Curves clustering with approximation of the density of functional random variables

Julien JACQUES and Cristian PREDA

Université Lille 1, CNRS and INRIA
France

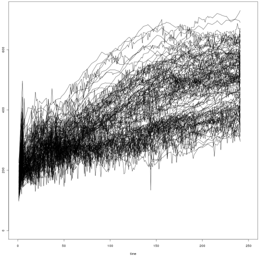
ESANN 2012
20th European Symposium on Artificial Neural Networks Bruges, Belgium, April 25-27, 2012

Introduction

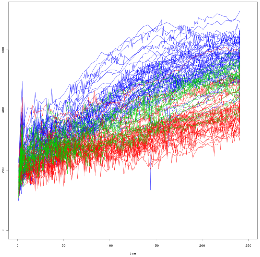
Clustering

Task of assigning a set of objects into groups (*clusters*).

Objects in a cluster are more similar to each other than to those in other clusters.



⇒
clustering



Clustering techniques for functional data

Parametric clustering techniques for curves are generally performed in two steps

- The discretization step aims to describe the functions in a finite dimensional space:
  - direct discretization $(X_{t_1}, \ldots, X_{t_p})$,
  - approximation of curves into a space spanned by a finite basis of functions

$$X(t) = \sum_{j=1}^{J} \alpha_j \Phi_j(t) + \epsilon(t)$$

  - use of on functional principal components (FPCA),
- The clustering step usually applies a multivariate clustering technique on the discretized version of the data:
  - k-means,
  - hierarchical clustering,
  - model-based clustering.

Clustering techniques for functional data

Two steps methods are not satisfactory

- discretization step is done independently on the clustering task,
- how to choose between the discretization techniques and the clustering ones in a unsupervised context ?

Clustering techniques for functional data

Two steps methods are not satisfactory

- discretization step is done independently on the clustering task,
- how to choose between the discretization techniques and the clustering ones in a unsupervised context ?

Funclust

Our model-based clustering method, by approximating the notion of density of a functionnal random variable, performed simultaneously dimension reduction and clustering.

Plan

1. Clustering functional data using density approximation
    - Defining a density approximation
    - The mixture model
    - Model inference

2. Numerical applications on real data

Plan

1. Clustering functional data using density approximation
   - Defining a density approximation
   - The mixture model
   - Model inference

2. Numerical applications on real data

Plan

1. Clustering functional data using density approximation
   - Defining a density approximation
   - The mixture model
   - Model inference
2. Numerical applications on real data

Model-based clustering for functional data

$X = \{X(t),\ t \in [0, T]\}$ a L_2-continuous stochastic process with values in $L_2([0, T])$.

Density probability for functional random variable

Probability density for functions is in general not well defined.

Model-based clustering for functional data

$X = \{X(t),\ t \in [0,T]\}$ a L_2-continuous stochastic process with values in $L_2([0,T])$.

Density probability for functional random variable

Probability density for functions is in general not well defined.

Karhunen-Loeve expansion or *principal component analysis*

$X(t)$ can be decomposed into $X(t) = \mu(t) + \sum_{j=1}^{\infty} C_j \psi_j(t)$ where

- μ is the mean function of X (let assume $\mu \equiv 0$)
- $C_j = \int_0^T X(t)\psi_j(t)dt$, are zero-mean random variables (principal components)
- ψ_j's form an orthonormal system of eigen-functions of the covariance operator of X:

$$\int_0^T Cov(X(t), X(s))\psi_j(s)ds = \lambda_j \psi_j(t), \quad \forall t \in [0,T],\ \lambda_1 \geq \lambda_2 \geq \dots$$

A surrogate for the density

Delaigle & Hall [2011]

For $h > 0$ and $\|.\|$ the usual L_2 norm

$$\log P(\|X - x\| \leq h) = \sum_{j=1}^{q} \log f_{C_j}(c_j(x)) + \xi(h, q(h)) + o(q(h)),$$

where f_{C_j} is the p.d.f. of C_j, and $\xi(h, q(h)) + o(q(h))$ does not depend on x.

A surrogate for the density

Delaigle & Hall [2011]

For $h > 0$ and $\|.\|$ the usual L_2 norm

$$\log P(\|X - x\| \leq h) = \sum_{j=1}^{q} \log f_{C_j}(c_j(x)) + \xi(h, q(h)) + o(q(h)),$$

where f_{C_j} is the p.d.f. of C_j, and $\xi(h, q(h)) + o(q(h))$ does not depend on x.

