SAPIENZA
UNIVERSITÀ DI ROMA

Dottorato di Ricerca in Statistica Metodologica

Tesi di Dottorato XXX Ciclo – anno 2016 - 2017

Dipartimento di Statistica, Probabilità e Statistiche Applicate

Classification and Regression Energy Tree for Functional Data

Marco Brandi

Advisor:
Prof. Pierpaolo Brutti
Dipartimento di Statistica, Probabilità e Statistiche Applicate,
Università La Sapienza, Roma

A Paolo e Anna.

Contents

List of Figures

List of Tables

Acknowledgments

Il primo ringraziamento va al mio Supervisore Prof. Pierpaolo Brutti, per il supporto e i consigli durante l'intero percorso dei miei studi.
Un ringraziamento sentito va anche ai Referee Prof. Alessandro Rinaldo e Prof. David Caseur per gli ottimi suggerimenti e per l'apprezzamento mostrato nel mio lavoro.
Vorrei ringraziare anche la Prof.ssa Laureti per tutti i consigli e per tutte le opportunità concesse di lavorare insieme a lei.
Giorgia, Alberto e Sabina meritano senz'altro di essere citati nei ringraziamenti per tutto il supporto morale e pratico che hanno avuto nella costruzione di questa tesi, così come tutti i compagni della stanza 41.
Un ringraziamento speciale ad Alessandra, la quale ha vissuto con me tutto questo periodo pieno di grandi traguardi raggiunti o solo sperati.
Per ultimo ringrazio la mia famiglia senza la quale mai avrei potuto raggiungere tale traguardo.

Abstract

Tree based methods for regression and classification have a long and successful history in statistics and data–analysis [5] and are essentially based on a recursive partition of the covariate space, possibly driven by specific testing procedures design to control branch creation. Starting from the conditional approach introduced in [11] where the choice of the split–variable and the split–value are divided into two different steps allowing an unbiased feature selection, in this work we introduce an energy based testing scheme [27] to validate each of these phases. Energy methods are based on metrics such as distance correlation which, under suitable conditions, ensures the independence of the variables and are therefore more informative than standard association measures. Moreover, as distance correlation measures can be defined for (almost) any kind of variables [18], our proposed framework is flexible enough to accomodate multiple types of covariates. We focus in particular on the case of functional covariates, for which we show simulated and real data examples, as well as comparisons with more established functional data analysis methods.

Introduction

The classification problem is one of the most important challenge for statistician, starting from Fisher in [9], with the research of more complicated models for increasingly complicated data. The complexity goes with the need to create models that can also be understood by less experienced people, because the ultimate goal for classification is to take decisions. For example one can imagine a financial company that must choose whether to grant a certain loan or not and they are interested to know if the applicant can return the money back with interest. They decide to lend the money after the inclusion of certain characteristics in a decision model created by past experiences analyzing some variables called covariates which may be of different types, this is the case of models for credit scoring where an application with neural networks can be find in [33].
There are a wide variety of algorithms used for classification, such as logistic regression model used for binary classification where an application for disease using microarray data can be find in [16]; the natural extension for a multilabel class problem of logistic regression model is the multinomial logistic model. The limit of these models is that are related with the covariates with functional form, in other words are not distribution free. In this case if the relationship assumed in the logistic framework between dependent variables and covariates is not verified, the results could be wrong.
The present dissertation introduces the existing methods for decision tree, where the nature of dependent variable could be categorical or continuous and extend in the case that the covariates are functions. The algorithm, first introduced in [5], called CART (*Classification and Regression Tree*), works top-down starting from the root where we have all observations; then two branches linking the children nodes to the root where the global observation are divided in two dataset and the procedure is iterated until the derived children nodes contain just one observation. This methods suffer of bias variable selection resolved in [11] where a testing procedure using permutation test is proposed. As many existing classification or regression methods are extended when the nature of the variables is functional, in this work a procedure, following the unbiased framework, that allow to use multivariate functional

covariates is proposed. The testing procedure is implemented using energy statistics such as distance correlation [30]. The power of distance correlation test is related to the fact that we can evaluate if there is association between the dependent variable and the covariates, independently to the form of the covariates, because distance correlation and more generally the energy statistics work with distances. In the original work there are proof only for data where we can compute Euclidean distance, in other words just for continuous covariates, but in [18] there is an extension to other metrics such as metric computed in Hilbert space where usually functional data belong.

The present dissertation is organized as follows. In Chapter 1 there is an introduction to the methods that allow us to use functional data in statistical analysis and in Chapter 2 energy statistics and distance correlation tests are introduced. The proposed method is given in Chapter 3 after an introduction to classification tree for non-functional covariates and several application on synthetic and real data are showed for univariate and multivariate functional covariates compared with other existing classification methods. In Chapter 4 an extension to the case that the dependent variable is continuous is showed and it is compared with existing methods for functional regression analysis.

Chapter 1

Functional Data Analysis

1.1 Functional Data Analysis Setup

Functional data analysis (FDA) has been recently developed and formalized, after the publication of the monograph of Ramsey [21]. In this book the authors provide a definition of functional objects and formalize several properties of functional data, such as data representation, descriptive statistics, smoothing techniques and dimensionality reduction.

In the traditional setup we have a sample space $\mathcal{X}$ and a parameter space Θ, and the goal is to make inference on the unknown parameter $\theta \in \Theta$. Classical statistic inference techniques treat data when the sample space is in $\mathbb{R}^d$ with $d \geq 1$ and also the parameter space can be multidimensional.

In FDA, sample space $\mathcal{X}$ belongs to an infinite d-dimensional space. In other words, in FDA, the sample consists in a set of n functions $X_1(t), \ldots, X_n(t)$ defined in a compact subset of the real line, usually $t \in [0, 1]$.

Data are often observed on a discrete grid $t_1, \ldots, t_n$, that can be fine or sparse. It is possible to consider all these values as a multivariate set, and analyze them with standard multivariate techniques, while failing to account for pattern of dependence that could exist between sequential observations. For this reason it is possible to treat this type of data as a functional object which can be used to understand the underlying phenomenon. In a recent review a concise description of functional data analysis is provided in [20]. The functions mentioned above can be viewed as a realizations of a one dimensional stochastic process, assumed to be in a Hilbert space, as $L^2(I)$. In this way, by definition, the process must necessarily satisfy $\mathbb{E}\left[\int_I X^2(t)dt\right] < \infty$. Normally we can not observe some latent or underlying behavior of the process due to the discrete observation of the phenomenon, and

we collect data during time or in fixed grid. Functional data can be observed in a dense or sparse grid and may vary through observations. A general assumption is that data are recorded on the same grid $t_1, \ldots, t_p$ for all n observations. If we use an instrument as for EEG, the grid is equally spaced $t_j - t_{j-1} = t_{j+1} - t_j$. In asymptotic this space tends to zero, and thus p goes to infinity. We have a high dimensionality problem, but we can regularize the functions imposing smoothness on the L^2 process. As it usually occurs in statistics, data can be affected by an error that can be viewed as random noise or measurement error; formally, we observe:

$$y_i = x_i(t) + \epsilon_i \tag{1.1}$$

A typical example of functional data in economics is intra-day stock quotes and continuous measurements of atmospheric monitoring networks in environmental studies.

1.2 Summary statistics for functional data

We hereby extend the concept of classical statistics to the case of functional data. If we want to compute the mean of a functional object, we no longer have a point estimation as the mean is now a function itself

$$\bar{X}(t) = n^{-1}\sum_{i=1}^{n} X_i(t) \tag{1.2}$$

The mean function is obtained as the mean across observations at each grid value. Analogous reasoning applies for the variance function

$$V_X(t) = (n-1)^{-1}\sum_{i=1}^{n}(X_i(t) - \bar{X}(t))^2 \tag{1.3}$$

The standard deviation function is the square root of the variance functions. we can compute the analogues of covariance and correlation in functional framework as follows:

$$COV_X(t_1, t_2) = (n-1)^{-1}\sum_{i=1}^{n}(X_i(t_1) - \bar{X}(t_1))(X_i(t_2) - \bar{X}(t_2)) \tag{1.4}$$

$$CORR_x(t_1, t_2) = \frac{COV_X(t_1, t_2)}{\sqrt{V_X(t_1)V_X(t_2)}} \tag{1.5}$$

1.3 How to manage functional data

We introduced above the concept of functional data and some key summary statistics. Yet, as already mentioned, we usually get to observe data on a discrete grid. How we can represent data as functions? We use a linear combination of bases functions to represent the data; we will consider the two most used: Fourier bases and B-spline bases. Due to the continuous nature of time we cannot observe the phenomenon at all points of T, but we can assume the existence of a function where the data belong. We impose that functions have to be smooth and to validate this property we need to check that functions have one or more derivatives.
A basis function is a set of known functions that we can use to approximate any function taking a finite linear combination of the bases. The Fourier bases are

$$1, \quad \sin(\omega t), \quad \cos(\omega t), \quad \sin(2\omega t), \quad \cos(2\omega t), \quad \sin(3\omega t), \quad \cos(3\omega t), \ldots$$

where $\omega = 2\pi/P$ is a constant that determines the period P of oscillation. We can define now the linear expansion of the function

$$X(t) = \sum_{k=1}^{K} c_k \phi_k(t) = \mathbf{\Phi}(t)\mathbf{c} \tag{1.6}$$

where ϕ_k are the bases functions, c_k are the coefficients and K is the number of bases chosen. Since bases functions are mutually orthogonal, $\mathbf{\Phi}'\mathbf{\Phi}$ is a diagonal matrix. We have a potentially infinite-dimensional object, restricted to K, but we do not consider the truncated expansion as a multivariate object. We obtain a perfect interpolation when $K = n$. If K is accounted for as a parameter, we may estimate it, based on data. Of course large values of K can interpolate better data but sometimes too much in detail, while with lower values of K we can lose information about nearest observations. In the left panel of Fig.1.1 we show a set of 5 Fourier Bases.

Coefficients may be derived using the Fast Fourier Transform that is very efficient when data are equally spaced. Of course if the nature of data is periodic the Fourier bases are the obvious candidate to represent the underlying functions.
The Spline method, on the other hand, was introduced in [7], to partition data based on different points, called knots. These are linked by polynomial segments of order m. As an example, in the right panel of Fig.1.1 we plot a cubic B-spline basis with 4 knots, delimited by the vertical dashed line. These segments are constrained to be smooth at the joins. The first step requires specification of the number of interior knots, that can be fixed or can vary across observations, sometimes it is useful to

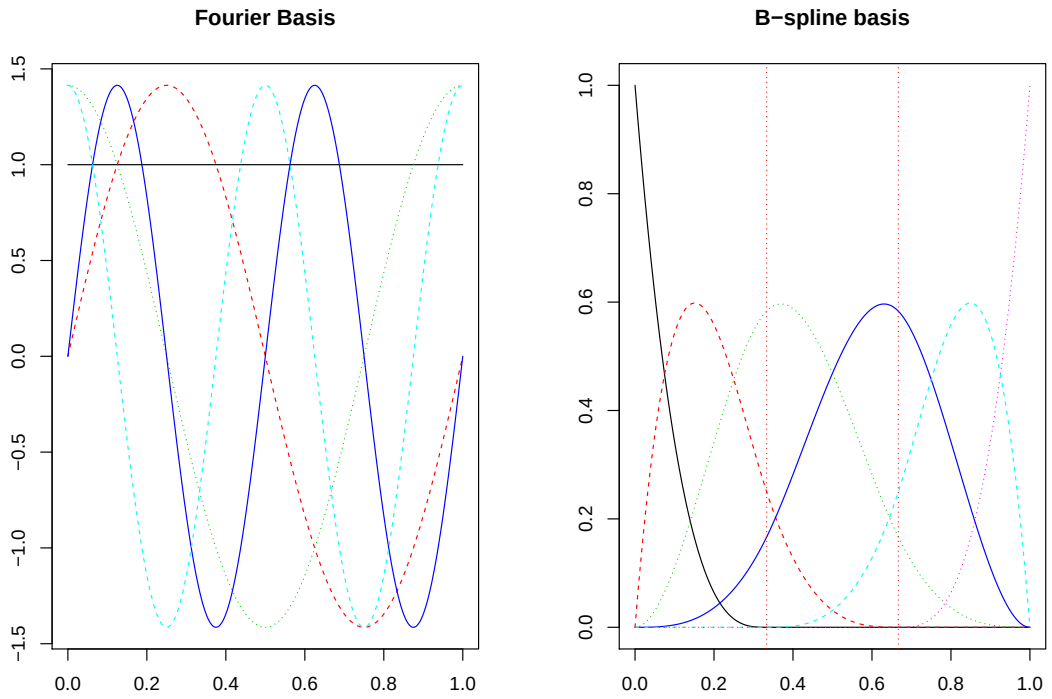

Figure 1.1: In the left panel a Fourier bases with 5 bases, in the right a cubic B-spline with 4 knots

place more knots where there is marked curvature and fewer when the function changes slowly. The basis is defined by the order of the polynomial segments $m+1$ and the number of knots. Normally fewer knots can approximate well functions.
In Fig.1.2 we plot an observation of the growth data from Berkley's growth dataset where there are measure of height for males and females aged from 1 to 18 years. Due to non nonlinear relationship, we could image that a function can be used to synthesize the overall observations. When 4 knots are considered (as in the left panel), the estimated function fails to capture the last part of information at higher values. Such behavior is captured by the cubic spline function if the number of knots is increased to 10 (right panel).

