Model-based clustering for functional data

Faicel Chamroukhi
chamroukhi.univ-tln.fr

ERCIM 2014, Pisa, Italy

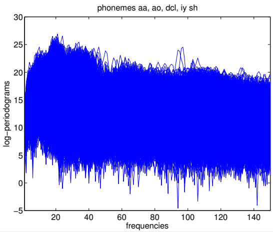
05 December 2014

Outline

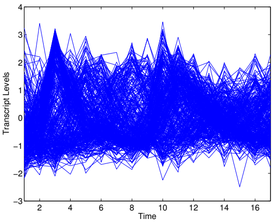
1. Context and objective
2. Model-based curve clustering with regression mixtures
3. Proposed robust EM-like algorithm for model-based curve clustering
4. Experimental study
5. Conclusion and perspectives

Context

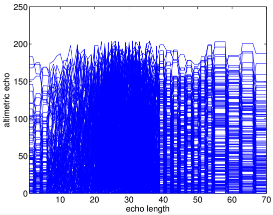
- Curve clustering framework: the data are curves or functions rather than vectors (functional data analysis framework)



Phonemes curves



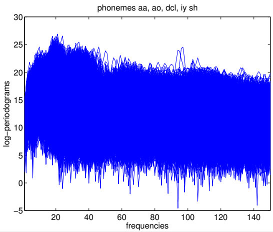
Yeast cell cycle curves



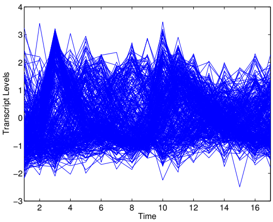
Satellite waveforms

Context

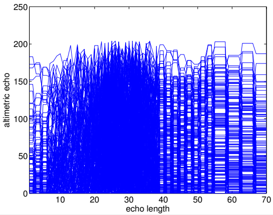
- Curve clustering framework: the data are curves or functions rather than vectors (functional data analysis framework)



Phonemes curves



Yeast cell cycle curves



Satellite waveforms

Objective

- Learn a probabilistic generative model for model-based curve clustering
- Consider the problem of clustering from a fully unsupervised prospective
- ⇒ Deal with the problems of choosing the number of clusters and initialization

Model-based curve clustering

- The aim curve clustering is to cluster n iid unlabeled curves $((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n))$ into K clusters
- We assume that each curve consists of m observations $\mathbf{y}_i = (y_{i1}, \ldots, y_{im})$ regularly observed at the inputs $\mathbf{x}_i = (x_{i1}, \ldots, x_{im})$
- $\Rightarrow$ find the unknown cluster labels $\mathbf{z} = (z_1, \ldots, z_n)$, with $z_i \in \{1, \ldots, K\}$, K being the number of clusters
- $\Rightarrow$ the curve clustering can be performed based on regression mixture models including polynomial regression mixtures (PRM) and polynomial spline regression mixtures (PSRM) (Gaffney, 2004; Chamroukhi, 2010).

Regression mixtures for model-based curve clustering

- The regression mixture models for clustering assume that each curve is drawn from one of K polynomial, spline, or B-spline clusters whose proportions are $(\pi_1, \ldots, \pi_K)$.
- The regression mixture density of the ith curve can be written as:

$$f(\mathbf{y}_i|\mathbf{x}_i;\boldsymbol{\Psi}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{y}_i;\mathbf{X}_i\boldsymbol{\beta}_k, \sigma_k^2 \mathbf{I}_m). \quad (1)$$

- model parameters: $\boldsymbol{\Psi} = (\pi_1, \ldots, \pi_K, \boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_K)$: where $\boldsymbol{\Psi}_k = (\boldsymbol{\beta}_k, \sigma_k^2)$ are respectively the regression coefficients and the noise variance

Regression mixtures for model-based curve clustering

- The regression mixture models for clustering assume that each curve is drawn from one of K polynomial, spline, or B-spline clusters whose proportions are $(\pi_1,\ldots,\pi_K)$.
- The regression mixture density of the ith curve can be written as:

$$f(\mathbf{y}_i|\mathbf{x}_i;\boldsymbol{\Psi}) = \sum_{k=1}^{K}\pi_k\,\mathcal{N}(\mathbf{y}_i;\mathbf{X}_i\boldsymbol{\beta}_k,\sigma_k^2\mathbf{I}_m). \quad (1)$$

- model parameters: $\boldsymbol{\Psi}=(\pi_1,\ldots,\pi_K,\boldsymbol{\Psi}_1,\ldots,\boldsymbol{\Psi}_K)$: where $\boldsymbol{\Psi}_k=(\boldsymbol{\beta}_k,\sigma_k^2)$ are respectively the regression coefficients and the noise variance

- The parameter vector $\boldsymbol{\Psi}$ is estimated by maximizing the log-likelihood:

$$\mathcal{L}(\boldsymbol{\Psi})=\sum_{i=1}^{n}\log\sum_{k=1}^{K}\pi_k\,\mathcal{N}(\mathbf{y}_i;\mathbf{X}_i\boldsymbol{\beta}_k,\sigma_k^2\mathbf{I}_m). \quad (2)$$

- The maximization can be performed iteratively via the EM algorithm (Dempster et al., 1977; McLachlan and Krishnan, 1997; Chamroukhi, 2010).

Limitations

1. The standard EM algorithm for regression mixture model is sensitive to initialization ⇒ requires careful initialization
2. It requires the number of clusters to be supplied by the user ⇒ requires to deal with the model selection

Limitations

1. The standard EM algorithm for regression mixture model is sensitive to initialization ⇒ requires careful initialization
2. It requires the number of clusters to be supplied by the user ⇒ requires to deal with the model selection

In general, theses two issues have been considered each *separately*:

- Initialization techniques: randomly, K-means, CEM, few runs of EM, etc
- Choosing the number of clusters via an afterward model selection procedure: BIC, AIC, ICL, etc

Limitations

1. The standard EM algorithm for regression mixture model is sensitive to initialization ⇒ requires careful initialization
2. It requires the number of clusters to be supplied by the user ⇒ requires to deal with the model selection

In general, theses two issues have been considered each *separately*:

- Initialization techniques: randomly, K-means, CEM, few runs of EM, etc
- Choosing the number of clusters via an afterward model selection procedure: BIC, AIC, ICL, etc

Idea of the proposed approach

- ⇒ Here we attempt to overcome these limitations simultaneously in this case of model-based curve clustering
- ⇒ We propose an EM-like algorithm which is robust with regard initialization and automatically estimate the number of clusters as the learning proceeds
- ⇒A fully unsupervised fitting of regression mixtures with unknown number of components.

