A survey on functional data clustering

Agnès Lagnoux

Institut de Mathématiques de Toulouse
TOULOUSE - FRANCE

Workshop - Inverse calibration of DEB numerical models for Indian Ocean tunas

Victoria, 26-29 May 2015

Outline of the talk

Introduction

Supervised classification

- The context
- Major functional discriminant analysis approaches

Unsupervised classification

- Some preliminaries
- Major functional data clustering approaches
- Model selection
- Software

We follow

1. Jacques, Julien and Preda, Cristian, *Functional data clustering: a survey*, Adv. Data Anal. Classif. (2014);
2. Ferraty, Frédéric and Romain, Yves, *The Oxford Handbook of Functional Data Analysis*, Oxford Handbooks in Mathematics (2010).

Functional data analysis extends the classical multivariate methods. Data are functions or curves such as the evolution of some stock-exchange index, the size of an individual...

In modern statistical terminology, the word “classification” is used with two principal meanings:

- “unsupervised classification” roughly stands for “clustering”,
- “supervised classification” is used as a synonym for the more classical name “discriminant analysis”.

In modern statistical terminology, the word “classification” is used with two principal meanings:

- “unsupervised classification” roughly stands for “clustering”,
- “supervised classification” is used as a synonym for the more classical name “discriminant analysis”.

Both meanings correspond with those given by the Oxford English Dictionary under the entry “classify”:

- first, *arrange a group in classes according to shared characteristics*;
- second, *assign to a particular class or category*.

Unsupervised classification

The basic aim of unsupervised classification techniques is to partition a sample $X_1, \ldots, X_n$ with a large n into a number k of clusters.

These clusters are defined in such a way that the members of each class are “similar” to each other in a sense that is prescribed the clustering algorithm used.

The number k of clusters has to be given in advance or determined simultaneously by the clustering algorithm.

Supervised classification

The supervised classification corresponds to the second meaning of “classify” mentioned above. Here k populations $P_1, \ldots, P_k$ are given and clearly defined in advance.

The practitioner has a training sample $(X_i, Y_i)_{i=1,\ldots,n}$ where the X_i's are independent copies of a random variable X and each Y_i is the index of the corresponding population to which X_i belongs to. The distribution of X is assumed to be different in each population.

The term “supervised” accounts for the fact that the elements of the training sample are supposed to be classified with no error.

The goal is then to assign a new realization of X into one of the populations P_j.

Outline of the talk

Introduction

Supervised classification

- The context
- Major functional discriminant analysis approaches

Unsupervised classification

- Some preliminaries
- Major functional data clustering approaches
- Model selection
- Software

The context

We concentrate on the binary problem: there is only two underlying populations P_0 and P_1.

The available information is given by a training sample $(X_i, Y_i)_{i=1\dots n}$ where X_i is a n-sample of the functional feature $X \in \mathbb{H}$ and

$$Y_i = \begin{cases} 0 & \text{if} \quad X_i \quad \text{belongs to population } 0, \\ 1 & \text{if} \quad X_i \quad \text{belongs to population } 1. \end{cases}$$

The problem is to find a classifier $g : \mathbb{H} \to \{0, 1\}$ that minimizes the criterion error $\mathbb{P}(g(X) \neq Y)$.

The context

An optimal (but unknown) classifier is

$$g^{\star}(x) = \mathbb{1}_{\{\eta(x) \geqslant 1/2\}}$$

where $\eta(x) = \mathbb{E}(Y|X = x)$. In practice, we use the training sample to construct a good classifier:

$$g^{\star}(x) = \operatorname*{Argmin}_{g} \hat{L}_n(g)$$

that minimizes the empirical risk $\hat{L}_n(g) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{g(X_i) \neq Y_i\}}$.

This methodology is more efficient than the one that consists to estimate the regression function η.