1.3.1 Coefficient Estimation and Choice of Number of Bases

Remind that we want to estimate a function over discrete observed points with relationship

$$y_i = x(t_i) + \epsilon_i \qquad i = 1, \ldots, n$$

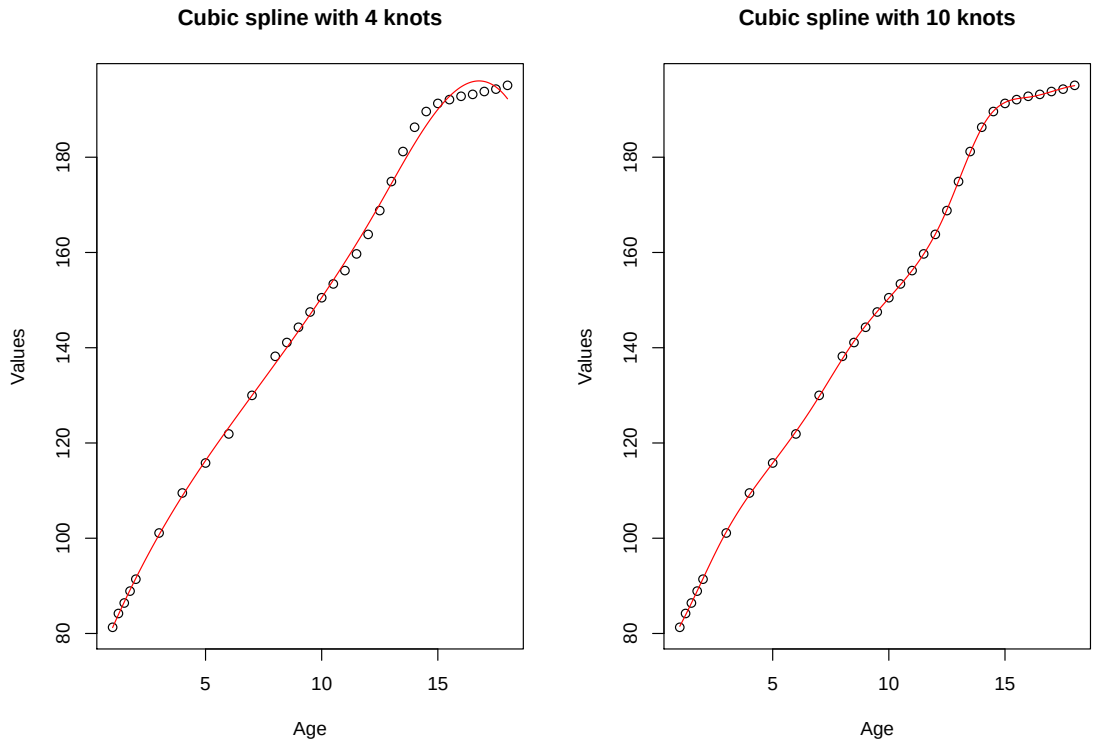

Figure 1.2: Function estimation using B-spline basis for the Barkley's growth data, using different number of knots equally spaced

with

$$x(t) = \sum_{k=1}^{K} c_k \phi_k = \mathbf{c}'\mathbf{\Phi}$$

We can use ordinary least squares to estimate coefficient vector $\mathbf{c}$ minimizing the sum of squares error

$$SSE(\mathbf{y}|\mathbf{c}) = \sum_{i=1}^{n} \left[y_i - \sum_{k=1}^{K} c_k \phi_k(t_i) \right]^2 = (\mathbf{y} - \mathbf{\Phi c})'(\mathbf{y} - \mathbf{\Phi c}) \tag{1.7}$$

Following the standard least squares minimization procedure, we derive this quantity and we equal it to zero, to obtain an estimate for $\mathbf{c}$

$$2\mathbf{\Phi}\mathbf{\Phi}'c - 2\mathbf{\Phi y} = 0$$

The following estimates for the vector of coefficients $\mathbf{c}$ and for the predicted value of $\mathbf{y}$ result:

$$\hat{\mathbf{c}} = (\mathbf{\Phi}'\mathbf{\Phi})^{-1}\mathbf{\Phi y} \qquad \text{and} \qquad \hat{\mathbf{y}} = \mathbf{\Phi}\hat{\mathbf{c}} = \mathbf{\Phi}(\mathbf{\Phi}'\mathbf{\Phi})^{-1}\mathbf{\Phi y} \tag{1.8}$$

This procedure is sound if the components $\epsilon_i, i = 1, \ldots, n$ are normally distributed with zero mean and equal variance σ^2. This is not our case, as the erratic component likely shows autocorrelation and we have to resort to other methods to estimate the coefficients. We extend the method sketched above by adding a symmetric, positive definite matrix $\mathbf{W}$ of weights that can capture different relationship between error terms. We want to minimize

$$SSE(\mathbf{y}|\mathbf{c}) = (\mathbf{y} - \mathbf{\Phi c})'\mathbf{W}(\mathbf{y} - \mathbf{\Phi c}) \tag{1.9}$$

As weighted matrix we can take the variance-covariance matrix of the residual $\mathbf{W} = \mathbf{\Sigma}_\epsilon$, if known, or we can estimate it.

All these methods belong to the general class of linear smoothing methods defined as

$$\hat{x}(t_i) = \sum_{l=1}^{n} S_i(t_l) y_l \tag{1.10}$$

Equivalently specified in matrix form:

$$\hat{x}(\mathbf{t}) = \mathbf{S}\mathbf{y} \tag{1.11}$$

where $\mathbf{S}$ is called the hat matrix, that converts observed y values to fit values $\hat{y}$.

If we choose a large number of basis K we can better fit data, but we can fit also undesired noise in data, i.e. overfitting. Instead a sufficiently small value for K can lead us to miss important feature of smooth function. For this reason we look at the variance-bias trade off. The bias is defined as

$$Bias(\hat{x}(t)) = x(t) - \mathbb{E}\left[\hat{x}(t)\right] \tag{1.12}$$

Usually bias decrease when $K \to n$ and is equal to zero if $K = n$. The variance is defined as

$$V(\hat{x}(t)) = \mathbb{E}\left[\{\hat{x}(t) - \mathbb{E}\left[\hat{x}(t)\right]\}^2\right] \tag{1.13}$$

In contrast to bias, the variance increases if $K \to n$. For a complete evaluation we use the $\mathcal{L}^2$ loss function

$$MSE(\hat{x}(t)) = \mathbb{E}\left[\{\hat{x}(t) - x(t)\}^2\right]$$

that can be decomposed in

$$MSE(\hat{x}(t)) = V(\hat{x}(t)) + Bias(\hat{x}(t))^2 \tag{1.14}$$

The equation in 1.14 is very familiar to statisticians. Usually, when we have unbiased estimator we can work simply on the variance, but in this case we have some bias and that ought to be accounted for. In other words variance and bias are inverse function of the same parameter, bias decrease with model complexity and variance increase otherwise. The best choice of K is obtained when the MSE, Eq.1.14 above combining variance and bias, is minimum. We show this property with Vancouver precipitation data, where we collect the precipitation in mm for all day in Vancouver. We fit data using B-spline bases, taking $x(t)$ as "true" function, then we compute error $\epsilon = y - x(t)$. Then we simulate 1000 records of data, starting from the "true" function randomly rearranging the "true" error . For each simulation we fit the function using Fourier bases. As we can see from Fig.1.3 the bias decreases as K grows, while the variance increases. The obvious best choice for K is at the minimum of the MSE curve (blue line).

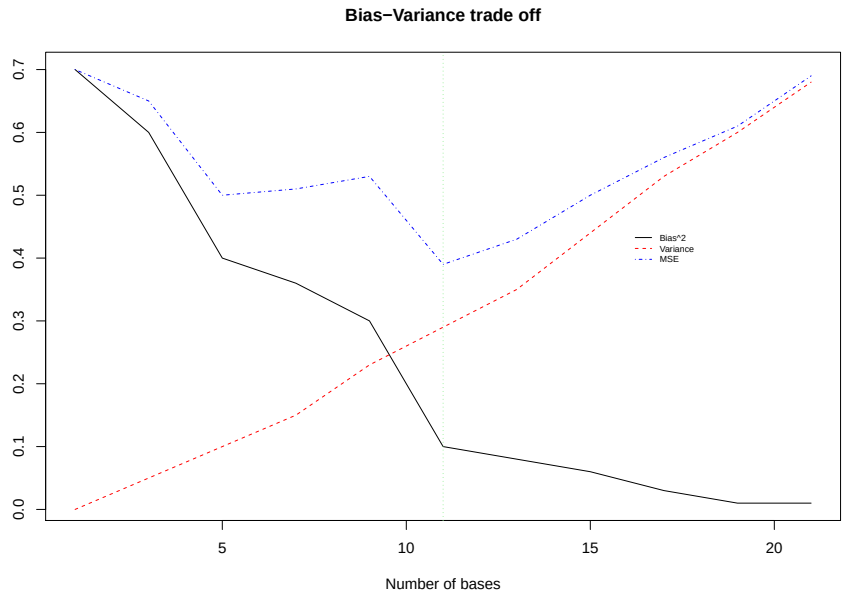

Figure 1.3: Bias-variance trade off in Vancouver precipitation data, compute with 1000 simulation.

1.3.2 Roughness Penalty

In the previous section we illustrated that Fourier bases are useful when the functions to be estimate are periodic, instead B-spline are more flexible and can be used also for non periodic data. Fitting the basis expansion by least squares method that do not allow to a direct control of smoothing.
The method of roughness penalty can approximate discrete data by a function. This

way, we seek to optimize a fitting criterion while accounting for the information on smoothing.
A complete unbiased estimation of $x(t)$ leads to a high variance; we fit perfectly the data but observe a rapid local variation of the curve. As we showed before , MSE bears information about bias and variance. In some cases, adding some bias might significantly reduce sample variance. We quantify roughness using the second derivative $\left[D^2x(t)\right]^2$, that gives us information about the curvature of function at point t. The penalty term is given by the integral of such second derivative

$$PEN_2(x) = \int \left[D^2x(s)\right]^2 ds \tag{1.15}$$

Highly variable functions will have high values of $PEN_2(x)$. As a consequence, piecewise linear curves will be penalized over smooth functions.