Penalized maximum likelihood estimation

- For estimating the regression mixture model ⇒ maximize a penalized log-likelihood function rather than the standard log-likelihood (2)
- penalize the log-likelihood by a term accounting for the model complexity

Penalized maximum likelihood estimation

- For estimating the regression mixture model ⇒ maximize a penalized log-likelihood function rather than the standard log-likelihood (2)
- penalize the log-likelihood by a term accounting for the model complexity

Regularization

- As the model complexity is mainly governed by the number of clusters (the hidden variables z_i) ⇒ use as penalty the entropy of the hidden variable z_i
- The (differential) entropy of one variable ($z_i \in \{1, \ldots, K\}$) is defined by

$$H(z_i) = -\mathbb{E}[\log p(z_i)] = -\sum_{k=1}^{K} p(z_i = k) \log p(z_i = k) = -\sum_{k=1}^{K} \pi_k \log \pi_k \cdot \qquad (3)$$

- The variables $\mathbf{z} = (z_1, \ldots, z_n)$ are i.i.d, ⇒ the whole entropy for $\mathbf{z}$ is:

$$H(\mathbf{z}) \quad = \quad -\sum_{i=1}^{n} \sum_{k=1}^{K} \pi_k \log \pi_k \cdot \qquad (4)$$

Penalized maximum likelihood estimation

- The objective function we propose to maximize is thus given bu the following penalized log-likelihood:

$$\begin{aligned}\mathcal{J}(\lambda,\boldsymbol{\Psi}) &= \mathcal{L}(\boldsymbol{\Psi}) - \lambda H(\mathbf{z}), \quad \lambda \geq 0 \\ &= \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{y}_i;\mathbf{X}_i\boldsymbol{\beta}_k,\sigma_k^2\mathbf{I}_m) + \lambda \sum_{i=1}^{n}\sum_{k=1}^{K} \pi_k \log \pi_k \qquad (5)\end{aligned}$$

- $\mathcal{L}(\boldsymbol{\Psi})$ is the observed-data log-likelihood maximized by the standard EM algorithm for regression mixtures (see Equation (2))
- When the entropy is large, the fitted model is rougher, and when it is small, the fitted model is smoother.
- $\lambda \geq 0$ is a smoothing parameter for establishing a trade-off between closeness of fit to the data and a smooth fit

- the model parameters $\mathbf{\Psi}$ are estimated by maximizing the penalized observed-data log-likelihood (5) $\mathscr{J}(\lambda, \mathbf{\Psi})$ given an i.i.d dataset of n curves $\mathscr{D} = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n))$
- $\mathscr{J}(\lambda, \mathbf{\Psi})$ is iteratively maximized by using a dedicated EM-like algorithm

- the model parameters $\mathbf{\Psi}$ are estimated by maximizing the penalized observed-data log-likelihood (5) $\mathcal{J}(\lambda, \mathbf{\Psi})$ given an i.i.d dataset of n curves $\mathcal{D} = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n))$
- $\mathcal{J}(\lambda, \mathbf{\Psi})$ is iteratively maximized by using a dedicated EM-like algorithm

⇒ The complete-data log-likelihood of $\mathbf{\Psi}$ in this penalized case is given y:

$$\mathcal{J}_c(\lambda, \mathbf{\Psi}) \quad = \quad \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \left[\pi_k \mathcal{N}(\mathbf{y}_i; \mathbf{X}_i \boldsymbol{\beta}_k, \sigma_k^2 \mathbf{I}_m)\right] + \lambda \sum_{i=1}^{n} \sum_{k=1}^{K} \pi_k \log \pi_k \cdot (6)$$

- z_{ik} is an indicator binary variable such that $z_{ik} = 1$ iff $z_i = k$ (i.e., if the ith curve $(\mathbf{x}_i, \mathbf{y}_i)$ is generated by cluster k)

Robust EM-like algorithm for regression mixtures

Start with an initial solution (parameter $\mathbf{\Psi}^{(0)}$ and a number of clusters K)

Robust EM-like algorithm for regression mixtures

Start with an initial solution (parameter $\mathbf{\Psi}^{(0)}$ and a number of clusters K)

1. **E-step** Compute the expected penalized complete-data log-likelihood (6)

$$\begin{aligned} Q(\lambda,\mathbf{\Psi};\mathbf{\Psi}^{(q)}) &= \mathbb{E}\big[\mathscr{J}_c(\lambda,\mathbf{\Psi})|\mathscr{D};\mathbf{\Psi}^{(q)}\big] \qquad (7)\\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(q)}\log\left[\pi_k\mathcal{N}(\mathbf{y}_i;\mathbf{X}_i\boldsymbol{\beta}_k,\sigma_k^2\mathbf{I}_m)\right]+\lambda\sum_{i=1}^{n}\sum_{k=1}^{K}\pi_k\log\pi_k \end{aligned}$$

Robust EM-like algorithm for regression mixtures

Start with an initial solution (parameter $\mathbf{\Psi}^{(0)}$ and a number of clusters K)

1. **E-step** Compute the expected penalized complete-data log-likelihood (6)

$$\begin{aligned} Q(\lambda, \mathbf{\Psi}; \mathbf{\Psi}^{(q)}) &= \mathbb{E}\big[\mathscr{J}_c(\lambda, \mathbf{\Psi})|\mathscr{D}; \mathbf{\Psi}^{(q)}\big] \qquad (7)\\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}^{(q)} \log\left[\pi_k \mathcal{N}(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_k, \sigma_k^2\mathbf{I}_m)\right] + \lambda \sum_{i=1}^{n}\sum_{k=1}^{K} \pi_k \log \pi_k \end{aligned}$$

⇒ simply consists in computing the posterior cluster probabilities:

$$\tau_{ik}^{(q)} = \frac{\pi_k^{(q)} \mathcal{N}(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_k^{(q)}, \sigma_k^{2(q)}\mathbf{I}_m)}{\sum_{h=1}^{K} \pi_h^{(q)} \mathcal{N}(\mathbf{y}_i; \mathbf{X}_i\boldsymbol{\beta}_h^{(q)}, \sigma_h^{2(q)}\mathbf{I}_m)} \cdot \qquad (8)$$