A surrogate for the density

We deduce that all the variation with x of $\log P(\|X - x\| \leq h)$, up to an approximation at order q, is captured by $\sum_{j=1}^{q} \log f_{C_j}(c_j(x))$ and then

$$P(\|X - x\| \leq h) \simeq \prod_{j=1}^{q} f_{C_j}(c_j(x)) = f_X^{(q)}(x)$$

Plan

1. Clustering functional data using density approximation
   - Defining a density approximation
   - The mixture model
   - Model inference

2. Numerical applications on real data

A model for functional data clustering

Let's go back to clustering

- $\underline{X} = (X_1, ..., X_n)$ be an i.i.d sample of size n of X,

A model for functional data clustering

Let's go back to clustering

- $\underline{X} = (X_1, ..., X_n)$ be an i.i.d sample of size n of X,
- for each X_i, $Z_i = (Z_{i1}, \ldots, Z_{iK}) \in \{0, 1\}^K$ is such that $Z_{ik} = 1$ if X_i belongs to the cluster k,

A model for functional data clustering

Let's go back to clustering

- $\underline{X} = (X_1, ..., X_n)$ be an i.i.d sample of size n of X,
- for each X_i, $Z_i = (Z_{i1}, \ldots, Z_{iK}) \in \{0, 1\}^K$ is such that $Z_{ik} = 1$ if X_i belongs to the cluster k,
- the goal is to predict $\underline{Z} = (Z_1, ..., Z_n)$.

A model for functional data clustering

Let's go back to clustering

- $\underline{X} = (X_1, ..., X_n)$ be an i.i.d sample of size n of X,
- for each X_i, $Z_i = (Z_{i1}, \ldots, Z_{iK}) \in \{0,1\}^K$ is such that $Z_{ik} = 1$ if X_i belongs to the cluster k,
- the goal is to predict $\underline{Z} = (Z_1, ..., Z_n)$.

A mixture model for functional data

We assume that X has the following density approximation,

$$f_X^{(q)}(x;\theta) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{q_k} f_{C_{j|Z_k=1}}(c_{jk}(x);\sigma_{jk}^2)$$

where

- C_j is assumed to be Gaussian (true if X is a Gaussian process)
- $\theta = (\pi_k, \sigma_{1k}^2, \ldots, \sigma_{q_k k}^2)_{1 \leq k \leq K}$ have to be *estimated*,
- $c_{jk}(x)$ have to be *computed*,
- $q = (q_1, \ldots, q_K)$ have to be *selected*.

Plan

1. Clustering functional data using density approximation
   - Defining a density approximation
   - The mixture model
   - Model inference

2. Numerical applications on real data

Model inference

The model parameters can be estimated by maximizing the *approximated likelihood*

An approximated likelihood

$$l^{(q)}(\theta; \underline{X}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \prod_{j=1}^{q_k} \underbrace{\frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp -\frac{1}{2} \left(\frac{C_{ijk}}{\sigma_{jk}} \right)^2}_{f_{C_{j|Z_k=1}}(C_{ijk})}$$

where C_{ijk} is the jth principal score of the curve X_i belonging to the group k.

Model inference

The model parameters can be estimated by maximizing the *approximated likelihood*

An approximated likelihood

$$l^{(q)}(\theta; \underline{X}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \prod_{j=1}^{q_k} \underbrace{\frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp -\frac{1}{2} \left(\frac{C_{ijk}}{\sigma_{jk}} \right)^2}_{f_{C_{j|Z_k=1}}(C_{ijk})}$$

where C_{ijk} is the jth principal score of the curve X_i belonging to the group k.