We modify the least squares criterion accordingly, , defining the penalized residual sum of squares

$$PENSSE_\lambda(x|\mathbf{y}) = [\mathbf{y} - x(\mathbf{t})]' \mathbf{W} [\mathbf{y} - x(\mathbf{t})] + \lambda PEN_2(x) \tag{1.16}$$

We estimate the function $x(t)$ such that it minimizes the penalized residual sum of squares. The non-negative term λ is called smoothing parameter. For larger value of λ we put more weights on the roughness penalty term and for $\lambda \to \infty$ the fitted curve yields standard linear regression. Conversely, for small values of λ the function is more variable, and for $\lambda \to 0$ the estimated function perfectly interpolates data. By remembering that $x(\mathbf{t}) = \sum_{k=1}^{K} c_k\phi_k(t) = \mathbf{c}'\mathbf{\Phi}(t)$, we can express the penalty term in a matrix form

$$\begin{aligned} PEN_2(x) &= \int \left[D^2x(s)\right]^2 ds \\ &= \int \left[D^2\mathbf{c}'\mathbf{\Phi}(s)\right]^2 ds \\ &= \int \left[\mathbf{c}'D^2\mathbf{\Phi}(s)D^2\mathbf{\Phi}'(s)\mathbf{c}\right] ds \\ &= \mathbf{c}' \left[\int D^2\mathbf{\Phi}(s)D^2\mathbf{\Phi}'(s)ds\right] \mathbf{c} \\ &= \mathbf{c}'\mathbf{R}\mathbf{c} \end{aligned} \tag{1.17}$$

where $\mathbf{R} = \int D^2\mathbf{\Phi}(s)D^2\mathbf{\Phi}'(s)ds$ is the penalty matrix. The penalized least squares criterion becomes

$$PENSSE_2(\mathbf{y}|\mathbf{c}) = (\mathbf{y} - \mathbf{\Phi c})' \mathbf{W} (\mathbf{y} - \mathbf{\Phi c}) + \lambda \mathbf{c}'\mathbf{R}\mathbf{c} \tag{1.18}$$

For the coefficients estimation we need to derive the equation with respect to vector $\mathbf{c}$ in 1.18 and match it to zero. The estimation is

$$\hat{\mathbf{c}} = \left(\mathbf{\Phi'W\Phi} + \lambda\mathbf{R}\right)^{-1}\mathbf{\Phi'Wy} \tag{1.19}$$

The smoothing matrix, that maps data into fit is defined by

$$\mathbf{S}_{\phi,\lambda} = \mathbf{\Phi}\left(\mathbf{\Phi'W\Phi} + \lambda\mathbf{R}\right)^{-1}\mathbf{\Phi'W} \tag{1.20}$$

if $\lambda \to 0$ we have the original least square criterion. We can compute analytically the matrix $\mathbf{R}$ when we use Fourier and B-spline bases, generally, if other bases are involved, we can approximate it with numerical algorithm. We obtain the degree of freedom of a spline smooth taking the trace of the smoothing matrix

$$df(\lambda) = tr(\mathbf{S}_{\phi,\lambda}) \tag{1.21}$$

λ is unknown, and thus we have to estimate it with cross-validation method. The idea is to divide original data in two set. The first, called training sample, where we fit the model and the second, called validation sample. Not considering data we use to fit model can allow us to generalize the model.
The extreme situation where we leave only one observation out and considering as validation sample the $n-1$ observation the model, is called leave-one-out cross validation. For each validation sample we compute the error sum of squares and we sum over all obtained values. We apply this procedure over a grid of values for λ. Two problems can be detected: first, for a big sample, such procedure may be computationally expensive and, as a second point, minimization of the error term resulting from cross-validation might lead to under-smoothing the data because the method sometimes fits the noise or high frequency variation. The generalized cross-validation is less subject to under smooth data and it is expressed

$$\begin{aligned} GCV(\lambda) &= \frac{n^{-1}SSE}{\left[n^{-1}tr(\mathbb{I} - \mathbf{S}_{\phi,\lambda})\right]^2} \\ &= \left(\frac{n}{n - df(\lambda)}\right)\left(\frac{SSE}{n - df(\lambda)}\right) \end{aligned} \tag{1.22}$$

The minimization of GCV with respect to λ will involve a large number of values of λ, using a grid or numerical optimization algorithm.
An example for the method is showed for phoneme data contained in R package *fda.usc* [8]. The dataset contains 250 curves for 5 different phoneme classes. The discrete observations measure the log-periodograms for frequencies in interval $[1 : 150]$. In the left panel of Fig. 1.4 three discrete observations and relative estimated

function are showed. In the right panel the GCV for different values of λ, each curve indicates a different number of bases, cubic B-spline bases are used. The minimum of combination between λ and J is given for values 64 and 27 respectively. Looking at discrete data we could expect large value for λ due the excessive roughness of original signal, in other words the method of roughness penalty tends to penalize this deviations in amplitude. Increasing the number of bases, that in B-spline bases corresponds to increasing the number of interior knots, it gives us a worse result gradually. The reason is that the method tends to estimate the relative noise beyond the signal.

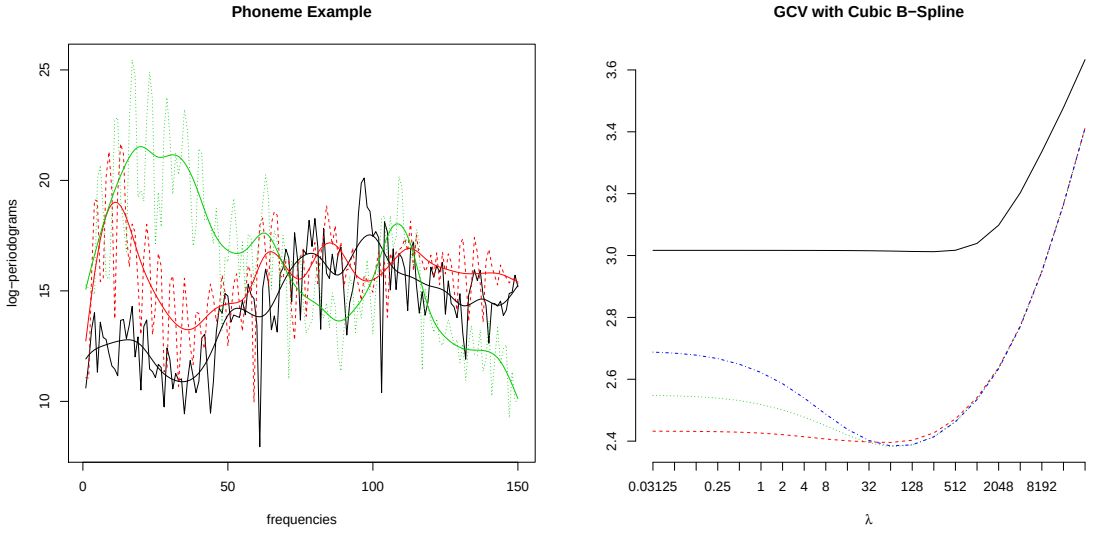

Figure 1.4: Discrete observations and estimated functions for 3 sample members. The generalized Cross-Validation function for different values of λ and for different number of bases. The optimal value for λ is 64 and for J is 27. Cubic B-spline are used for this example.

Chapter 2

Energy Statistics

Energy statistics are defined as functions of distances between statistical observations, based on the notion of Newton's gravitational potential energy which is a function of the distance between bodies. In [30] the authors give us an overview of energy statistics . They show tests based on energy statistics may be regarded as more general and more powerful than classical tests based on quantities such as correlation, F-statistic, etc. The general intuition behind energy statistics is to treat statistical observations as heavenly bodies with a statistical potential energy, which is zero if and only if a statistical null hypothesis is true.

In classical statistics we suppose data are (approximately) normally distributed so it is possible to apply the theory of Gaussian distributions for inference. Also when the data are not normal and n is large we can apply the limit theory and treat the data as normal. When data are not just real numbers but functions or graphs, what is the solution? In this case simple operations, as addition or multiplications, could be a big problem. We can solve it if the observations are elements of a metric space, so we can overcame the difficulties working directly with the (nonnegative) distances. Now we have real numbers and we can make inference on these distances, called *energy inference*.

While normal theory methods such as two-sample-t-test are used to compare means, the energy approach wants to test the equality of distributions and it can detect any difference between them. Working with distances, the energy statistics are invariant to any transformation of the original data, which includes translation, reflection and angle-preserving rotation of coordinate axes. We show also the energy counterparts of variance, covariance and correlation, named distance variance, distance covariance and distance correlation. The distance correlation coefficient will be equal to zero if and only if the variables are independent, in contrast with Pearson's correlation coefficient that sometimes is equal to zero when the variables are dependent (as it

is the case for non-normal data).
For data $\mathbf{X} = X_1, \ldots, X_n$ and $\mathbf{Y} = Y_1 \ldots, Y_n$ in Euclidean spaces, define the distance of $\mathbf{X}$ and $\mathbf{Y}$ as

$$D := 2\sum_{i=1}^{n}\sum_{j=1}^{n}|X_i - Y_j| - \sum_{i=1}^{n}\sum_{j=1}^{n}|X_i - X_j| - \sum_{i=1}^{n}\sum_{j=1}^{n}|Y_i - Y_j| \tag{2.1}$$

D is always nonnegative and it is the square of a metric in the space of samples of size n. For statistical purposes we could work with powers of distances $|X_i - Y_j|^\alpha$, with $0 < \alpha < 2$. The distance matrix now has the form:

$$D_\alpha := 2\sum_{i=1}^{n}\sum_{j=1}^{n}|X_i - Y_j|^\alpha - \sum_{i=1}^{n}\sum_{j=1}^{n}|X_i - X_j|^\alpha - \sum_{i=1}^{n}\sum_{j=1}^{n}|Y_i - Y_j|^\alpha \tag{2.2}$$

The choice of α derived that in this range D_α remains a square of a metric.
We define now the (potential) energy of X with respect Y or otherwise, that take their values in a metric space with a generic distance function δ:

$$\epsilon(X, Y) = 2\mathbb{E}[\delta(X, Y)] - \mathbb{E}[\delta(X, X')] - \mathbb{E}[\delta(Y, Y')] \tag{2.3}$$

where X' and Y' are iid clones of X and Y respectively and $E|X|_d < \infty, E|Y|_d < \infty$
.
The following properties of energy are the baseline of next step:

1. $\epsilon(X, Y) \geq 0$
2. $\epsilon(X, Y) = 0 \iff X \stackrel{d}{=} Y$

All Euclidean spaces, all separable Hilbert space, all Hyperbolic spaces and many graphs with geodesic distances are metric spaces where these properties are verified [18]. A necessary and sufficient condition for property 1 is the conditional negative definiteness of δ.

For a d-dimensional sample $X_1, \ldots, X_n$ and for a symmetric kernel function of Euclidean distances between sample elements $h : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, energy statistics are U-statistics or V-statistics based on distances:

$$U_n = \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}h(X_i, X_j) \tag{2.4}$$

or

$$V_n = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}h(X_i, X_j) \tag{2.5}$$

An example of U-statistics for dispersion is Gini's mean difference

$$\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}|X_i - X_j| \tag{2.6}$$

The widely used distance is the L_2-distance. Let F is the cumulative distribution function (cdf) and F_n the relative empirical cdf, the L_2-distance is defined as

$$\int_{-\infty}^{\infty}(F_n(x) - F(x))^2 dx \tag{2.7}$$

This distance is not distribution-free and critical values depend on F. This problem can be solved substituting dx by $dF(x)$ in order to obtain

$$\int_{-\infty}^{\infty}(F_n(x) - F(x)^2 dF(x) \tag{2.8}$$

When $d > 1$ the main problem is that this distance is not rotation invariant; this could be a problem if we want to test multivariate normality.

Denote the characteristic function of the probability density function f and g by $\hat{f}$ and $\hat{g}$. The Fourier transform of the cumulative distribution function $F(x) = \int_{-\infty}^{x} f(u)du$ is $\hat{f}(t)/(it)$, where $i = \sqrt{-1}$. Now we can write the distance between the cdf of X and Y as

$$2\pi\int_{-\infty}^{\infty}(F(x) - G(x))^2 dx = \int_{-\infty}^{\infty}\frac{|\hat{f}(t) - \hat{g}(t)|^2}{t^2}dt$$

Then the energy distance can be defined

$$2E|X - Y|_d - E|X - X'|_d - E|Y - Y'|_d = \frac{1}{c_d}\int_{\mathbb{R}^d}\frac{|\hat{f}(t) - \hat{g}(t)|^2}{|t|_d^{1+d}}dt$$

where $c_d = \frac{\pi^{(1+d)/2}}{\Gamma(\frac{1+d}{2})}$
The energy distance $\epsilon(X, Y) \geq 0$ and exactly equal to zero if and only if when X and Y are identically distributed.

2.1 Testing for equal distributions

In [27] the authors introduce a new method to test the equality of distributions in a two-sample context. The existing methods have been explored in the classical literature, with a major focus on the univariate case. The main test for comparing distributions in the univariate case are Kolmogorov-Smirnov and Cramér-von Mises tests, which are not distribution free for the extension in the multivariate framework.