Robust EM-like algorithm for regression mixtures

Start with an initial solution (parameter $\boldsymbol{\Psi}^{(0)}$ and a number of clusters K)

1. **E-step** Compute the expected penalized complete-data log-likelihood (6)

$$\begin{aligned} Q(\lambda,\boldsymbol{\Psi};\boldsymbol{\Psi}^{(q)}) &= \mathbb{E}[\mathscr{J}_c(\lambda,\boldsymbol{\Psi})|\mathscr{D};\boldsymbol{\Psi}^{(q)}] \qquad (7)\\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(q)}\log\left[\pi_k\mathcal{N}(\mathbf{y}_i;\mathbf{X}_i\boldsymbol{\beta}_k,\sigma_k^2\mathbf{I}_m)\right]+\lambda\sum_{i=1}^{n}\sum_{k=1}^{K}\pi_k\log\pi_k \end{aligned}$$

⇒ simply consists in computing the posterior cluster probabilities:

$$\tau_{ik}^{(q)}=\frac{\pi_k^{(q)}\mathcal{N}(\mathbf{y}_i;\mathbf{X}_i\boldsymbol{\beta}_k^{(q)},\sigma_k^{2(q)}\mathbf{I}_m)}{\sum_{h=1}^{K}\pi_h^{(q)}\mathcal{N}(\mathbf{y}_i;\mathbf{X}_i\boldsymbol{\beta}_h^{(q)},\sigma_h^{2(q)}\mathbf{I}_m)}\cdot \qquad (8)$$

2. **M-step** Updating step: $\boldsymbol{\Psi}^{(q+1)}=\arg\max_{\boldsymbol{\Psi}}Q(\lambda,\boldsymbol{\Psi};\boldsymbol{\Psi}^{(q)})$.

1. The mixing proportions updates are obtained by maximizing the function

$$Q_{\pi}(\lambda;\mathbf{\Psi}^{(q)}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(q)}\log\pi_k + \lambda\sum_{i=1}^{n}\sum_{k=1}^{K}\pi_k\log\pi_k$$

⇒ This can be solved using Lagrange multipliers :

$$\pi_k^{(q+1)} = \frac{1}{n}\sum_{i=1}^{n}\tau_{ik}^{(q)} + \lambda\pi_k^{(q)}\left(\log\pi_k^{(q)} - \sum_{h=1}^{K}\pi_h^{(q)}\log\pi_h^{(q)}\right) \qquad (9)$$

1. The mixing proportions updates are obtained by maximizing the function

$$Q_\pi(\lambda;\boldsymbol{\Psi}^{(q)}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(q)}\log\pi_k + \lambda\sum_{i=1}^{n}\sum_{k=1}^{K}\pi_k\log\pi_k$$

⇒ This can be solved using Lagrange multipliers :

$$\pi_k^{(q+1)} = \frac{1}{n}\sum_{i=1}^{n}\tau_{ik}^{(q)} + \lambda\pi_k^{(q)}\left(\log\pi_k^{(q)} - \sum_{h=1}^{K}\pi_h^{(q)}\log\pi_h^{(q)}\right) \quad (9)$$

2. The regression parameters for each class k are updated by maximizing

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

⇒ consists in analytic solutions of K weighted least-squares problems:

$$\boldsymbol{\beta}_k^{(q+1)} = \Big[\sum_{i=1}^{n}\tau_{ik}^{(q)}\mathbf{X}_i^T\mathbf{X}_i\Big]^{-1}\sum_{i=1}^{n}\tau_{ik}^{(q)}\mathbf{X}_i^T\mathbf{y}_i \quad \sigma_k^{2(q+1)} = \frac{\sum_{i=1}^{n}\tau_{ik}^{(q)}\|\mathbf{y}_i-\mathbf{X}_i\boldsymbol{\beta}_k\|^2}{m\sum_{i=1}^{n}\tau_{ik}^{(q)}} \quad (10)$$

- for very small value of λ: the update of the mixing proportions is close to the one in the standard EM update
- however for a large value of λ : the penalization term will play its role in order to make clusters competitive ⇒ allows for discarding illegitimate clusters and enhancing actual clusters This depends on

$$\left(\log\pi_k^{(q)} - \sum_{h=1}^{K}\pi_h^{(q)}\log\pi_h^{(q)}\right) > or < 0$$

- a cluster k can be discarded if its proportion is less than $\frac{1}{n}$, $(\pi_k^{(q)} < \frac{1}{n})$.
- Finally, the penalization coefficient λ is set in an adaptive way to be large for enhancing competition

Initialization and stopping rule

- initialization of the number of clusters : $K^{(0)} = n$
- initialization of the mixing proportions : $\pi_k^{(0)} = \frac{1}{K^{(0)}}$, $(k = 1, \ldots, K^{(0)})$,
- to initialize the regression parameters $\boldsymbol{\beta}_k$ and the noise variances $\sigma_k^{2(0)}$, fit a polynomial regression model to each curve k :

  $$\boldsymbol{\beta}_k^{(0)} = \left(\mathbf{X}^T\mathbf{X}_k\right)^{-1}\mathbf{X}_k\mathbf{y}_k \text{ and } \sigma_k^{2(0)} = \frac{1}{m}\|\mathbf{y}_k - \mathbf{X}_k\boldsymbol{\beta}_k^{(0)}\|^2.$$

  However, to avoid singularities at the starting point, we set $\sigma_k^{2(0)}$ as a middle value in the following sorted range $\|\mathbf{y}_i - \mathbf{X}\boldsymbol{\beta}_k^{(0)}\|^2$ for $i = 1, \ldots, n$.
- $\Rightarrow \boldsymbol{\Psi}_k^{(0)} = (\boldsymbol{\beta}_k^{(0)}, \sigma_k^{2(0)})$.
- The proposed EM algorithm is stopped when the maximum variation of the estimated regression parameters between two iterations $\max_{1\leq k\leq K^{(q)}} \|\boldsymbol{\beta}_k^{(q+1)} - \boldsymbol{\beta}_k^{(q)}\|$ is less than a threshold ϵ (e.g., 10^{-6}).