In the present clustering setting, this approximated likelihood can be maximized by an EM-like algorithm

EM-like algorithm

As for the EM algorithm, we maximize the completed approximated likelihood:

$$L_c^{(q)}(\theta;\underline{X},\underline{Z}) = \sum_{i=1}^{n}\sum_{g=1}^{K} Z_{ik}\left(\log \pi_k + \sum_{j=1}^{q_k} \log f_{C_{j|Z_k=1}}(C_{ijk})\right),$$

The algorithm will iterate the following steps:

E estimate Z_{ik}, i.e. compute $E_{\theta^{(h)}}[Z_{ik}|\underline{X}=\underline{x}]$ (usual E step)

EM-like algorithm

As for the EM algorithm, we maximize the completed approximated likelihood:

$$L_c^{(q)}(\theta; \underline{X}, \underline{Z}) = \sum_{i=1}^{n} \sum_{g=1}^{K} Z_{ik} \left(\log \pi_k + \sum_{j=1}^{q_k} \log f_{C_{j|Z_k=1}}(C_{ijk}) \right),$$

The algorithm will iterate the following steps:

E estimate Z_{ik}, i.e. compute $E_{\theta^{(h)}}[Z_{ik} | \underline{X} = \underline{x}]$ (usual E step)

U update the principal scores C_{ijk} according to $\hat{Z}_{ik}$,

EM-like algorithm

As for the EM algorithm, we maximize the completed approximated likelihood:

$$L_c^{(q)}(\theta; \underline{X}, \underline{Z}) = \sum_{i=1}^{n} \sum_{g=1}^{K} Z_{ik} \left(\log \pi_k + \sum_{j=1}^{q_k} \log f_{C_{j|Z_k=1}}(C_{ijk}) \right),$$

The algorithm will iterate the following steps:

- E estimate Z_{ik}, i.e. compute $E_{\theta^{(h)}}[Z_{ik}|\underline{X} = \underline{x}]$ (usual E step)
- U update the principal scores C_{ijk} according to $\hat{Z}_{ik}$,
- S select the approximation order q_k according to $\hat{Z}_{ik}$,

EM-like algorithm

As for the EM algorithm, we maximize the completed approximated likelihood:

$$L_c^{(q)}(\theta; \underline{X}, \underline{Z}) = \sum_{i=1}^{n} \sum_{g=1}^{K} Z_{ik} \left(\log \pi_k + \sum_{j=1}^{q_k} \log f_{C_{j|Z_k=1}}(C_{ijk}) \right),$$

The algorithm will iterate the following steps:

- E estimate Z_{ik}, i.e. compute $E_{\theta^{(h)}}[Z_{ik}|\underline{X} = \underline{x}]$ (usual E step)
- U update the principal scores C_{ijk} according to $\hat{Z}_{ik}$,
- S select the approximation order q_k according to $\hat{Z}_{ik}$,
- M maximize $E_{\theta^{(h)}}[L_c^{(q)}(\theta; \underline{X}, \underline{Z})|\underline{X} = \underline{x}]$ according to θ (usual M step).

Plan

1. Clustering functional data using density approximation
   - Defining a density approximation
   - The mixture model
   - Model inference

2. Numerical applications on real data

Clustering Kneading data

The data:

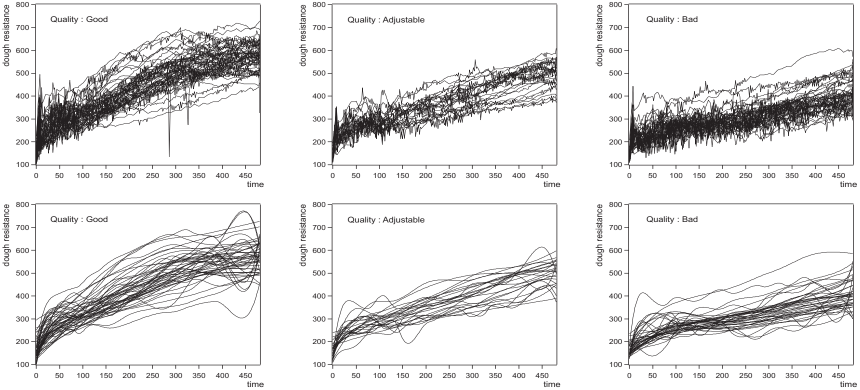


Figure: Kneading data : resistance of dough during the kneading process (480 seconds) for 115 flours. Top : observed data. Bottom : smoothed data using cubic B-splines

Clustering Kneading data

The models under competition:

2-steps methods

- discretization: discretized data, spline coeff., FPCA scores
- clustering: kmeans, hclust, and several parsimonious Gaussian models (HDDC, MixtPPCA, mclust)

