension of a model. Ann Statist. 1978;6:461–464.

[41] Akaike H. A new look at the statistical model identification. IEEE Trans Automat Control. 1974;19(6):716–723.

[42] Biernacki C, Celeux G, Govaert G. Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans Pattern Anal Mach Intell. 2000;22(7):719–725.

[43] McLachlan GJ. On bootstrapping the likelihood ratio test stastistic for the number of components in a normal mixture. J Roy Statist Soc Ser C. (Applied Statistics) 1978;36:318–324.

[44] Turner R. Estimating the propagation rate of a viral infection of potato plants via mixtures of regressions. Appl Stat. 2000;49:371–384.

[45] Richardson S, Green PJ. On Bayesian analysis of mixtures with an unknown number of components (with discussion). J R Stat Soc Ser B Stat Methodol. 1997;59(4):731–792.

[46] Chamroukhi F, Samé A, Aknin P, Govaert G. Model-based clustering with hidden Markov model regression for time series with regime changes. Proceedings of the IEEE international joint conference on neural networks (IJCNN); 2011. p. 2814–2821.

[47] Chamroukhi F. Robust EM algorithm for model-based curve clustering. Proceedings of the international joint conference on neural networks (IJCNN), IEEE, Dallas, Texas; August 2013. p. 1–8.

[48] Kooperberg C, Stone CJ. A study of logspline density estimation. Comput Statist Data Anal. 1991;12(3): 327–347.

[49] Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and regression trees. New York: Wadsworth; 1984.

[50] Hastie T, Tibshirani R. Discriminant analysis by gaussian mixtures. J R Stat Soc B. 1996;58:155–176.

[51] Hastie T, Buja A, Tibshirani R. Penalized discriminant analysis. Ann Stat. 1995;23:73–102.

[52] Cho RJ, Campbell MJ, Winzeler EA, et al. A genome-wide transcriptional analysis of the mitotic cell cycle. Mol Cell. 1998;2(1):65–73.

[53] Yeung KY, Fraley C, Murua A, Raftery AE, Ruzzo WL. Model-based clustering and data transformations for gene expression data. Bioinformatics. 2001;17(10):977–987.

[54] Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE. Adaptive mixtures of local experts. Neural Comput. 1991;3(1):79–87.

[55] Jordan MI, Jacobs RA. Hierarchical mixtures of experts and the EM algorithm. Neural Comput. 1994;6:181–214.

[56] Rand WM. Objective criteria for the evaluation of clustering methods. J Amer Statist Assoc. 1971;66(336):846–850.

Appendix 1. Construction of B-splines basis functions

Given the sequence of knots $\xi_0 < \xi_1 < \cdots < \xi_{L+1}$ (ξ_0 and ξ_{L+1} are the two bounds of x), let us define the augmented knot sequence ζ such that

- $\zeta_1 \leq \zeta_2 \ldots \leq \zeta_M \leq \xi_0$;
- $\zeta_{M+\ell} = \xi_\ell,\ \ell = 1, \ldots, L$;
- $\xi_{L+1} \leq \zeta_{L+M+1} \leq \zeta_{KL+M+2} \ldots \leq \zeta_{L+2M}$.

The actual values of these additional knots beyond the boundary are arbitrary, and a common choice is to make them all the same and equal to ξ_0 and ξ_{L+1} , respectively. Let us denote by $B_{\ell,M}(t)$ the ℓth B-spline basis function of order M for the knot-sequence $\zeta_1 \leq \zeta_2 \cdots \leq \zeta_M \leq \xi_0 < \xi_1 < \cdots < \xi_L < \xi_{L+1} \leq \zeta_{L+M+1} \leq \zeta_{L+M+2} \cdots \leq \zeta_{L+2M}$. These basis functions are defined recursively as follows:

- $B_{\ell,1}(x_{ij}) = \mathbb{1}_{[\zeta_j,\zeta_{j+1}]}, \forall \ell = 1, \ldots, L + 2M - 1$;
- $B_{\ell,M}(x_{ij}) = ((x_{ij} - \zeta_\ell)/(\xi_{\ell+M-1} - \zeta_\ell))B_{\ell,M-1}(x_{ij}) + ((\zeta_{\ell+M} - x_{ij})/(\zeta_{\ell+M} - \zeta_{\ell+1}))B_{\ell+1,M-1}(x_{ij}), \quad \forall \ell = 1, \ldots, L + M$.