Choosing the order of regression and spline knots number and locations

- For a general use of the proposed algorithm for the polynomial regression mixture, the order of regression can be chosen by cross-validation techniques as in (Gaffney, 2004).
- In our experiments, we report the results corresponding to the for which th polynomial regression mixture provides the best fit.
- The PRM model may be too simple to capture the full structure of the data, in particular for curves with high non-linearity of with regime changes ⇒ The (B)-spline regression models in this case are more adapted.
- For the (B)-spline models, the most widely used orders are $M = 1{,}2$ and 4 (Hastie et al., 2010).
- For smooth functions approximation, cubic (B)-splines (of order 4) are sufficient to approximate smooth functions.

Choosing the order of regression and spline knots number and locations

- The number of knots and their locations: a common choice is to place a number of knots uniformly spaced across the range of x.
- One can also use automatic techniques for the selection of the number of knots and their locations as reported in (Gaffney, 2004). For example, this can be performed by using cross validation as in (Ruppert and Carroll, 2003).
- 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, we fix the number and location of knots.
- ⇒ We use knot sequences which are uniformly spaced across the range of x.
- The studied problems are not not very sensitive to the number and location of knots (Few number of equispaced knots are sufficient to fit the studied data)

Experimental study

- evaluation of the proposed approach on simulated data and real-world data.
- The algorithms have been implemented on MATLAB[a] and the experiments were performed on a personal laptop.
- We evaluate the proposed EM algorithm for the three regression mixtures models: the polynomial regression mixture, spline regression mixture and B-spline regression mixture, 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).
- We perform experiments on simulated data and the waveforms benchmark of Breiman (Breiman et al., 1984a). Then, we consider three real-world data sets covering three different application area: phonemes, gene expression time course data and radar waveform data.

[a] The MATLAB codes are available upon request from the author.

Evaluation on simulated curves

- Evaluation of the proposed approach on simulated curves.
- Simulated linear and non-linear arbitrary curves (not simulated according to the model)
- See how the performance of the proposed robust EM algorithm with regard to finding the correct number of clusters and the actual partition

Waveform curves of Brieman

- The waveform data consist in a three-class problem where each curve is generated as follows (Breiman et al., 1984b):
  - $\mathbf{y}_1(t) = uh_1(t) + (1-u)h_2(t) + \epsilon_t$ 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.
- 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 second.

Simulated Brieman waveforms

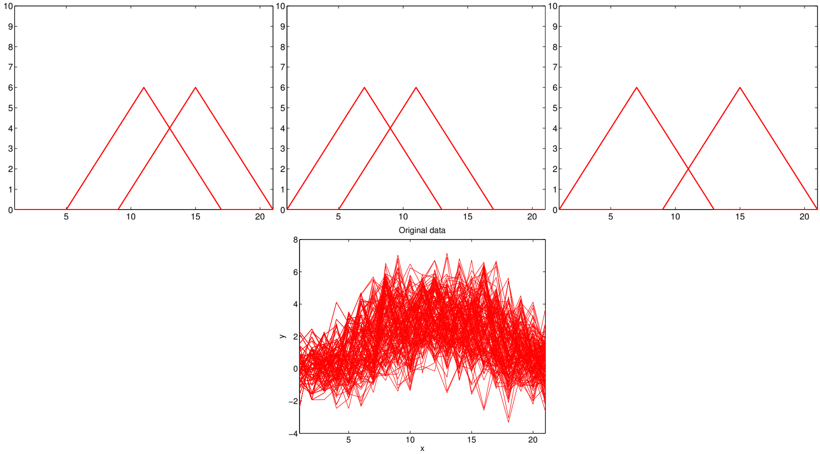


Figure : Waveform mean functions from the generative model before the Gaussian noise is added, and a sample of 150 waveforms.

EM-PRM clustering results

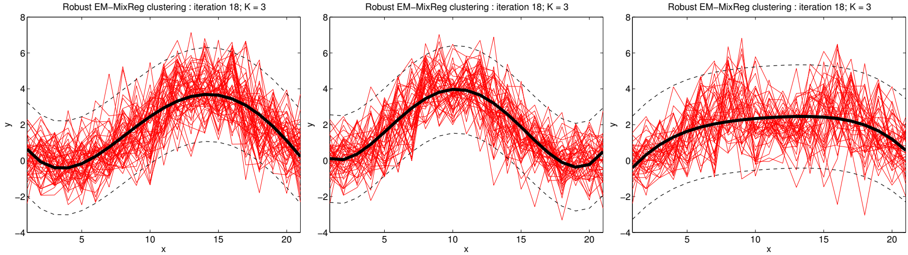


Figure : Clustering results obtained by the proposed robust EM algorithm and the PRM (polynomial degree $p = 4$) model for the waveform data.

EM-PSRM clustering results

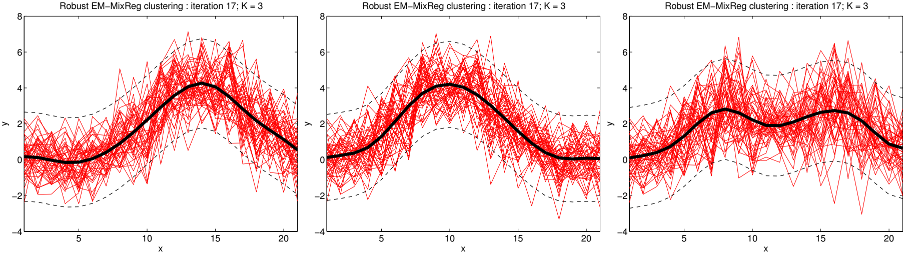


Figure : Clustering results obtained by the proposed robust EM algorithm and the SRM with a cubic-spline of three knots for the waveform data.

EM-PbSRM clustering results

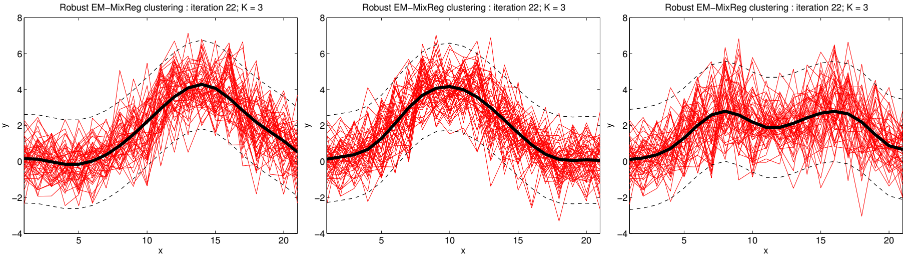


Figure : Clustering results obtained by the proposed robust EM algorithm and the bSRM with a cubic B-spline of three knots for the waveform data.