Other distribution-free tests were proposed in the past using the nearest neighbors in the Euclidean distance metric. In two-sample context we have two independent samples $\mathbf{X} = X_1, \ldots, X_{n_1}$ and $\mathbf{Y} = Y_1 \ldots, Y_{n_2}$ such that each observation belongs to $\mathbb{R}^d$ with $d \geq 1$ with different size $n_1 \neq n_2$ and we want to test

$$H_0 : F_1 = F_2 \tag{2.9}$$

where F_1 and F_2 are the cdfs of $\mathbf{X}$ and $\mathbf{Y}$ respectively. The obvious alternative is $F_1 \neq F_2$. The natural extension for a k-sample test is

$$H_0 : F_1 = \cdots = F_k \tag{2.10}$$

The alternatives hypothesis are $F_i \neq F_j$ at least one pair, $1 \leq i \leq j \leq k$.
The Eq. (2.3) serves as the base for the following procedure within the two-sample framework using the Euclidean distance as δ. Let $\mu_{XY} = \mathbb{E}||X - Y||$, $\mu_X = \mathbb{E}||X - X'||$, $\mu_Y = \mathbb{E}||Y - Y'||$, then the expected value of the energy distance is

$$\begin{aligned} \mathbb{E}[\epsilon(\mathbf{X}, \mathbf{Y})] &= \frac{n_1 n_2}{n_1 + n_2} \left(2\mu_{XY} - \frac{n_1 - 1}{n_1}\mu_X - \frac{n_2 - 1}{n_2}\mu_Y \right) \\ &= \frac{n_1 n_2}{n_1 + n_2} (2\mu_{XY} - \mu_X - \mu_Y) + \frac{n_2 \mu_X}{n_1 + n_2} + \frac{n_1 \mu_Y}{n_1 + n_2} \end{aligned}$$

If $X \overset{d}{=} Y$ all mean values are equal, implying quantity $2\mu_{XY} - \mu_X - \mu_Y$ is equal to zero and

$$\mathbb{E}[\epsilon(\mathbf{X}, \mathbf{Y})] = \frac{n_2 \mu_X + n_1 \mu_Y}{n_1 + n_2} = \mu_{XY} = \mathbb{E}||X - Y||$$

If X and Y are not identically distributed it follows from Property 2 that the quantity $2\mu_{XY} - \mu_X - \mu_Y = c \geq 0$. In other words, let $n = n_1 + n_2$, the expected value of the energy distance $\mathbb{E}[\epsilon(\mathbf{X}, \mathbf{Y})]$ is asymptotically a constant proportional to n. If n tends to infinity not only the expected value, but also $\epsilon(\mathbf{X}, \mathbf{Y})$ converges in distribution to infinity under the null hypothesis. Hence large values of $\epsilon(\mathbf{X}, \mathbf{Y})$ correspond to different distribution for X and Y. Next we formalize the test statistics that allows us to accept or refuse the equality in distribution. Consider the same two-sample framework with different sample sizes n_1 and n_2, the test statistics is

$$\begin{aligned} \epsilon_{n_1, n_2} &= \frac{n_1 n_2}{n_1 + n_2} \Big(\frac{2}{n_1 + n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} ||X_i - Y_j|| - \\ &- \frac{1}{n_1^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} ||X_i - X_j|| - \frac{1}{n_2^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} ||Y_i - Y_j|| \Big) \end{aligned}$$

To obtain a p-value from this test we need to know the joint distribution of random variables X and Y to derive the distribution of the test statistics, that is usually unknown. We describe the permutation method that allow us to obtain a p-value without the knowledge of the distribution of the test statistics in 2.3.

2.2 Distance Correlation

In this section we provide the definition of distance covariance and distance correlation that measure the dependence between random vector $X \in \mathbf{R}^p$ and $Y \in \mathbf{R}^q$ with finite first moments in arbitrary dimension. The distance covariance is defined as a measure of the distance between the joint characteristic function and the product of the characteristic function of X and Y, as the square root of

$$\mathcal{V}^2(X,Y) = ||\hat{f}_{X,Y}(t,s) - \hat{f}_X(t)\hat{f}_Y(s)||^2 \tag{2.11}$$

By definition of the norm $||\cdot||$, it is clear that $\mathcal{V}^2(X,Y) \geq 0$ and $\mathcal{V}^2(X,Y) = 0$ if and only if X and Y are independent. Distance variance is defined as the square root of

$$\mathcal{V}^2(X,X) = ||\hat{f}_{X,X}(t,s) - \hat{f}_X(t)\hat{f}_X(s)||^2 \tag{2.12}$$

We can define the distance correlation (dCor) between random vectors X and Y as the nonnegative number $\mathcal{R}(X,Y)$

$$\mathcal{R}^2(X,Y) = \begin{cases} \frac{\mathcal{V}^2(X,Y)}{\sqrt{\mathcal{V}^2(X)\mathcal{V}^2(Y)}} & \mathcal{V}^2(X)\mathcal{V}^2(Y) > 0 \\ 0 & \mathcal{V}^2(X)\mathcal{V}^2(Y) = 0 \end{cases} \tag{2.13}$$

Let $\hat{f}^n_X(t)$, $\hat{f}^n_Y(s)$, $\hat{f}^n_{,Y}(t,s)$ th empirical characteristic functions, we can consider the distance between these quantity as an estimation of distance covariance. An important result is given in [31] where we can relate sample characteristic functions to distances, then

$$||\hat{f}^n_{X,Y}(t,s) - \hat{f}^n_X(t)\hat{f}^n_Y(s)||^2 = S_1 + S_2 - 2S_3 \tag{2.14}$$

where

$$S_1 = \frac{1}{n^2}\sum_{k,l=1}^{n} |X_k - X_l|_p |Y_k - Y_l|_q \tag{2.15}$$

$$S_2 = \frac{1}{n^2}\sum_{k,l=1}^{n} |X_k - X_l|_p \frac{1}{n^2}\sum_{k,l=1}^{n} |Y_k - Y_l|_q \tag{2.16}$$

$$S_3 = \frac{1}{n^3}\sum_{k=1}^{n}\sum_{l,m=1}^{n} |X_k - X_l|_p |Y_k - Y_m|_q \tag{2.17}$$

First we compute all the pairwise distances between sample observations of X, following we compute all the pairwise distances between the observation in Y. We compute Euclidean distance for X sample

$$a_{kl} = |X_k - X_l|, \text{ for } k,l = 1,\ldots,n \tag{2.18}$$

Similarly we compute distances for sample Y

$$b_{kl} = |Y_k - Y_l|, \text{ for } k, l = 1, \ldots, n \tag{2.19}$$

We can compute row means, column means and global mean for each sample as

$$\bar{a}_k = \frac{1}{n}\sum_{i=1}^{n} a_{ki}, \qquad \bar{a} = \frac{1}{n^2}\sum_{k,l=1}^{n} a_{kl} \text{ for } k, l = 1, \ldots, n \tag{2.20}$$

In the same way we obtain $\bar{b}_l$ and $\bar{b}$ for the Y sample. Then we can define the centered matrix as

$$A_{kl} = a_{kl} - \bar{a}_k - \bar{a}_l + \bar{a} \qquad B_{kl} = b_{kl} - \bar{b}_k - \bar{b}_l + \bar{b} \tag{2.21}$$

We obtain the centered distance matrix A_{kl} and B_{kl} such that their row and column sum to zero. The sample distance covariance is given by the square root of

$$\mathcal{V}_n^2(X, Y) = \frac{1}{n^2}\sum_{k,l=1}^{n} A_{kl}B_{kl} \tag{2.22}$$

The sample distance correlation is defined as

$$\mathcal{R}_n^2(X, Y) = \begin{cases} \frac{\mathcal{V}_n^2(X,Y)}{\sqrt{\mathcal{V}_n^2(X)\mathcal{V}_n^2(Y)}} & \mathcal{V}_n^2(X)\mathcal{V}_n^2(Y) > 0 \\ 0 & \mathcal{V}_n^2(X)\mathcal{V}_n^2(Y) = 0 \end{cases} \tag{2.23}$$

In [31] is shown that we have almost sure convergence:

$$\lim_{n\to\infty} \mathcal{V}_n^2(X, Y) = \mathcal{V}^2(X, Y) \tag{2.24}$$

$$\lim_{n\to\infty} \mathcal{R}_n^2(X, Y) = \mathcal{R}^2(X, Y) \tag{2.25}$$

Under dependence of (X,Y) the statistic $n\mathcal{V}_n^2(X, Y) \to \infty$ as $n \to \infty$, so a test that rejects independence for a large value of this statistic is consistent against dependent alternatives.

In [29] the authors show that for high dimension data the sample distance correlation coefficient tends to 1 also in the case that X and Y are independent, in other words

$$\mathcal{R}_n^2(X, Y) \longrightarrow_{p,q,\to\infty} 1 \tag{2.26}$$

They proposed a modification such that under independence where a transformation of the distance correlation converges, when $p, q \to \infty$, to a T-student distribution. The authors re-arranging the centered distance matrix in the following way:

$$A_{kl}^* = \begin{cases} \frac{n}{n-1}\left(A_{kl} - \frac{a_{kl}}{n}\right) & k \neq l \\ \frac{n}{n-1}\left(\bar{a}_i - \bar{a}\right) & k = l \end{cases} \qquad B_{kl}^* = \begin{cases} \frac{n}{n-1}\left(B_{kl} - \frac{b_{kl}}{n}\right) & k \neq l \\ \frac{n}{n-1}\left(\bar{b}_i - \bar{b}\right) & k = l \end{cases} \tag{2.27}$$

The modified distance covariance is

$$\mathcal{V}_n^*(X,Y) = \frac{\mathcal{U}_n^*(X,Y)}{n(n-3)} = \frac{1}{n(n-3)}\left[\sum_{k,l=1} A_{kl}^* B_{kl}^* - \frac{n}{n-2}\sum_{k=1}^{n} A_{kk}^* B_{kk}^*\right] \tag{2.28}$$

and $E\left[\mathcal{U}_n^*(X,Y)\right] = n(n-3)\mathcal{V}(X,Y)$ so $\mathcal{V}_n^*(X,Y)$ is an estimator of the squared distance covariance. The modified distance correlation is defined as

$$\mathcal{R}_n^*(X,Y) = \frac{\mathcal{V}_n^*(X,Y)}{\sqrt{\mathcal{V}_n^*(X)\mathcal{V}_n^*(Y)}} \tag{2.29}$$

and if p, q tend to infinity, under the independence hypothesis the quantity

$$\mathcal{T} = \sqrt{\nu-1}\frac{\mathcal{R}_n^*}{\sqrt{1-(\mathcal{R}_n^*)^2}} \tag{2.30}$$

converges in distribution to T-student with $\nu-1$ degrees of freedom where $\nu = \frac{n(n-3)}{2}$. Asymptotically the quantity $\sqrt{\nu-1}\mathcal{R}_n^*$ is distributed as a standard normal.