Linear discrimination rules

The principle is to generalize the Fisher's methodology of the multivariate case and consists in projecting the infinite dimensional x onto the real line and comparing such a projection with those of the mean functions

$$\mu_j(t) = \mathbb{E}(X(t)|Y = j),\ j = 0, 1.$$

The projection "direction" β would be selected as the maximizer of the distance between the projected class means $\langle \beta, \mu_0 \rangle$ and $\langle \beta, \mu_1 \rangle$ which leads to maximizing

$$\frac{\mathrm{Var}(\mathbb{E}(\langle \beta, X \rangle | Y))}{\mathbb{E}(\mathrm{Var}(\langle \beta, X \rangle | Y))}$$

under the constraint $\int \beta(t) \langle \beta, \mathrm{Cov}(X(t), X(\cdot)|Y = j) \rangle dt = 1$, $j = 0, 1$.

Problem: the covariance operator associated with the kernel $\mathrm{Cov}(X(t), X(s)|Y = j)$ is not in general invertible and thus the above optimization problem has no solution!

$\Rightarrow$ find approximated solutions.

k-NN rules

This technique, adapted in both the multivariate and the functional settings, consists in assigning the data x to the population P_0 as soon as the majority of the k-nearest neighbors of x belongs to P_0. It amounts to replace the unknown regression function $\eta(x) = \mathbb{E}(Y|X = x)$ with the regression estimator

$$\eta_n(x) = \frac{1}{k} \sum_{i=1}^{n} \mathbb{1}_{\{X_i \in k(x)\}} Y_i$$

where "$X_i \in k(x)$" means that X_i is one of the nearest neighbors of x.

In practice the value of k can be chosen by minimizing

$$\tilde{L}_n = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{g_{ni}(X_i) \neq Y_i\}}$$

where g_{ni} denotes the leave-one-out k-NN rule based on the original sample of size n in which the ith observation (X_i, Y_i) has been deleted.

Kernel rules

We use a moving window rule, which is based on the majority vote of the data surrounding the point x to be classified that assigns x to P_0 if

$$\sum_{i=1}^{n} \mathbb{1}_{\{Y_i=0,\ D(X_i,x)\leqslant h\}} > \sum_{i=1}^{n} \mathbb{1}_{\{Y_i=1,\ D(X_i,x)\leqslant h\}}$$

and to P_1 otherwise.

A smoother and more general version is given by

$$g_n(x) = \begin{cases} 0 & \text{if} \quad \sum_{i=1}^{n} \mathbb{1}_{\{Y_i=0\}} K\left(\frac{D(X_i,x)}{h}\right) > \sum_{i=1}^{n} \mathbb{1}_{\{Y_i=1\}} K\left(\frac{D(X_i,x)}{h}\right) \\ 1 & \text{otherwise} \end{cases}$$

where the Kernel is a non increasing known function.

Kernel rules

Some popular choices for K are

- the Gaussian kernel $K(x) = e^{-x^2}$,
- the Epanechnikov kernel $K(x) = (1 - x^2)\mathbb{1}_{[0,1]}(x)$
- or even the uniform kernel $K(x) = \mathbb{1}_{[0,1]}(x)$.

Notice that it amounts to replace the unknown regression function $\eta(x) = \mathbb{E}(Y|X = x)$ with the kernel regression estimator

$$\eta_n(x) = \frac{\sum_{i=1}^{n} Y_i K\left(\frac{D(X_i,x)}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{D(X_i,x)}{h}\right)}$$

Classification based on partial least squares (PLS)

The idea behind PLS is similar to that of principal components: to project the data along directions of high variability. The main difference is that PLS also takes into account the variable response Y when defining the projection directions.

More precisely, let X be a p-dimensional random vector of explanatory variables with covariance matrix Σ_X and Y the real random response with variance σ^2. Denote Σ_{XY} the $p \times 1$ matrix of covariances between X and Y. The first pair of PLS directions are the unit vector a_1 and the scalar b_1 that maximizes with respect to a and b the expression

$$\frac{(\mathrm{Cov}(a^T X, bY))^2}{b^2(a^T a)};$$

in fact, a_1 is the eigenvector of $\Sigma_{XY}\Sigma_{YX}$ corresponding to the largest eigenvalue of this matrix and b_1 fulfills $b_1 = \Sigma_{XY} a_1$.