Clustering results

Estimated number of clusters, misclassification error rate and the absolute error between the true clusters proportions and variances and the estimated ones.

	actual	EM-PRM	EM-SRM	EM-bSRM
K	3	3	3	3
misc. error	-	4.31 ± (0.42)%	2.94 ± (0.88)%	2.53 ± (0.70)%
σ_1	1	0.128 ± (0.015)	0.108 ± (0.015)	0.103 ± (0.012)
σ_2	1	0.102 ± (0.015)	0.090 ± (0.011)	0.079 ± (0.010)
σ_3	1	0.223 ± (0.021)	0.180 ± (0.014)	0.141 ± (0.013)
π_1	$\frac{1}{3}$	0.0037 ± (0.0018)	0.0035 ± (0.0015)	0.0034 ± (0.0015)
π_2	$\frac{1}{3}$	0.0029 ± (0.0023)	0.0018 ± (0.0015)	0.0012 ± (0.0011)
π_3	$\frac{1}{3}$	0.0040 ± (0.0062)	0.0037 ± (0.0015)	0.0035 ± (0.0014)

Table : Clustering results over 20 different samples of 500 curves.

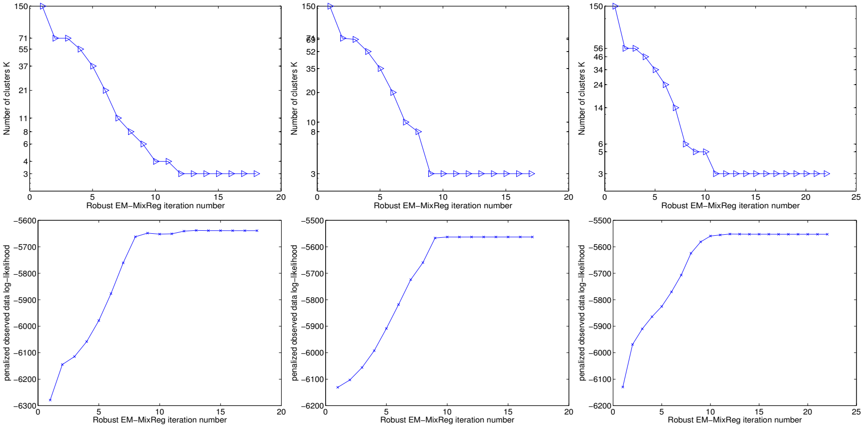


Figure : Variation of the number of clusters and the value of the objective function during the iterations of the algorithm for the PRM (left), SRM (middle) and bSRM (right) for the waveform data.

Simulated Brieman waveforms

- The clustering results are accurate for the three approaches
- The number of clusters is correctly estimated by the three models
- Similar results have also been obtained on mixed linear curves and arbitrary curves
- For this dataset, the spline regression models provide slightly better results in terms of clusters approximation than the polynomial regression mixture

Experiments on real data

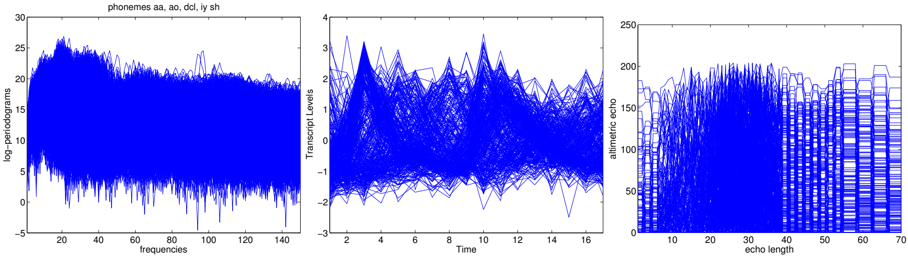


Figure : Real data: Phonemes of the classes ”ao", "aa", "iy", "dcl", "sh" (left), the Yeast cell cycle data (middle) and the Topex/Poseidon satellite data (right).

Phonemes data

1000 log-periodograms (200 per cluster)

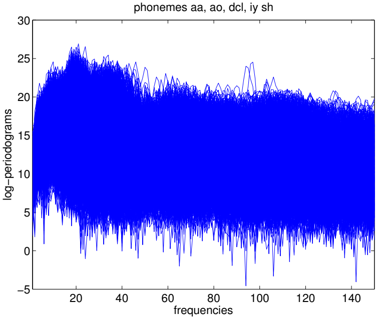


Figure : Phonemes data "ao", "aa", "yi", "dcl", "sh".

Phonemes data

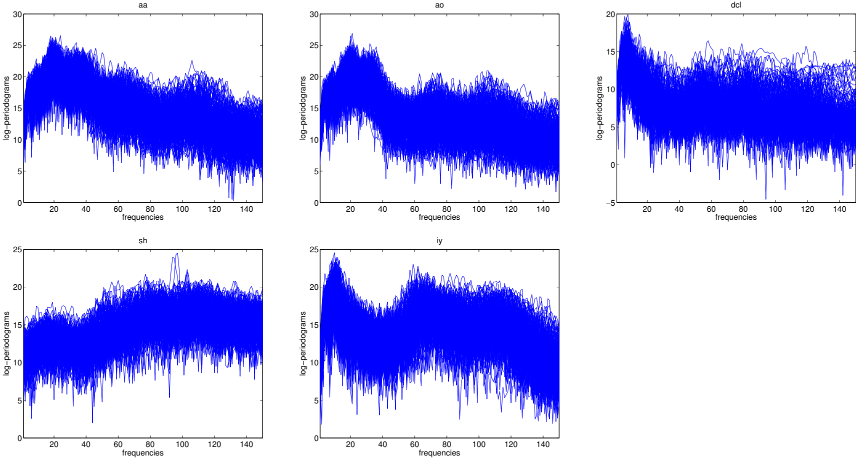


Figure : Curves of the actual five phoneme classes: "ao", "aa", "yi", "dcl", "sh".