2.3 An Extension of Analysis of Variance

In [28] the authors proposed the nonparametric extension using distance correlation of the analysis of variance. For a K independent samples $A_1, \ldots, A_K$ with cumulative distribution $F_1, \ldots, F_K$ and size $n_1, \ldots, n_K$ respectively, the hypothesis for equal distribution is:

$$H_0 : F_1 = \ldots = F_K \tag{2.31}$$

and the alternative hypothesis is that there exist one $F_j \neq F_k$ for some $1 \leq j < k \leq K$. The test statistic proposed for testing equality of distributions is

$$F_\alpha = \frac{S_\alpha(K-1)}{W_\alpha(N-K)} \tag{2.32}$$

where $N = \sum_{k=1}^{K} n_k$, S_α is the between-sample measure of dispersion and W_α is the within-sample measure of dispersion defined as

$$S_\alpha(A_1, \ldots, A_K) = \sum_{1\leq j<k\leq K}\left(\frac{n_j+n_k}{N}\right) d_\alpha(A_j, A_k) \tag{2.33}$$

and

$$W_\alpha = \sum_{j=1}^{K}\frac{n_j}{2} g_\alpha(A_j, A_j) \tag{2.34}$$

g_α is a distance computed between two objects (in our case will be functions) and

$$d_\alpha(A_j, A_k) = \frac{n_j n_k}{n_j+n_k}\left[2g_\alpha(A_j, A_k) - g_\alpha(A_j, A_j) - g_\alpha(A_k, A_k)\right] \tag{2.35}$$

The distribution of test statistic defined in 2.32 cannot be derived analytically, except in particular cases, and support the alternatives hypothesis for large value of F_α. We can compute the p-value for this specific test via permutation implementation that allow us to find the result for the test without the knowledge of the exact distribution of F_α.
Define $\nu = 1, \ldots, N$ the label associated to each observations and let $\pi(\nu)$ denote a permutation of the elements of ν. Under the null hypothesis the statistics computed for every permutation π of ν are identical distributed to the observed test statistic. It is not necessary to take all the permutations but just a small number R (normally between 99 and 999)

1. Compute the observed statistic $F_\alpha = F_\alpha(A, \nu)$
2. For each replicate $r = 1, \ldots, R$ generate a random permutation $\pi_r = \pi(\nu)$ and compute the test statistic $F_\alpha^{(r)} = F_\alpha(A, \pi_r)$
3. Compute the empirical p-value

$$\hat{p} = \frac{\left\{1 + \sum_{r=1}^{R} I(F_\alpha^{(r)} \geq F_\alpha)\right\}}{R+1} \tag{2.36}$$

At each permutation the value of the test statistic $F_\alpha^{(r)}$ changes, but under H_0 the distribution of F_α does not change. Define the sequence of test statistics of permuted data as $F_\alpha^{(1)}, \ldots, F_\alpha^{(R)}$. Since the label is meaningless under H_0, the ranks of $F_\alpha, F_\alpha^{(1)}, \ldots, F_\alpha^{(R)}$ are uniformly distributed. In other words if it is possible to order these values, the test statistic computed in the original sample could be everywhere in the ordered sample.
The p-value is the fraction of times that the permuted statistic test $F_\alpha^{(r)}$ is large than F_α. We reject the null hypothesis when the empirical p-value is less than α'. There are not approximations or asymptotic theory applications and the distribution of the test statistics F_α and the distribution assumed in the null hypothesis do not matter.
An extension of distance correlation to functional data can be find in [10] where an application to multivariate functional data is showed. The authors prove the power of distance correlation in contrast with the functional canonical correlation who is not able to capture non-linear relationship between multivariate functional variables.

Chapter 3

Classification Energy Tree

In this chapter we introduce the classification tree algorithms and some existing methods for classification when data are functions. Then a new method for classification is proposed using distance correlation tests to measure association between the response variable and the functional covariates. Several application to simulated data are showed, ending with an application to real data for ECG measured on patients who may have a disease called LBBB.

3.1 Classification Tree Algorithms

In a sample of dimension n we observe a dependent variable $y_i = \{0, 1, \ldots, M\} \in \mathcal{Y}$ and a set of covariates $X = (X_1, \ldots, X_p) \in \mathcal{X}$. The aim of classification is to classify each observation starting from the observed value for the covariates. In statistics, classification problems are widely treated, Fisher [9] in 1936 introduced a method extended by the formalization of linear discriminant analysis [13]. Often the assumptions of multivariate normality and a common covariance matrix for LDA are not realistic for real data. Several methods are proposed to classify data, such as logistic regression model or support vector machine [4], but we focus on classifiers based on binary decision tree which makes no assumptions about the form of the underlying distribution.
The first idea of this approach can be find in Automatic Interaction Detection algorithms in [25] who had a series of developments such as CHAID used when explanatory variables are categorical [15]. The theory of classification tree-structured model was first introduced in the influential paper CART (Classification and Regression Trees) [5]. The algorithm usually works top-down, choosing one of p variables, then find the best value that splits data into two subsets, and iteratively repeats the

procedure. For example in Fig 3.1 we show graphically the output for classification tree for Iris data. We have measure for petal and sepal width and length and the class of iris flowers: setosa, versicolor and virginica. In the first node we can see the distribution of each class in the dataset, then two branches that connect the child nodes to father node. The first variable chosen is Petal Length, that is used to split the dataset in two subsets defined by the threshold value 2.4. For observations greater than the threshold we have only observations with state "Setosa", then the child node is pure, in other words we have only one class in the child node and we have no reason to split again. A different situation applies to the second child node, we could split again and we repeat the procedure. The variable chosen at second step is Petal Length and for a threshold value of 4.8 we partition the second subset in two subsets. The resulting nodes are almost pure, where two classes are present but the number of observation of one of them is negligible and the algorithm decides to not split again. In mathematical terms we have to find a function $d(x)$ that maps each point in $\mathcal{X}$ data into $\mathcal{Y}$. The usual form of $d(x)$ for classification aim is the expected misclassification cost. We have M distinct values in $\mathcal{Y}$ and can create a partition of $\mathcal{X}$ into M disjoint pieces $A_m = \{x : d(x) = m\}$ such that $\cup_{m=1}^{M} A_m = \mathcal{X}$. CART algorithms choose the combination between variable and observed values for each variable in $\mathcal{X}$ that optimizes a node impurity criterion such as Gini index $i(t) = 1 - \sum_{m=1}^{M} p^2(m|t)$ where t is a node and $p(m|t)$ is the proportion of of observations that belongs to class m in node t. For a split that divides data in two nodes named t_L and t_R of proportional dimension p_R and p_L respectively, the algorithm selects the split that maximizes the decrease in impurity

$$i(t) - p_L(t_L) - p_R(t_R) \tag{3.1}$$

As a remark, the procedure that leads to select a split suffers of bias toward variables [17].
Since the size of tree can increase without limits, a procedure called pruning has been proposed to reduce the size with optimal dimension. Define the cost-complexity criterion

$$R_a = MC + aL \tag{3.2}$$

where MC is the misclassification rate in root node and L is the number of leaves in the tree. The idea is to have low values for MC, while penalizing on the number of leaves in the tree. If we define T_0 as the biggest tree, we have to find a sub-tree T_a that minimizes Eq.3.2, choosing a by Cross-Validation.

In [11], the authors show a method that allows us to avoid the major problems encountered by classification trees: bias variable selection and overfitting. The

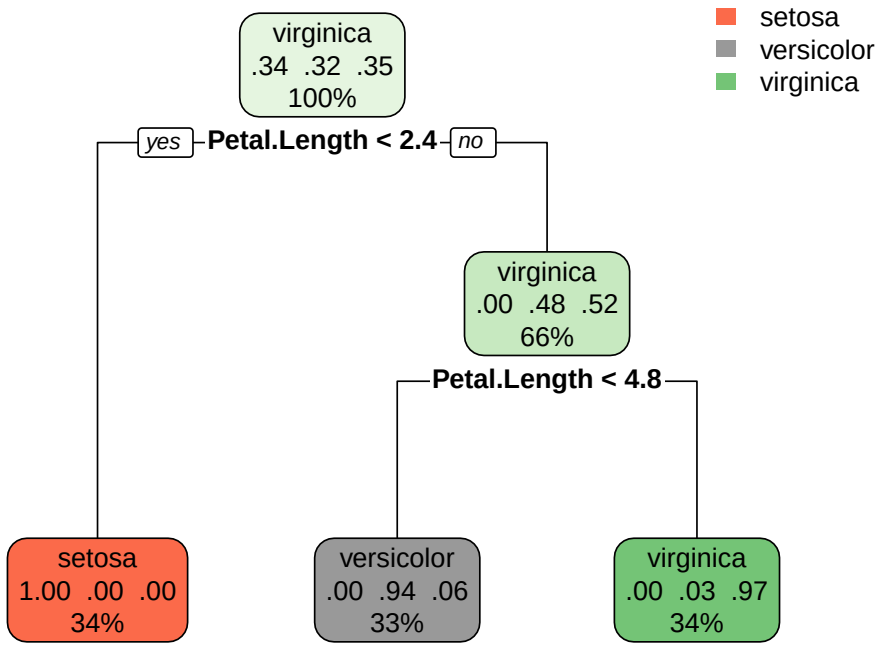

Figure 3.1: Classification tree example on iris data

CART algorithm and the following developments have not statistical evaluation at each step, and they have not a defined stopping rule. The basic idea of the unbiased procedure is to divide the unique step of search of variable selection and split value into two different phases. The first step is to test the global null hypothesis that there is not association between response variable and the covariates, $H_0 = \bigcap_{j=1}^{p} H_0^j$ where

$$H_0^j = D(Y|X_j) = D(Y) \tag{3.3}$$

Where $D(Y|X_j)$ is the conditional distribution of the response variable with respect the jth covariates.

If we are not able to reject the global null hypothesis we stop the algorithm. Otherwise, if we reject the global null hypothesis at a significance level α we select the

variable with lowest p-value obtained from every single test. The association between response variable and each covariates is measured with a statistic, a linear functional of the form::

$$T_j(Y, X, w) = vec\left(\sum_{i=1}^{n} w_i g_j(X_{ji}) h(Y_i, (Y_1, \dots Y_n))^T\right) \in \mathbb{R}^{p_j q} \tag{3.4}$$

where w is a vector of weights, g_j is a non-random transformation of X_j and h is the influence function. The distribution under H_0^j depends on the joint distribution of (Y,X) that is unknown. A permutation test is implemented fixing the covariates and conditioning on all permutations of response variable. The derivation of conditional expectation μ_j and covariance Σ_j under H_0 given all possible permutations, showed in [26], allows us to standardize the linear statistics $T \in \mathbb{R}^{pq}$. An example of univariate statistic that map the observed multivariate linear statistics into the real line, is the maximum of the absolute value of the standardized linear statistics

$$c_{max}(t, \mu, \Sigma) = \max_{k=1,\dots,pq} \left| \frac{(t-\mu)_k}{\sqrt{(\Sigma)_{kk}}} \right| \tag{3.5}$$

where c_{max} depends on data by t and μ and Σ are fixed.
If the X_j are measured at different scale we cannot directly compare each test statistic in an unbiased way. For this reason we compare the P-value computed at each test that are not inflected by the measurement scale. The procedure works as follows. As a first step, we select variable X_{j^*}, with $j^* = argmin_{j=1,\dots,p} P_j$ where

$$P_j = \mathbb{P}_{H_0^j}\left(c(T_j(Y, X, w), \mu, \Sigma) \geq c(t_j, \mu, \Sigma) | S(Y, X, w)\right) \tag{3.6}$$

where $S(Y, X, w)$ is the set of all possible permutation of Y. For a global testing, a multiple test procedure based on $P_1, \dots, P_p$ is used, such as Bonferroni adjusted p-values. We reject H_0 when the minimum of the adjusted p-values defined as

$$P_{adj} = 1 - (1 - P)^p \tag{3.7}$$

are less than α [24]. The splitting criteria follows the concept of the variable selection, once we have a selected variable j^* we can formalized a linear statistics similar to eq. 3.4 for all possible subsets A in space $\mathcal{X}^*$

$$T_{j^*}^A(Y, X, w) = vec\left(\sum_{i=1}^{n} w_i I(X_{j^* i} \in A) h(Y_i, (Y_1, \dots Y_n))^T\right) \in \mathbb{R}^q \tag{3.8}$$

that define a two sample statistics. Therefore we can compute the conditional expectation μ^A and conditional covariance Σ^A and we choose the best subset A^* such that

$$A^* = argmax_A c(t_{j^*}, \mu_j^*, \Sigma_j^*) \tag{3.9}$$

3.2 Classification Methods for Functional Variable

In literature several methods for classification are extended when covariates are functional. In [2] an extension for logistic regression is proposed for a binary response variable y_i, that without loss of generality shall take value 0 or 1. We can associate the response variable to a functional covariates $\{x_i(t), \quad t = 1, \ldots, T\}$ and we consider the the functional logistic model

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \int x_i(t)\beta(t)dt \tag{3.10}$$

where $\pi_i = \mathbb{P}(Y_i = 1|x_i)$ is the probability of Y conditional on a fixed value $x_i \in \mathcal{X}_i$. We can expand the parameter function $\beta(t)$ using a basis $\mathbf{\Phi}(t) = (1, \phi_1(t), \ldots, \phi_m(t))$ as

$$\beta(t) = \beta_0 + \sum_{k=1}^{m} \beta_k \phi_k(t) \tag{3.11}$$

An extension to multinomial logistic model is shown in [19]. Other methods to classify data such as Linear Discriminant Analysis are developed in [14].
The thesis focuses in classification within the framework of tree-based methods and an extension is proposed [3] that use representative curves to classify data: the algorithm starts computing representative curves with a clustering approach, then we partition data computing distances between observations and representative curves. Therefore a impurity measure score such as Gini index is computed and if this partition produces an improvement on Gini index we iterate the procedure, otherwise we stop the algorithm. This method has not statistical validation for the improvement of purity measure (such as in CART algorithm) and there is not extension for a multivariate functional observations. In the R package fda.usc [8] we can find a method that uses coefficients computed after the basis expansion and computes CART algorithm using coefficients as multivariate data.