Classification based on partial least squares (PLS)

The remaining PLS directions can be found in a similar way by imposing the additional condition that the next direction a_{k+1} is orthogonal to the previous ones.

In our particular binary context, the matrix $\Sigma_{XY}\Sigma_{YX}$ can be naturally estimated by

$$S_{XY}S_{YX} = \sum_{i=0}^{1} \frac{1}{(n-1)^2} n_i^2 (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$$

where $\bar{x} = (n_0\bar{x}_0 + n_1\bar{x}_1)/(n_0 + n_1)$ and for $i = 0, 1$, $\bar{x}_i$ denotes the vector of means estimated from the n_i sample elements belonging to the population P_i.

Classification based on reproducing kernels

We use a plug-in classifier obtained by replacing the regression function $\eta(\cdot) = \mathbb{E}(Y|X = \cdot)$ with the function $\hat{\eta} \in \mathcal{H}_k$ minimizing the "regularized empirical risk"

$$\frac{1}{n}\sum_{i=1}^{n} C(X_i, Y_i, \hat{\eta}(X_i)) + J(\hat{\eta})$$

where

- C is a loss function convex to the third argument,
- J is a penalty term.

One could take

$$C(X, Y, \hat{\eta}(X)) = (Y - \hat{\eta}(X))^2 \quad \text{and} \quad J(\hat{\eta}) = \lambda\|\hat{\eta}\|_k^2$$

or even in the binary case the logistic loss function:

$$C(X, Y, \hat{\eta}(X)) = -Y \log \hat{\eta}(X) + \log(1 - 1/(1 - \hat{\eta}(X)).$$

Classification based on depth measures

Assume we have

- a measurement $D(P_i, x)$ of a data x in the population P_i, $i = 0, 1$.
- empirical versions $D_{ni}(x)$ of these depth measures.

For example, in the real line, the depth of a data x with respect to a population P can be

$$F(x)(1 - F(x-)) \quad \text{or even} \quad \min\{F(x), 1 - F(x)\}.$$

A natural classifier would be $g(x) = \mathbb{1}_{\{D_{n1}(x) > D_{n0}(x)\}}$ which amounts to assign x to the population in which we estimate it is most deeply placed.

Other techniques of classification have been developed and rely on support vector machine or even on neural networks.

Outline of the talk

Introduction

Supervised classification

- The context
- Major functional discriminant analysis approaches

Unsupervised classification

- Some preliminaries
- Major functional data clustering approaches
- Model selection
- Software

Cluster analysis deals with the problem of identifying groups, relatively isolated form each other, of similar points.

Unlike supervised classification, there is no training sample to serve as a guide.

In the finite dimensional case, there are two main techniques for cluster analysis: k-means and hierarchical clustering. The first one is an extension of the notion of mean. The second one relies on the use of a matrix of distance between the sample points. This two methods are presented in the sequel.

Classically, as we will explained later, when dealing with functional data, we will adopt two principal strategies that consists in:

1. adapting the k-means technique or the hierarchical clustering one to the functional case choosing dissimilarity distances adapted to this case,
2. or reducing the dimension and using the clustering techniques of the multivariate case.

Functional principal component analysis

From the set of functional data $\{X_1, \ldots, X_n\}$, one is interested in an optimal representation of curves into a function space of reduced (finite) dimension.

Functional principal component analysis (FPCA) has been introduced to address this problem and is widely used in data clustering.

To fix ideas, we assume that X lives in L^2 and moreover that X is centered.

Functional principal component analysis

We define a covariance operator Γ associated with X that traduces the correlations between $X(t)$ and $X(s)$ for any s and t.

The spectral analysis of Γ provides a countable set of positive eigenvalues $(\lambda_j)_{j\geqslant 1}$ associated to an orthonormal basis of eigenfunctions $(f_j)_{j\geqslant 1}$.