PRM clustering results for Phonemes

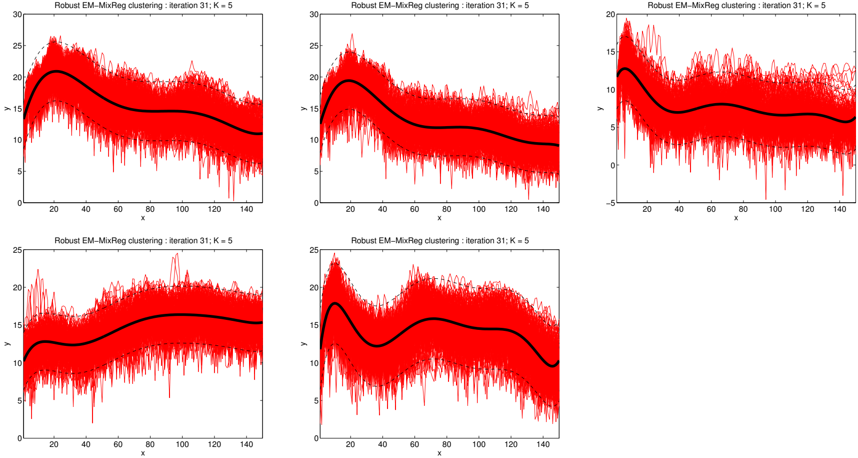


Figure : Clustering results obtained by the proposed robust EM for PRM ($p = 7$)

PbSRM clustering results for Phonemes

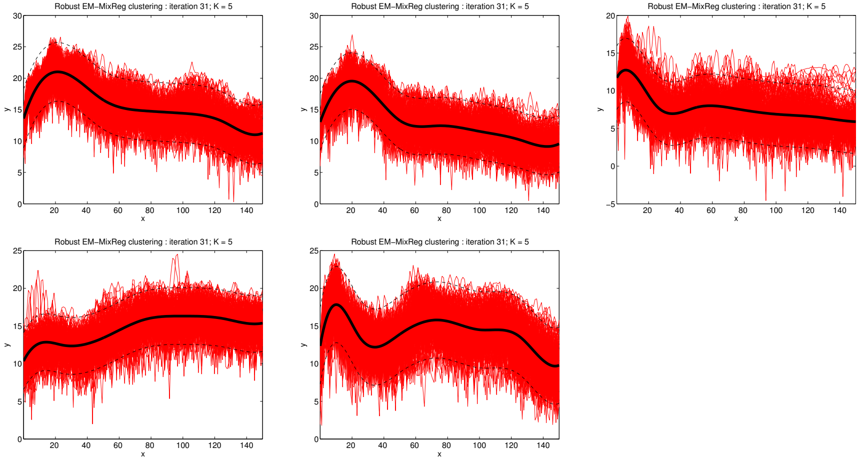


Figure : Clustering results obtained by the proposed robust EM for bSRM

Clustering results for Phonemes

- The spline regression mixture (SRM) results are closely similar to those provided by the B-spline mixture (bSRM)
- The number of phoneme classes is correctly estimated by the three models.
- The spline regression models provide better results in terms of clusters approximation than the polynomial regression mixture (here $p = 7$).
- Notice that the value of $p = 7$ correspond to the polynomial regression mixture model with the best error rate for p varying from 4 to 8.
- Values of the estimated number of clusters and the misc. error rates:

	EM-PRM	EM-SRM	EM-bSRM
Estimated K	5	5	5
Misc. error rate	14.29 %	14.09 %	14.2 %

Table : Clustering results for the phonemes data.

- The spline regressions mixture perform better than the polynomial regression mixture. ⇒ In a general use of functional data modeling, the spline are

Clustering results for Phonemes

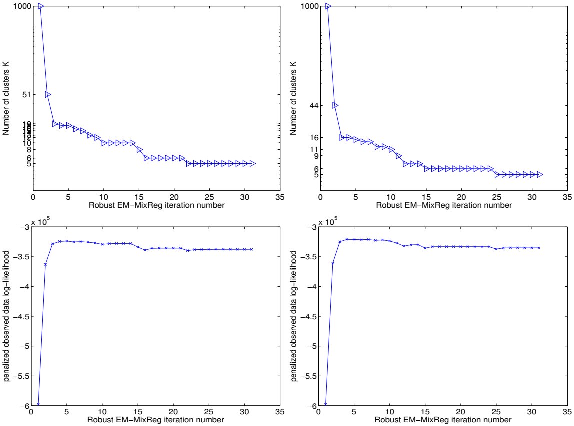


Figure : Variation of the number of clusters and the value of the objective function during the iterations of the algorithm for the PRM (left) and bSRM (right) for the phonemes data.

Clustering results for Phonemes

- The number of clusters decreases very rapidly from 1000 to 51 for the polynomial regression mixture model, and to 44 for the spline and B-spline regression mixture models one iteration to another for the three models.
- The majority of illegitimate clusters is discarded at the beginning of the learning process.
- Then, the number of clusters gradually decreases and the algorithm converge towards a partition with the actual number of clusters for the three models after at most 43 iterations.
- The spline and B-spline regression mixture models behaves in a very similar way.
- We can also notice that the objective function becomes horizontal once the number of clusters is stabilized.

Yeast cell cycle data

- We consider yeast cell cycle data (time course Gene expression data) as in (Yeung et al., 2001) [1]
- This data set referred to as the subset of the 5-phase criterion in (Yeung et al., 2001) contains 384 genes expression levels over 17 time points.
- The utility of the cluster analysis is therefore to reconstruct a class partition.

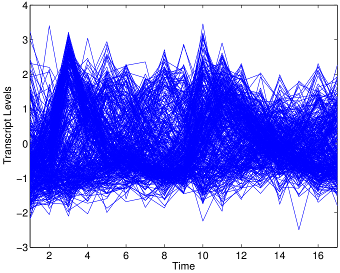


Figure : Yeast cell cycle data.