3.3 Classification energy tree for functional variables

We propose a procedure that follow the unbiased framework introduced in [11] combined with the approach in [23], where the authors proposed a clustering method for curves after transforming and smoothing, and we consider the use of multivariate

functional covariates. We have a dependent variable $Y = \{0, 1, \ldots, M\}$ and a set of functional covariates $\{X_k\}_{k=1,\ldots p}$ where

$$X_{ik} = f_{ik}(t) + \epsilon_k \qquad \epsilon_k \sim \mathcal{N}(0, \sigma_k^2) \qquad i = 1, \ldots, n \quad k = 1, \ldots, p \tag{3.12}$$

Remember that we observe discrete values for each observation equally spaced for simplification in $t_1, \ldots, t_m$ and we can derive the bases expansion using a set of bases and computing the coefficients using the methods proposed in Chapter 1. The first steps are to transforming and smoothing, choosing the number of basis according to the methods introduced in 1.3.2. We refer always to a cubic B-spline basis and the number of bases is given by the sum of the order of the B-spline (in this case 3) plus one and the number of the interior knots

$$J = 4 + \text{Number of interior knots} \tag{3.13}$$

At the end of the smoothing process we have a functional observation defined as $\hat{f}_{ik}^J(t) = \sum_{j=1}^J \hat{\theta}_{ijk}\phi_j(t)$, for $\quad i = 1, \ldots, n \quad k = 1, \ldots, p$.
The next step is to global test if there is association between the response variable and the functional covariates. For this purpose we use the extension of analysis of variance using distance correlation introduced in 2.3 based on permutation test based on distances computed from functional observations using Simpson's rule [8], which is a L2 weighted distance

$$d_{ij} = ||f_i(t) - f_j(t)||^2 = \left(\frac{1}{\int_a^b w(t)dt} \int_a^b |f_i(t) - f_j(t)|^2 w(t)dt \right)^{\frac{1}{2}} \tag{3.14}$$

We obtain a sequence of p-value $P_1, \ldots, P_p$ that we adjust Bonferroni correction as in Eq.3.7. If none of the adjusted p-values is lower than the significance level α then the algorithms stops, otherwise we select the functional covariates f_{k^*} with the minimum adjusted p-value $P_{k^*}^{adj}$, rejecting the hypothesis that there is not difference between classes.
Next we need to split the data into two subsets and we use the estimated coefficients obtained from the basis expansion. We use again the analysis of variance with distance correlation on coefficients $\hat{\theta}_{k^*1}, \ldots, \hat{\theta}_{k^*J}$. The sequential tests give us a sequence of p-value $P_{k^*1}, \ldots, P_{k^*J}$ where we use the Bonferroni correction and we select the coefficient with the minimum adjusted p-value $P_{k^*j^*}$. The final step is to find the coefficient that split data. We ordered the values of selected coefficient $\hat{\theta}_{1k^*j^*} \leq \ldots \leq \hat{\theta}_{nk^*j^*}$. We select the coefficient that minimizes the entropy as

$$\hat{\theta}_{i^*k^*j^*} = \min_{i=1,\ldots,n} \left(-\sum_{m=1}^{M} p_{im} \log p_{im} \right) \tag{3.15}$$

where p_{im} are the proportions of labels of classes. The procedure repeat iteratively the algorithm on each subset until the stopping rule is verified, in other words we

are not able to reject the null hypothesis, or the node is pure.
All the nodes and the split decision are computed modifying an existing code in *partykit* R package [12]. The algorithm outline is showed below

1. Transform data $(X_{ik}(t_1), \ldots, X_{ik}(t_m)) \rightarrow \left(\hat{\theta}_{ik1}, \ldots \hat{\theta}_{ikm}\right)$ for $i = 1, \ldots, n \quad k = 1, \ldots, p$
2. Smooth data $\hat{f}_{ik}^J = \sum_{j=1}^J \hat{\theta}_{ijk}\phi_j(t)$, for $\quad i = 1, \ldots, n \quad k = 1, \ldots, p$
3. Global test if there is association between Y and $\{X_k\}_{k=1,\ldots,p}$
4. Choose the functional covariates $\hat{f}_{ik^*}^J$ associated the minimum adjusted P-value
5. Choose coefficient $\hat{\theta}_{k^*j^*}$ associated with the minimum adjusted P-value
6. Split data for the observed coefficient value that minimizes the entropy
7. Iterate the procedure on the two obtained subsets

For a complete evaluation of different methods for classification we predict the class label starting from "new" observations, usually we partition data in train and test subsets and use the test dataset for this purpose. For comparison with the other methods we compute the overall accuracy starting from the confusion matrix for a multiclass framework defined in Tab.3.1 computing

$$Acc = \frac{\sum_{j=1}^M n_{jj}}{n_{test}} \tag{3.16}$$

where n_{test} is the dimension of the test dataset

	Class 1	n_{11}	n_{12}	$\ldots$	n_{1M}
True class	Class 2	n_{21}	n_{22}	$\ldots$	n_{2M}
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	Class M	n_{M1}	n_{M2}	$\ldots$	n_{MM}
		Class 1	Class 2	$\ldots$	Class M
		Predicted class			

Table 3.1: Confusion matrix for a multiclass classification framework

3.4 Simulation study

In this section we show different applications to simulated univariate and multivariate functional covariates where we evaluate the classification power using the train and test framework for a repeated simulated data. On test subset we estimate the class membership with selected model and we compute the accuracy of the estimation.

3.4.1 Univariate Functional Covariate

We applied the classification energy tree to a simulated CBF dataset proposed in [22] where we compute signals for cylinder, bell. funnel classes of the form

$$c_i = (6+\eta)\chi_{[a,b]i} + \epsilon_i \tag{3.17}$$

$$b_i = (6+\eta)\chi_{[a,b]i}(i-a)/(b-a) + \epsilon_i \tag{3.18}$$

$$f_i = (6+\eta)\chi_{[a,b]i}(b-i)/(b-a) + \epsilon_i \tag{3.19}$$

where $i = 1, \ldots, 128$, a is an integer value randomly selected from uniform distribution on the interval $[16, 32]$, $b-a$ is also an integer valued selected from uniform distribution on $[32, 96]$, η and $\epsilon_i \sim \mathcal{N}(o, \sigma^2)$ and $\chi_{[a,b]i}$ is the characteristic function on the interval $[a, b]$. For each class we simulate $n = 266$ observations that we show in fig.3.2 when $\sigma^2 = 1$. We simulate $M = 100$ dataset and we randomly divide data into train and test subsets of dimension 70 % and 30% respectively. The first step is to transforming and smooth data using a cubic B-spline as base functions. The number of interior knots and for λ is chosen by Generalized Cross Validation following the procedure introduced in 1.3.2, in Fig. 3.3 we show the result of the transformation and smoothing step for a simulated dataset. After the transformation and smoothing step, for each train data we execute the classification energy tree model (we show an output in Fig.3.4). For a comparison we repeat the classification with other methods already existing such as classification tree implemented in [8] and functional multinomial logistic model. In fig 3.4 we show the output for one of the simulated CBF dataset where we can see that the first chosen coefficient is the 5th and the split value is 3.04. We consider the prediction accuracy defined in Eq.3.16 and in tab 3.2 we show the mean and the standard deviation of accuracy computed at each test dataset and for different values of σ . The classification energy tree has the highest accuracy mean for prediction aim and this mean decrease as σ increase. We can compare the tree methods for functional data with a measure called depth of a decision tree, that is the length of the longest path from a root to a terminal node. We compare with the other classification tree method proposed and we

show in Tab 3.3 and we observe that the proposed method has greater size with respect the already existing method, this lead to a conclusion that our algorithm can capture some dynamics that our method manages to capture certain dynamics otherwise neglected by the regression tree and this can be reflected also on the prediction performance. As expected, the depth of both methods increase as the variance term increase.

Table 3.2: Accuracy mean and standard deviation for 100 samples of Saito dataset simulated for different values of σ^2 and for different classification methods

	$\sigma = 0.5$		$\sigma = 1$		$\sigma = 1.5$		$\sigma = 2$	
Method	Mean Acc.	SD Acc.	Mean Acc.	SD Acc.	Mean Acc.	SD Acc.	Mean Acc.	SD Acc.
Classification Energy Tree	0.9592	0.0172	0.9408	0.0194	0.9172	0.0205	0.8845	0.0224
Functional Tree [8]	0.8518	0.0246	0.8487	0.0253	0.8457	0.0262	0.8270	0.0271
FMLM	0.8533	0.0396	0.8576	0.0390	0.8703	0.0475	0.8474	0.0347

3.4.2 Multivariate Functional Covariates

We extend the process to multivariate functional covariates, simulating datasets that have three functional covariates and a dependent variable $Y \in \{R, B, G\}$, in other words we consider a three class problem as in the univariate case. For each class we simulate n observations that follow a centerline of the form

$$X(t) = m(t) + \epsilon(t) \qquad \text{where} \qquad Cov(\epsilon(t), \epsilon(s)) = Cov(s, t) \tag{3.20}$$

In Tab.3.4 we show the centerline for each class and for the simulated covariates we use different values for a and b. For the covariance functions we sample α and β from a uniform distribution in $[0, 1]$, a is sampled from discrete uniform in $[1, 4]$ and b from a uniform in $[1, 4]$ with a covariance of the form $Cov(s, t) = \alpha e^{-\beta|s-t|}$ and $t = 1, \ldots, 100$. We add some random noise to obtain a complete evaluation, so we finally observe

$$X_k^* = X_k(t) + \mathcal{N}(0, \sigma^2) \qquad \text{for } k = 1, 2, 3 \tag{3.21}$$

In Fig.3.5 we show the simulated multivariate functional data for three classes and $\sigma = 0.1$. The different colors indicates the class membership and the column indicates the covariates membership.

Table 3.4: Centerline formula for each class

Class	Centerline
R	$\cos(a + b\pi t)$
B	$\sin(a + b\pi t)$
G	$\cos(a + b\pi t)\sin(a + b\pi t)$

We simulated $M = 100$ datasets and, following the procedure, the first step is to transform and smooth the discrete data according with the usual framework. We use the estimated function in classification energy tree algorithm. For the aim of classification we split, as in the univariate case, the dataset in train and test subset, respectively of dimension 70% and 30%. We repeat the simulation and classification for each simulated dataset and we show in Fig.3.6 an output of classification energy tree. The tab 3.6 shows the results of different methods for the simulated datasets, as in the previous case we can see that classification energy tree has the highest accuracy for prediction with respect the other methods.