Then the Karhunen-Loeve expansion holds:

$$X(t) = \sum_{j\geqslant 1} \langle X, f_j \rangle f_j(t),\ \forall t.$$

Truncating at the first q terms, one obtains the best approximation in norm L_2 of X_t by a sum of quasi-deterministic processes:

$$X(t)^{(q)} = \sum_{j=1}^{q} \langle X, f_j \rangle f_j(t),\ \forall t.$$

k-means in the multivariate case

Very popular for cluster analysis in data mining, *k*-means clustering is a method of vector quantization, originally introduced in signal processing.

Concretely, given a set of observations $(x_1, x_2, \ldots, x_n)$, *k*-means clustering aims to partition the n observations into $k \leqslant n$ sets $S = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares. In other words, its objective is to find:

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mathbf{m}_i\|^2$$

where m_i is the mean of points in S_i.

This results in a partitioning of the data space into Voronoi cells.

k-means in the multivariate case

Given an initial set of k-means $m_1^{(1)}, \ldots, m_k^{(1)}$, the algorithm proceeds by alternating between two steps:

1. **Assignment step**: assign each observation to the cluster whose mean yields the least within-cluster sum of squares. Since the sum of squares is the squared Euclidean distance, this is intuitively the "nearest" mean.

$$S_i^{(t)} = \big\{ x_p : \big\| x_p - m_i^{(t)} \big\|^2 \le \big\| x_p - m_j^{(t)} \big\|^2 \; \forall j, 1 \le j \le k \big\},$$

   where each x_p is assigned to exactly one $S^{(t)}$, even if it could be assigned to two or more of them.

2. **Update step**: calculate the new means to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j.$$

   Since the arithmetic mean is a least-squares estimator, this also minimizes the within-cluster sum of squares objective.

k-means in the multivariate case

The algorithm has converged when the assignments no longer change. Since both steps optimize the objective, and there only exists a finite number of such partitioning, the algorithm must converge to a (local) optimum. There is no guarantee that the global optimum is found using this algorithm.

k-means clustering use cluster centers to model the data, like the expectation-maximization algorithm for mixtures of Gaussian distributions but tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes.

Hierarchical clustering in the multivariate case

In data mining, hierarchical clustering is a method of cluster analysis that builds a hierarchy of clusters, generally according to two techniques:

1. agglomerative which is a “botton up” approach where each observation starts in its own cluster and pairs of clusters are merged as one moves up iteratively the hierarchy.
2. divisive wich is a “top down” approach where all the observations start in the same initial cluster and splits are performed recursively as one moves down the hierarchy.

The results are usually presented in a dendogram.

Hierarchical clustering in the multivariate case

A measure of dissimilarity between sets of observations is required and thus

1. an appropriate metric (a measure of distance between pairs of observations). Usually, one uses the Euclidian distance or its square, the L^1 or L^∞ distances;
2. a linkage criterion that specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets. Some commonly used linkage criteria between two sets of observations A and B are:
   - the maximum or complete linkage clustering: $\max\{d(a,b)/\ a \in A,\ b \in B\}$,
   - the minimum or single-linkage clustering: $\min\{d(a,b)/\ a \in A,\ b \in B\}$,
   - the mean or average linkage clustering: $\frac{1}{|A||B|}\sum_{a\in A,\ b\in B} d(a,b)$,
   - the centroid linkage clustering: $\|c_s - c_t\|$ where c_s and $c - t$ are the centers of clusters s and t respectively.

Two-stages approaches

These approaches consist of

1. a first step in which the dimension of the data is reduced. This reducing dimension step consists generally in approximating the curves into a finite basis of functions: using splines or functional principal component analysis.
2. a second step in which classical clustering tools for finite dimensional data are used: like e.g. *k*-means or hierarchical clustering.

Nonparametric approaches

Nonparametric approaches for functional data clustering are divided into two categories:

1. methods who apply usual nonparametric clustering techniques with specific distances or dissimilarities (k-means or hierarchical clustering for functional data),
2. methods which propose new heuristics or geometry criteria to cluster functional data.