Yeast cell cycle data

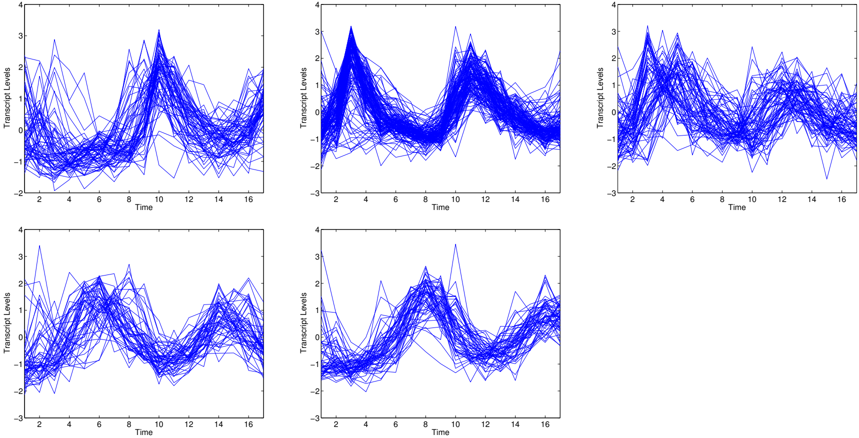


Figure : The five "actual" clusters of the used yeast cell cycle data according to Yeung et al. (2001).

SRM Clustering results for the yeast cell cycle data

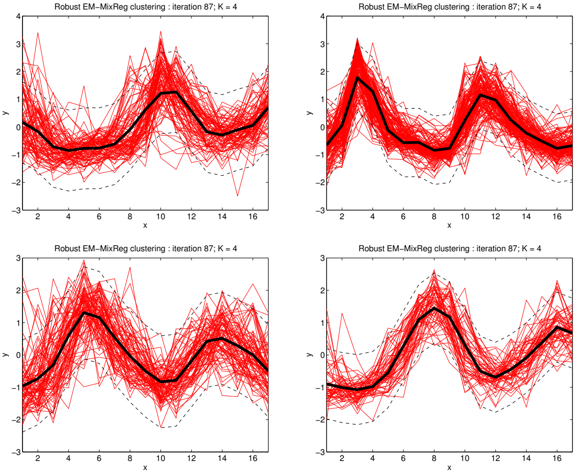


Figure : Clustering results obtained by the proposed robust EM algorithm and the SRM model with a cubic spline of 7 knots for the yeast cell cycle data.

bSRM Clustering results for the yeast cell cycle data

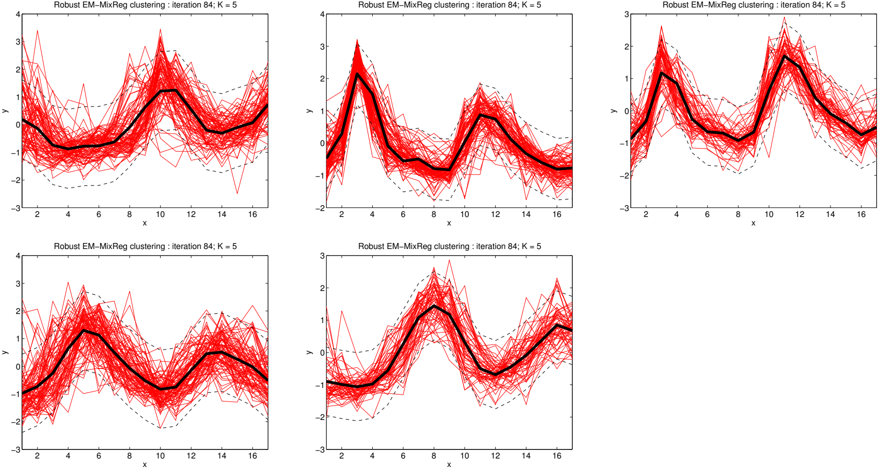


Figure : Clustering results obtained by the proposed robust EM algorithm and the

Clustering results for the yeast cell cycle data

- Both the PRM model and the SRM provide similar partitions with four clusters.
- The second and third clusters for PRM and SRM look to be merged into the second cluster for the bSRM solution and the partition of (Yeung et al., 2001) .
- Note that some model selection criteria in (Yeung et al., 2001) also provide four clusters in some situations.
- the bSRM model infers an accurate partition with the actual number of clusters. The Rand index for the obtained partition is 0.7914 which indicates that the partition is quite well defined.

Clustering results for the yeast cell cycle data

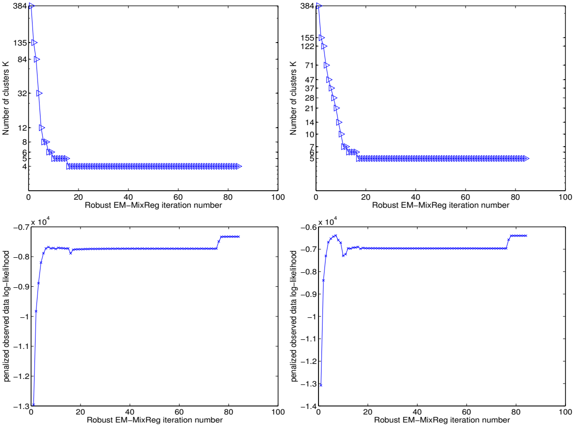


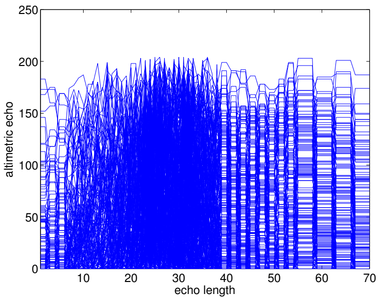
Figure : Variation of the number of clusters and the value of the objective function during the iterations of the algorithm for the PRM (left) and bSRM (right) for the yeast data.

Clustering results for the yeast cell cycle data

- The number of clusters starts with $n = 384$ clusters and more than half is discarded after one iteration. Then it gradually decreases and stabilized until convergence.
- The shape of the objective function also becomes horizontal when it is converged
- Fig. 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 stabilized until convergence. The shape of the objective function also becomes horizontal when it is converged.