Functional methods

- funHDDC (Bouveyron & J. [2011])
- our method funclust

Clustering Kneading data

The results: Percentage of correct classification

2-steps methods	discretized (241 instants)	spline coeff. (20 splines)	FPCA scores (4 components)
HDDC	66.09	53.91	44.35
MixtPPCA	65.22	64.35	62.61
mclust	63.48	50.43	60
kmeans	62.61	62.61	62.61
hclust	63.48	63.48	63.48

functional methods	
fun-HDDC[1]	62.61
funclust	**71.30**

[1] best model selected by BIC

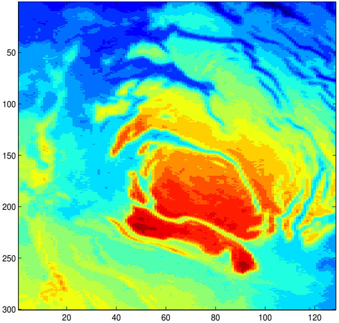
Mars surface characterization

The data

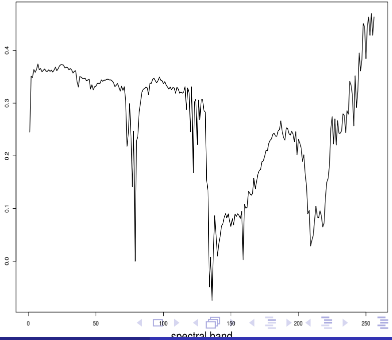
Hyperspectral images (OMEGA instrument, Mars Express spacecraft)

C. Bernard-Michel, S. Douté, M. Fauvel, L. Gardes and S. Girard *Retrieval of Mars surface physical properties frim OMEGA hyperspectral images using regularized sliced inverse regression*, Journal of Geophysical Research, 2009, 114, E06005.

Image 300 × 128



For each pixel



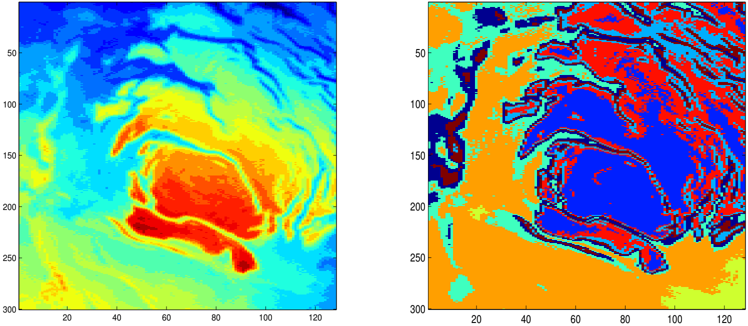
Mars surface characterization

Goal of the study

- Characterization of the surface materials,
- ⇒ clustering of the 38400 pixels,
- number of groups expected by the experts: 8.

Mars surface characterization

Results obtained with funclust



Mars photography and Funclust classification in 8 groups

Conclusion

- a new insight for functional data clustering: Use an approximation of the notion of density *rather* than modeling a discretization of the curves,
- good performance on simulated and real data sets.

Future works

- clustering of multidimensional functional data,
- clustering of qualitative functional data,
- clustering of heterogeneous data (functional data, finite-dimensional quantitative or qualitative data)...

Bibliography

C. Bouveyron, S. Girard and C. Schmid *High-Dimensional Data Clustering*, CSDA, 2007, 52[1], 502–519.

C. Bouveyron and J. Jacques, *Model-based Clustering of Time Series in Group-specific Functional Subspaces*, ADAC, 2011, 5[4], 281-300.

A. Delaigle and P. Hall, *Defining probability density for a distribution of random functions*, Annals of Statistics, 2011, 38, 1171–1193.

J. Jacques and C. Preda, *Model-based clustering of functional data*, Preprint HAL n°00 628247.

J.O. Ramsay and B.W. Silverman, *Functional data analysis*, Springer, New York, 2005.

G.M. James and C.A. Sugar, *Clustering for sparsely sampled functional data*, JASA, 2003, 98[462], 397–408.