Table 3.3: Depth mean and standard deviation for 100 samples of Saito dataset simulated for different values of σ^2 and for different classification methods

	$\sigma = 0.5$		$\sigma = 1$		$\sigma = 1.5$		$\sigma = 2$	
Method	Depth mean	Depth SD	Depth mean	Depth SD	Depth mean	Depth SD	Depth mean	Depth SD
Classification Energy Tree	5.92	1.21	6.2	1.19	6.6	0.95	6.78	0.95
Functional Tree [8]	5.22	1.46	4.5	1.14	4.28	0.86	4.46	1.12

Table 3.5: Accuracy mean and standard deviation for 100 samples of multivariate datasets simulated for different values of σ^2 and for different classification methods

	$\sigma = 0.1$		$\sigma = 0.5$		$\sigma = 1$	
Method	Mean Acc.	SD Acc.	Mean Acc.	SD Acc.	Mean Acc.	SD Acc.
Classification Energy Tree	0.8926	0.0604	0.8415	0.0705	0.7169	0.0769
Functional Tree [8]	0.7374	0.0880	0.7415	0.0875	0.6963	0.0968
FMLM	0.5310	0.1194	0.4990	0.1036	0.5360	0.1114

Table 3.6: Depth mean and standard deviation for 100 samples of multivariate datasets simulated for different values of σ^2 and for different classification methods

	$\sigma = 0.1$		$\sigma = 0.5$		$\sigma = 1$	
Method	Depth mean	Depth SD	Depth mean	Depth SD	Depth mean	Depth SD
Classification Energy Tree	3.42	0.57	3.9	0.67	4.58	0.67
Functional Tree [8]	3.27	0.69	3.31	0.58	3.33	0.62

3.5 A case study

We present the power of the classification tree framework explained and motivated above for a real dataset. We have a sample of 100 patients and on each of them we detect an 8-lead ECG, the signals have been registered and smoothed over an evenly spaced grid of 1024 time points at 1kHz. The peculiarity is that the patients are 50% healthy while the remaining half has a disease called Left-Bundle-Branch-Block (LBBB), which results in irregularities in ECG traces and this will be our response variable Y with two levels: Healty and LBBB; the data are furnished in R package *roahd* [32] . We observe 8 curves for each patient, so the dimension of the multivariate functional is 8 $\{X_k\}_{k=1,\dots,8}$and, as already said, the data are smoothed with an unknown method. The data are showed in Fig. 3.7 where the blue line indicates the healthy patients and the purple the LBBB. Graphically we can see the difference between curves traces of two classes at some point of the ECG. But, since in our algorithm we need to know the coefficients of the expansion in basis, we re-smooth data using cubic B-spline method. The aim is to classify correctly the belonging of the curves to the right class and to avoid overfitting we decide to use a 10-fold Cross Validation. In other words we partition the complete dataset in ten equal subsets, 1/10 of the original dataset will compose the test subset and the rest 9/10 will be the train subset. We run classification energy tree and the other methods used in simulation study to the train subset and we make prediction on test subset. We repeat the procedure changing the test subset and one of the 9/10 of the train data, in this way all the partition will compose the train and test data in turn. At the end we predict the belonging to the class for all sample observations, comparing with the true observed values for the response variable. In Fig. 3.8 the output for 1-fold is showed. The first functional variable chosen is the third, that, as we can see from the original curves, are very different among classes. The algorithm keep on until the children nodes are pure or we have not sufficient difference between classes. In Tab. 3.7 we show the confusion matrix for the classification Energy Tree, where the diagonal element indicates the correct classification. The accuracy for the proposed method is 0.95, greater than the accuracy for functional tree and functional multinomial logit model that are equal to 0.84 and 0.71 respectively.

Table 3.7: Confusion matrix for predicted values of dependent variable versus the true values

	Healthy	LBBB
Healthy	46	1
LBBB	4	49

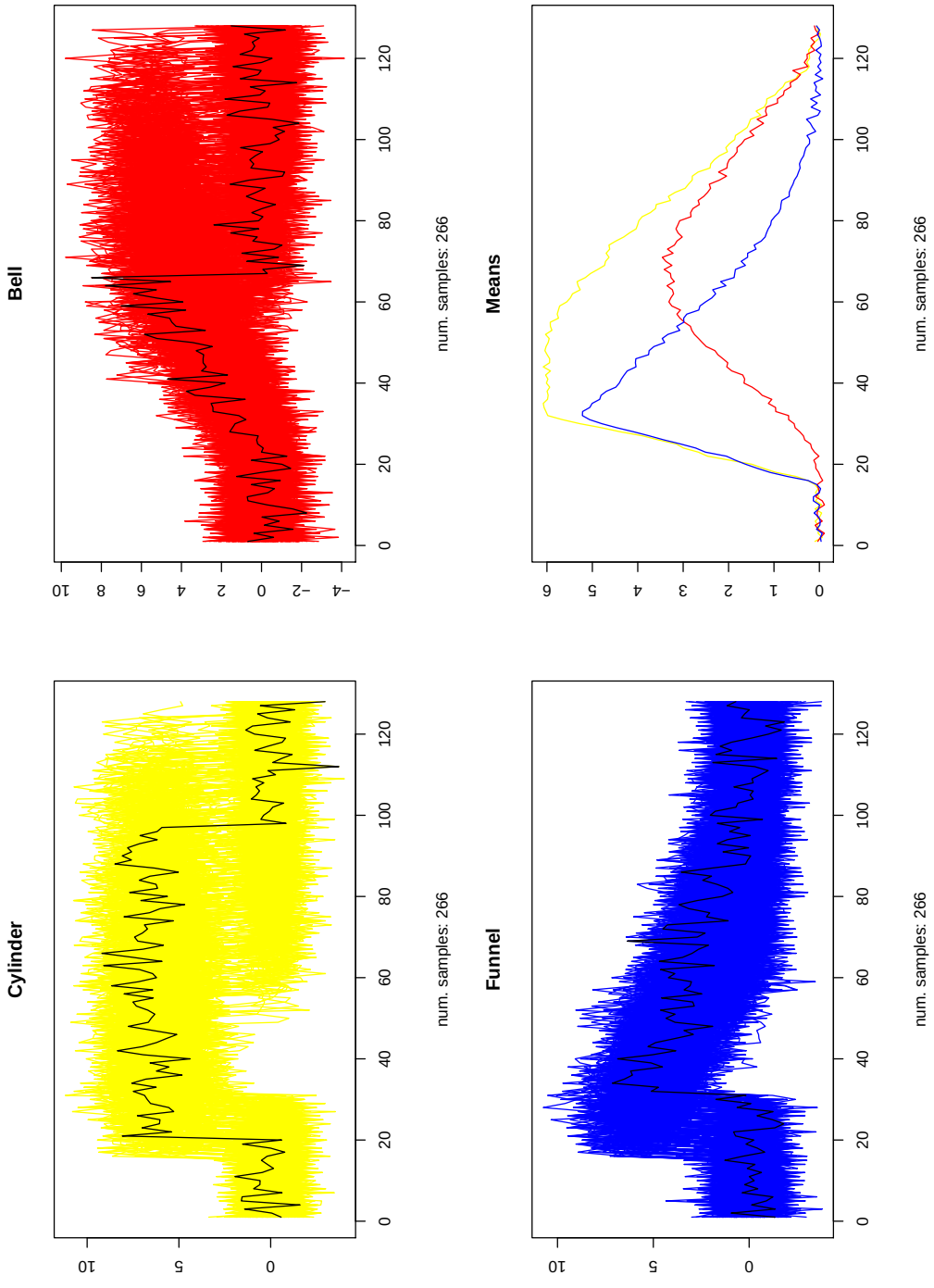

Figure 3.2: Cylinder, Bell, Funnel dataset. The bold line in each plot represent a single observation. In the bottom right we show the mean function for each class.

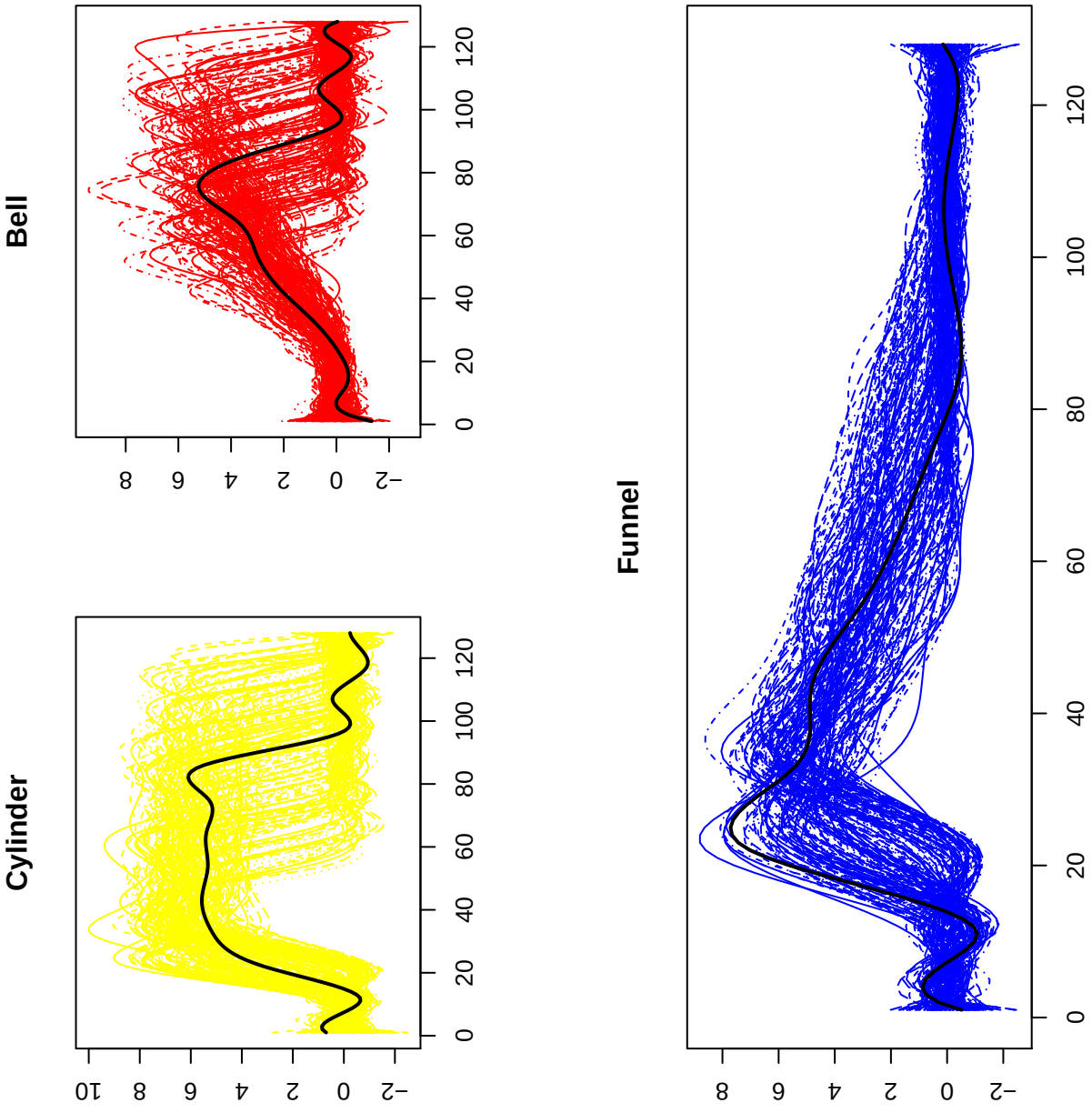

Figure 3.3: Transformed and smoothed data with cubic B-spline with 15 bases (11 interior knots)

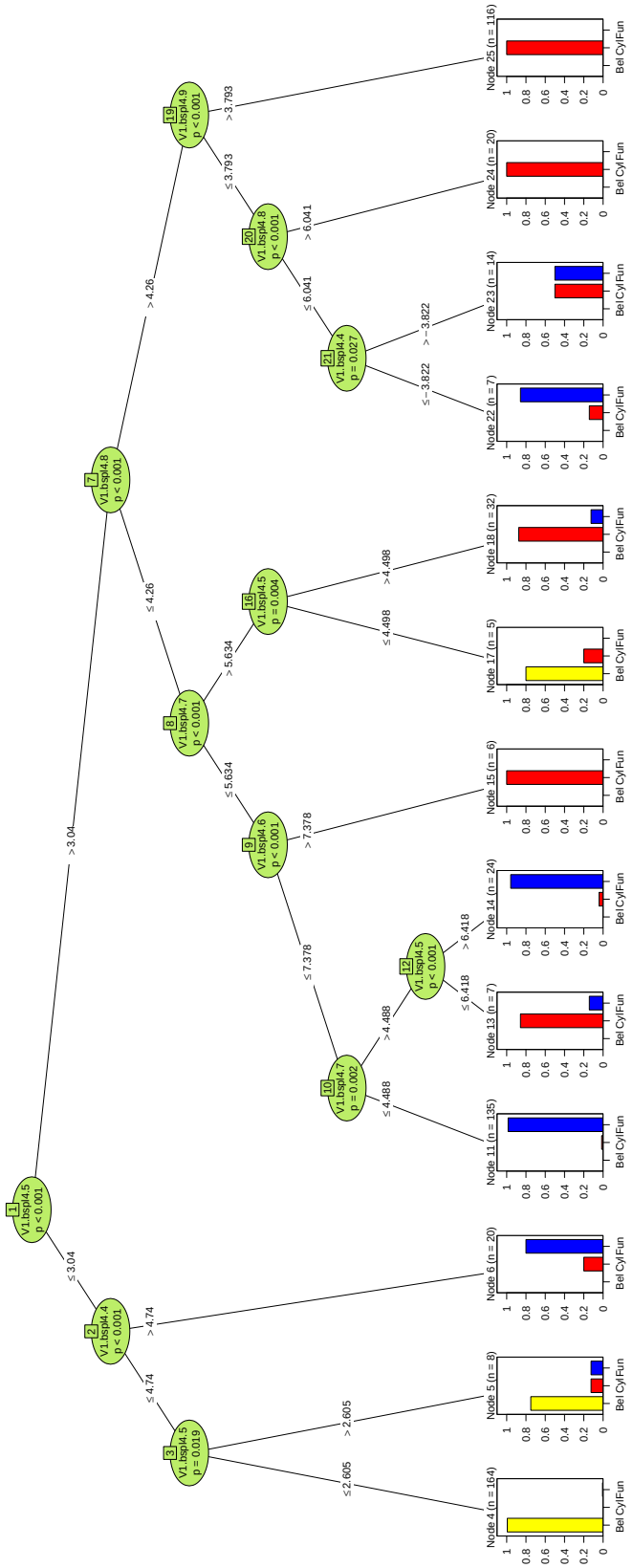

Figure 3.4: Output of classification energy tree for a train simulated data of Saito dataset. In each node we indicate the combination between functional covariates and chosen coefficient, also the p-values obtained from energy test are showed. In the branches are indicated the chosen values of the splits.