Topex/Poseidon satellite data

- This data set were registered by the satellite Topex/Poseidon around an area of 25 kilometers upon the Amazon River (used in (Dabo-Niang et al., 2007))
- The data contain $n = 472$ waveforms of the measured echoes, sampled at $m = 70$ (number of echoes).
- The actual partition is unknown



bSRM clustering results for the satellite data

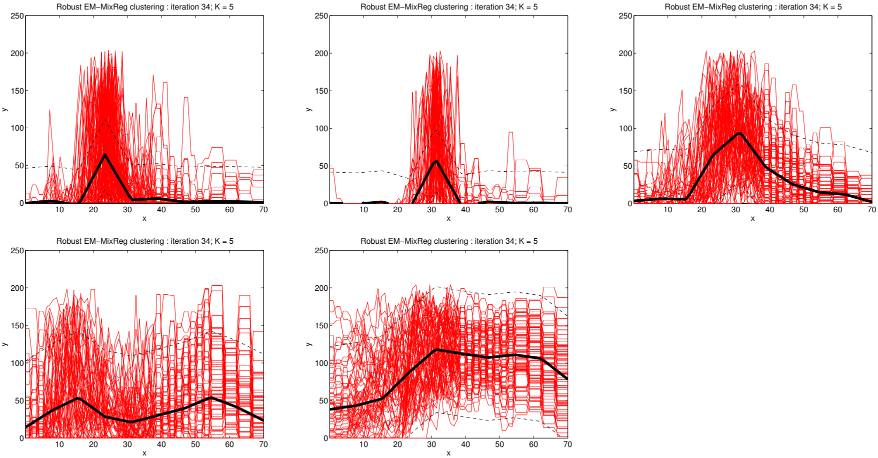


Figure : Clustering results obtained by the proposed robust EM algorithm and the bSRM model with a linear B-spline of 8 knots for the satellite data.

Clustering results for the satellite data

- We used quadratic (B)-spline mixtures (to allow piecewise linear approximation and thus to better recover the possible peaks and transitions).
- The solutions provided by the spline and B-spline regression mixture are very close and are more informative about the underlying structure of this dataset
- both the SRM and the bSRM provide a five class partition with clearly informative clusters: We can indeed see different forms of waves that summarize the general underlying structure governing this dataset.
- We can observe that the first and the second clusters 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. The fourth cluster contains curves that look to have two large peaks. The fifth cluster looks to contain curves without peaks and with a part rather flat.

Clustering results for the satellite data

- Furthermore, we can see that the structure more clear with the cluster mean (prototypes) than with the raw curves. The spline regression mixture models thus helps 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 cluster (five) also equals the one found by (Dabo-Niang et al., 2007) by using another hierarchical nonparametric 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 PRM and the bSRM are similar to those in (Dabo-Niang et al., 2007).

Clustering results for the satellite data

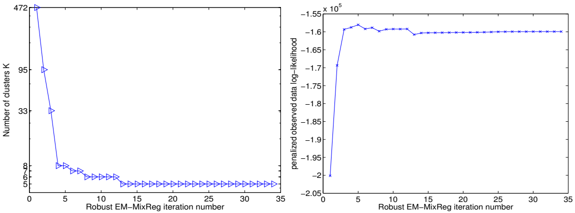


Figure : Variation of the number of clusters and the value of the objective function during the iterations of the algorithm for bSRM for the satellite data.

- the algorithms converge after at most 35 iterations.
- 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 objective function becomes horizontal at converge which correspond to the stabilization of the partition.

Conclusion

- We presented a new robust EM-like algorithm for model-based curve clustering.
- It optimizes a penalized observed-data log-likelihood using the entropy of the hidden structure.
- The proposed algorithm is robust with regard to both the initialization and determining the optimal number of clusters for regression mixtures.
- ⇒ Consists in a fully unsupervised learning technique
- 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 one 45 seconds for the phonemes data. For the other data, it took only few seconds. This makes is useful for real practical situations.
- The experimental results on simulated data and real data demonstrates the potential benefit of the proposed approach for practical use in curve clustering.

Perspectives

- Extend it to the mixed effects regression mixtures (Wang et al., 2012) to deal with the problem of within-cluster variability (current work)
- Consider generative hidden process regression Chamroukhi et al. (2009) and hidden process regression mixtures (Chamroukhi et al., 2010)(Chamroukhi, 2010) (Samé et al., 2011) which also performs curve segmentation.
- One interesting future direction is to extend the proposed approach to the problem of fitting mixture of experts (Jacobs et al., 1991) and hierarchical mixture of experts (Jordan and Jacobs, 1994) with unknown number of experts.

References I

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification And Regression Trees. Wadsworth, New York, 1984a.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification And Regression Trees. Wadsworth, New York, 1984b.

F. Chamroukhi. Hidden process regression for curve modeling, classification and tracking. Ph.D. thesis, Université de Technologie de Compiègne, Compiègne, France, 2010.

F. Chamroukhi. Robust EM algorithm for model-based curve clustering. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, pages 1–8, Dallas, Texas, August 2013.

F. Chamroukhi, A. Samé, G. Govaert, and P. Aknin. Time series modeling by a regression approach based on a latent process. Neural Networks, 22(5-6):593–602, 2009.

F. Chamroukhi, A. Samé, G. Govaert, and P. Aknin. A hidden process regression model for functional data description. application to curve discrimination. Neurocomputing, 73(7-9):1210–1221, March 2010.

Sophie Dabo-Niang, Frédéric Ferraty, and Philippe Vieu. On the using of modal curves for radar waveforms classification. Computational Statistics & Data Analysis, 51(10):4878 – 4890, 2007.

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

S. J. Gaffney. Probabilistic Curve-Aligned Clustering and Prediction with Regression Mixture Models. PhD thesis, Department of Computer Science, University of California, Irvine, 2004.

T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics. Springer, second edition edition, 2010.

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. Neural Computation, 3(1): 79–87, 1991.

M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. Neural Computation, 6:181–214, 1994.

G. J. McLachlan and T. Krishnan. The EM algorithm and extensions. New York: Wiley, 1997.

M.P. Ruppert, D. Wand and R.J. Carroll. Semiparametric Regression. Cambridge University Press, 2003.

References II

A. Samé, F. Chamroukhi, G. Govaert, and P. Aknin. Model-based clustering and segmentation of time series with changes in regime. Advances in Data Analysis and Classification, 5(4):1–21, 2011.

K. Wang, S.K. Ng, and G.J McLachlan. Clustering of time-course gene expression profiles using normal mixture models with autoregressive random effects. BMC Bioinformatics, 13:300, 2012.

Ka Yee Yeung, Chris Fraley, A. Murua, Adrian E. Raftery, and Walter L. Ruzzo. Model-based clustering and data transformations for gene expression data. Bioinformatics, 17(10):977–987, 2001.

Thank you!