EM-like algorithm - details

Details of the 4 step of our EM-like algorithm

EM-like algorithm - E step

Let $\theta^{(h)}$ be the current value of the parameter θ at this step of the algorithm

EM-like algorithm - E step

Compute $E_{\theta^{(h)}}[L_c^{(q)}(\theta; \underline{X}, \underline{Z})|\underline{X} = \underline{x}]$: it consists of computing

$$\begin{aligned} t_{ik} &= E_{\theta^{(h)}}[Z_{ik}|\underline{X} = \underline{x}] = P_{\theta^{(h)}}(Z_{ik} = 1|\underline{X} = \underline{x}) \\ &\simeq \frac{\pi_k^{(h)} \prod_{j=1}^{q_k} f^{(h)}_{C_{j|Z_{ik}=1}}(c_{ijk})}{\sum_{\ell=1}^{K} \pi_\ell^{(h)} \prod_{j=1}^{q_\ell} f^{(h)}_{C_{j|Z_{i\ell}=1}}(c_{ij\ell})}. \end{aligned}$$

where $f^{(h)}_{C_{j|Z_{ik}=1}} = \frac{1}{\sqrt{2\pi}\sigma_{jk}^{(h)}} \exp -\frac{1}{2}\left(\frac{c_{ijk}}{\sigma_{jk}^{(h)}}\right)^2$

EM-like algorithm - U step

Function Principal Component Analysis computation

FPCA eigenfunctions and scores computation needs to assume a basis expansion for the observed curves. Let:

- $\phi = (\phi_1, \ldots, \phi_L)$ be such a basis,
- $\Gamma = (\gamma_{i\ell})_{1 \leq i \leq n, 1 \leq \ell \leq L}$ be the expansion coefficients matrix of $x_1, \ldots, x_n$ in this basis,
- $W = \int \phi\phi'$ denotes the inner product matrix of basis functions.

EM-like algorithm - U step

Function Principal Component Analysis computation

FPCA eigenfunctions and scores computation needs to assume a basis expansion for the observed curves. Let:

- $\phi = (\phi_1, \ldots, \phi_L)$ be such a basis,
- $\Gamma = (\gamma_{i\ell})_{1 \leq i \leq n, 1 \leq \ell \leq L}$ be the expansion coefficients matrix of $x_1, \ldots, x_n$ in this basis,
- $W = \int \phi\phi'$ denotes the inner product matrix of basis functions.

EM-like algorithm - U step

Principal component scores C_{ijk} are updated according to t_{ik}'s:

- curve centering $\Gamma_g = (I_n - 1\!\!1_n(t_{1,g}, \ldots, t_{n,g}))\Gamma$
  I_n and $1\!\!1_n$ are the identity $n \times n$-matrix and the unit n-vector
- jth principal component scores vector C_{jk}: jth eigenvector of $\Gamma_g W \Gamma_g' T_g$ associated to jth eigenvalue λ_{jk}.

EM-like algorithm - S step

EM-like algorithm - S step

We need to select the approximation order q_k for each group.

- likelihood criterion are useless,
- but usual empirical criterion can be used:
  - Cattel scree-test,
  - proportion of the explained variance *(used in the applications)*.

EM-like algorithm - M step

The M step is now usual.
We have to maximize

$$E_{\theta^{(h)}}[L_c^{(q)}(\theta; \underline{X}, \underline{Z}) | \underline{X} = \underline{x}]$$

according to θ.

EM-like algorithm - M step

The maximum are

$$\pi_k^{(h+1)} = \frac{1}{n} \sum_{i=1}^{n} t_{ik}, \quad \text{and} \quad \sigma_{jk}^{2\,(h+1)} = \lambda_{jk}, \quad 1 \leq j \leq q_k$$

where λ_{jk} is already computed at the U step.

Clustering Kneading data

Convergence of the EM like algorithm

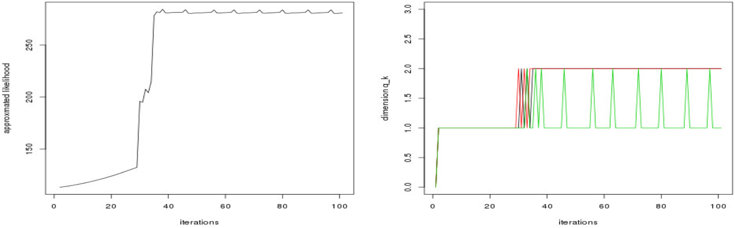


Figure: Left panel: growth of the approximated likelihood (left). Right panel: evolution of the selection of the dimension q_1 (black), q_2 (red) and q_3 (green).