For the B-spline regression model, the jth row $\mathbf{b}_j$ $(j = 1, \ldots, m_i)$ of the $m_i \times (L + M)$ B-spline regression matrix $\mathbf{B}_i$ for the ith curve is then constructed as follows:

$$\mathbf{b}_j = [B_{1,M}(x_{ij}),\ B_{2,M}(x_{ij}), \ldots, B_{L+M,M}(x_{ij})].$$

Appendix 2. Estimation of the mixing proportions

Consider the problem of finding the maximum of the function (21)

$$Q_\pi(\lambda, \pi_1, \ldots, \pi_K; \boldsymbol{\theta}^{(q)}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}^{(q)} \log \pi_k + \lambda n \sum_{k=1}^{K} \pi_k \log \pi_k \tag{A1}$$

with respect to the mixing proportions $(\pi_1, \ldots, \pi_K)$ subject to the constraint $\sum_{k=1}^{K} \pi_k = 1$. To perform this constrained maximization, we introduce the Lagrange multiplier α and the resulting Lagrangian function is given by:

$$L(\pi_1, \ldots, \pi_K) = \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}^{(q)} \log \pi_k + \lambda n \sum_{k=1}^{K} \pi_k \log \pi_k + \alpha(1 - \sum_{k=1}^{K} \pi_k). \tag{A2}$$

Taking the derivatives of the Lagrangian with respect to π_k for $k = 1, \ldots, K$, we obtain:

$$\frac{\partial L(\pi_1, \ldots, \pi_K)}{\partial \pi_k} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(q)}}{\pi_k} + \left(\lambda \sum_{i=1}^{n} (\log \pi_k + 1)\right) - \alpha. \tag{A3}$$

Then, setting these derivatives to zero yields:

$$\frac{\sum_{i=1}^{n} \tau_{ik}^{(q)}}{\pi_k} + n\lambda \log \pi_k + n\lambda = \alpha. \tag{A4}$$

By multiplying each hand side of Equation (A4) by π_k and summing over k we get

$$\sum_{k=1}^{K} \pi_k \times \left(\frac{\sum_{i=1}^{n} \tau_{ik}^{(q)}}{\pi_k} + n\lambda \log \pi_k + n\lambda\right) = \sum_{k=1}^{K} \alpha \times \pi_k, \tag{A5}$$

which implies that

$$n + n\lambda \sum_{k=1}^{K} \pi_k \log \pi_k + n\lambda = \alpha. \tag{A6}$$

Then, from Equation (A4), it follows that

$$\sum_{i=1}^{n} \tau_{ik}^{(q)} + n\lambda\pi_k \log \pi_k + n\lambda\pi_k = n\pi_k + n\lambda\pi_k \sum_{h=1}^{K} \pi_h \log \pi_h + n\lambda\pi_k \tag{A7}$$

and hence

$$n\pi_k = \sum_{i=1}^{n} \tau_{ik}^{(q)} + n\lambda\pi_k \log \pi_k - n\lambda \sum_{h=1}^{K} \pi_h \log \pi_k, \tag{A8}$$

and we therefore get the updating formula for the mixing proportions π_k's:

$$\pi_k^{(q+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(q)}}{n} + \lambda\pi_k^{(q)} \left(\log \pi_k^{(q)} - \sum_{h=1}^{K} \pi_h^{(q)} \log \pi_h^{(q)}\right) \quad \forall k \in \{1, \ldots, K\}. \tag{A9}$$