Nonparametric approaches

In the first category of methods, the proximity of two curves x and y is measured through

$$d_l(x,y) = \left(\int_{\mathcal{T}} (x^{(l)}(t) - y^{(l)}(t))^2 dt \right)^{1/2}$$

where $x^{(l)}$ is the l-th derivative of x. Usually the proximity measures d_0, d_1 or even $(d_0^2 + d_1^2)^{1/2}$ are combined with k-means clustering. An other way to perform clustering that have been explored consists in using d_0 or d_2 and hierarchical clustering.

Nonparametric approaches

The second category of nonparametric approaches proposes new heuristics to cluster functional data.

For example, two dynamic programming algorithms perform simultaneously clustering and piecewise estimation of the cluster centers.

A new procedure is developed to identify simultaneously optimal clusters of functions and optimal subspaces for clustering.

Model-based approaches

In *fclust*, the expansion coefficients of the curves into a spline basis of functions are distributed according to a mixture Gaussian distributions with means μ_k, specific to each cluster and common variance Σ:

$$\alpha_i \sim \mathcal{N}(\mu_k, \Sigma).$$

Contrary to the two-stage approaches, in which the basis expansion coefficients are considered fixed, they are considered as random, what allows to proceed efficiently with sparsely sampled curves. Parsimony assumptions on the cluster means μ_k allow to define parsimonious clustering models and low-dimensional graphical representation of the curves.

Model-based approaches

The use of spline basis is convenient when the curves are regular, but are not appropriate for peak-like data as encountered in mass spectrometry for instance. For this reason, a Gaussian model on a wavelet decomposition of the curves has been proposed, which allows to deal with a wider range of functional shapes than splines.

An interesting approach assumes that the curves arise from a mixture of regressions on a basis of polynomial functions, with possible changes in regime at each instant of time.

Choosing the number of clusters

If classical model selection tools, as BIC, AIC or ICL are frequently used in the context of model-based clustering to select the number of clusters, more specific criteria have also been introduced.

First of all, Bayesian model for functional data clustering defines a framework in which the number of clusters can be directly estimated. For instance, a uniform prior over the range $\{1, ..., n\}$ for the number of clusters can be considered, which is then estimated when maximizing the posterior distribution.

Choosing the number of clusters

More empirical criteria have also been used for functional data clustering: for instance, the clustering is repeated several times for each number of clusters and that leading to the highest stability of the partition is retained.

Even more empirical and very sensitive, one way to proceed is to retain the number of clusters leading to a partition having the best physical interpretation.

An original model selection criterion has been considered: this criterion is defined as the averaged Mahalanobis distance between the basis expansion coefficients and their closest cluster center.

Choosing the approximation basis

Almost all clustering algorithms for functional data needs the approximation of the curves into a finite dimensional basis of functions. Therefore, there is a need to choose an appropriate basis and thus, the number of basis functions.

For instance, Fourier basis can be suitable for periodic data, whereas spline basis is the most common choice for non-periodic functional data.

The other solution is to use less subjective criteria such as penalized likelihood criteria BIC, AIC or ICL.

Software

Whereas there exist several software solutions for finite dimensional data clustering, the software devoted to functional data clustering is less developed.

A MATLAB toolbox, *Curve Clustering Toolbox*, implements a family of two-stage clustering algorithms combining mixture of Gaussian models with spline or polynomial basis approximation.

Software

Under the **R** software environment, two-stage methods can be performed using for instance the functions *kmeans* or *hclust* of the *stats* package, combined with the distances available from the *fda* or *fda.usc* packages. Alternatively, several recent model-based clustering algorithms have been implemented by their authors and are available under different forms:

- **R** functions for *funHDDC* and *funclust* are available from request from their authors. An **R** package is available since 2013 on the CRAN website,
- an **R** function for *fclust* is available directly from James's webpage,
- the package *curvclust* for **R** is probably the most finalized tool for curves clustering in **R** and implements the wavelets-based methods.