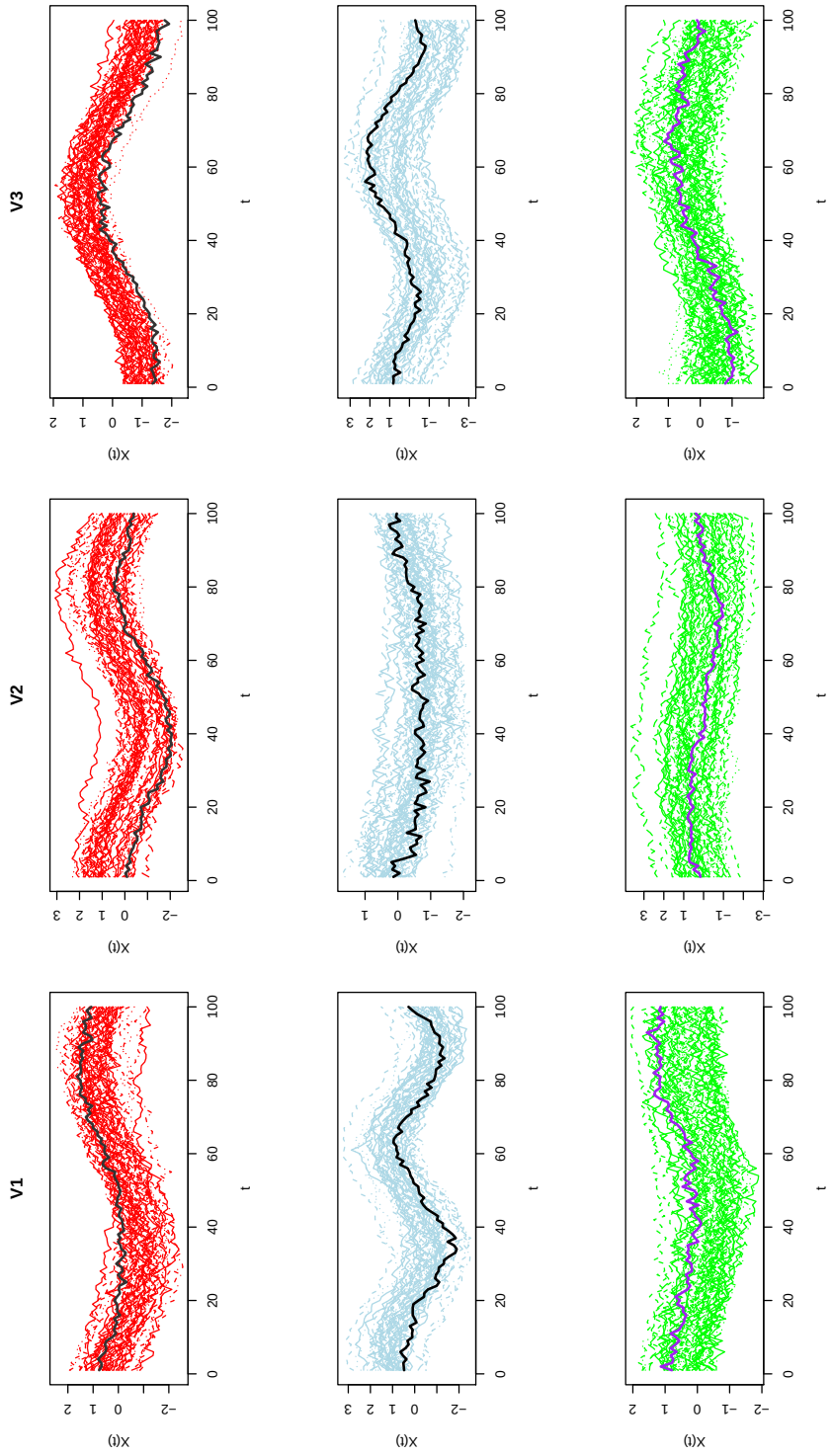

Figure 3.5: Simulated data for 3 variate functional variables for 3 classes with $\sigma = 0.1$. The red observations are from first class, blue observations are from second class and green observations are from third class. The bold line in each plot represents a single observation for the different classes and for the covariates

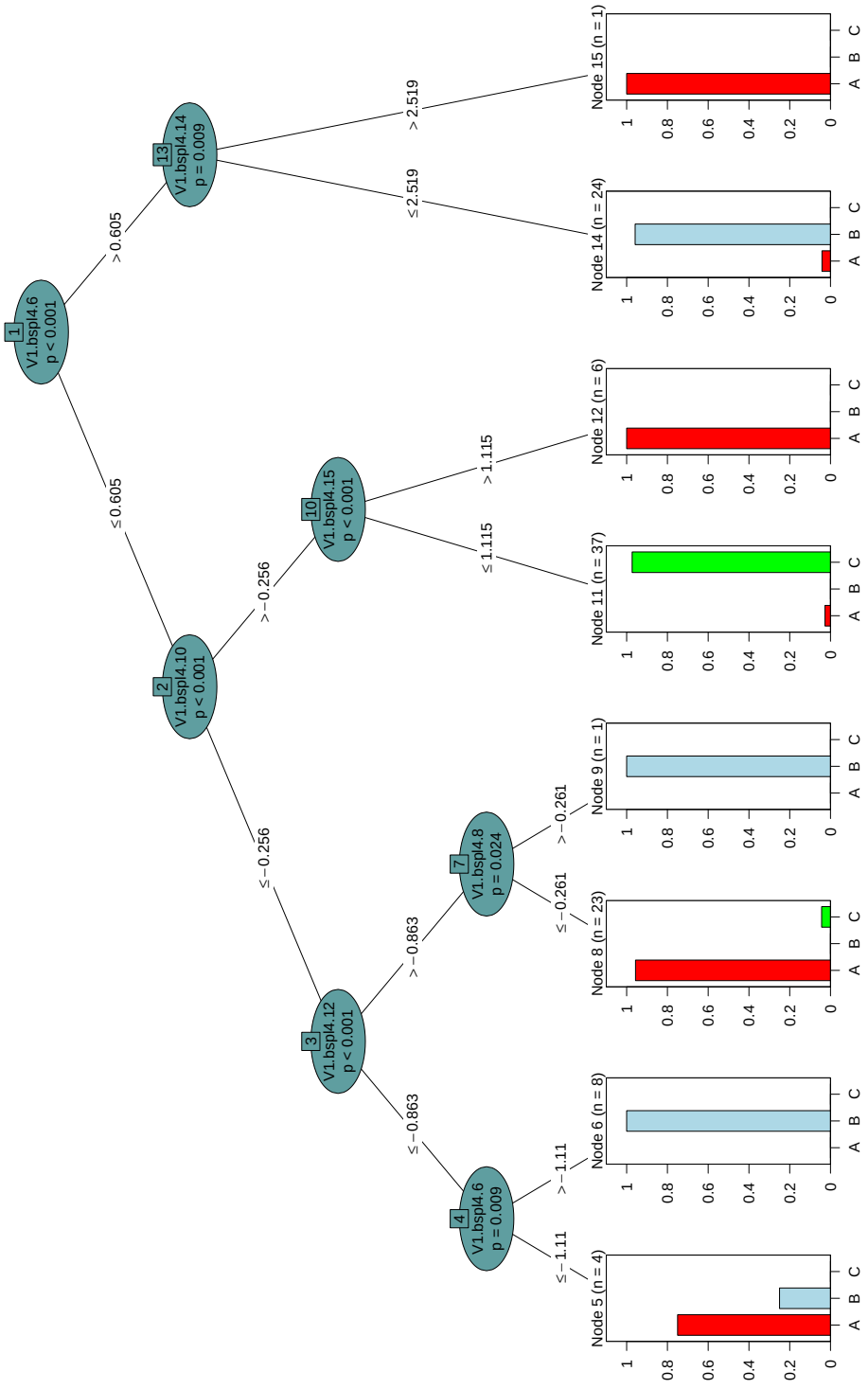

Figure 3.6: Output of classification energy tree for a train simulated data of a 3 variate functional data. In each node we indicate the combination between functional covariates and coefficient selected at each step, and the p-values obtained from energy test

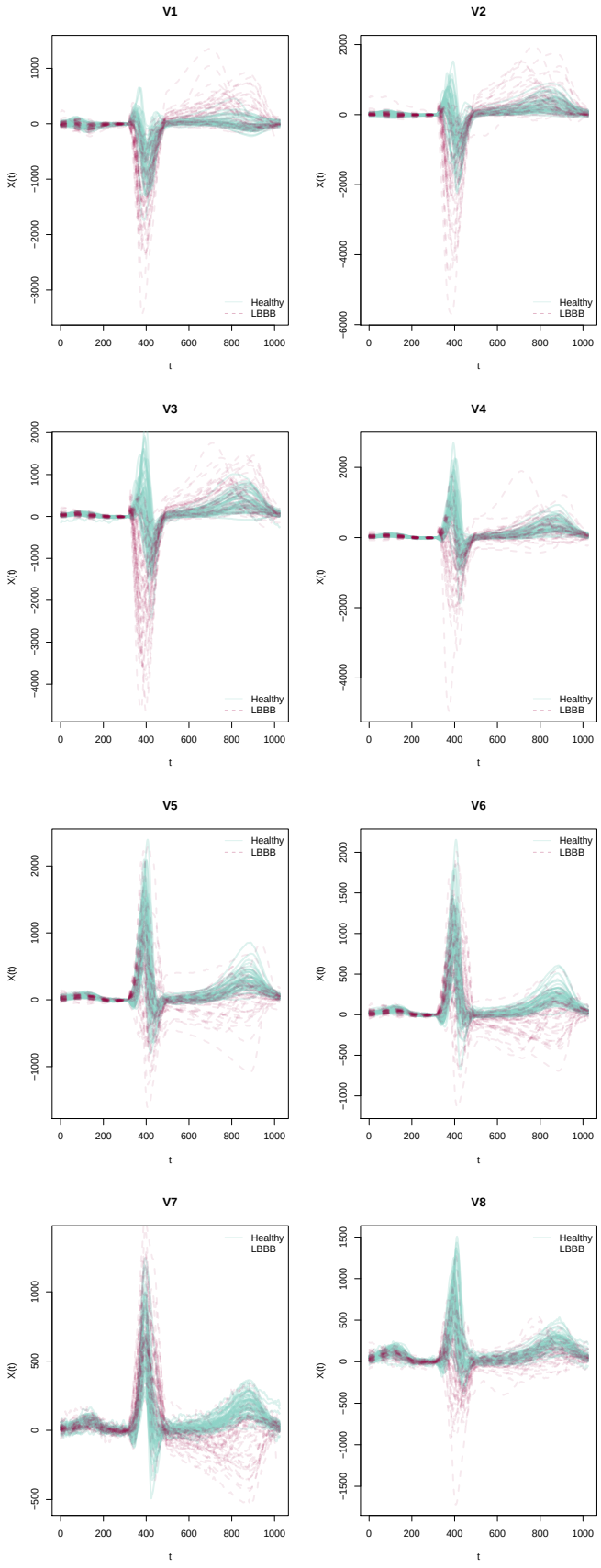

Figure 3.7: Data of 8-lead ECG detected on 100 patients, the blue lines are the healthy patients and the purple are the patients with LBBB disease.

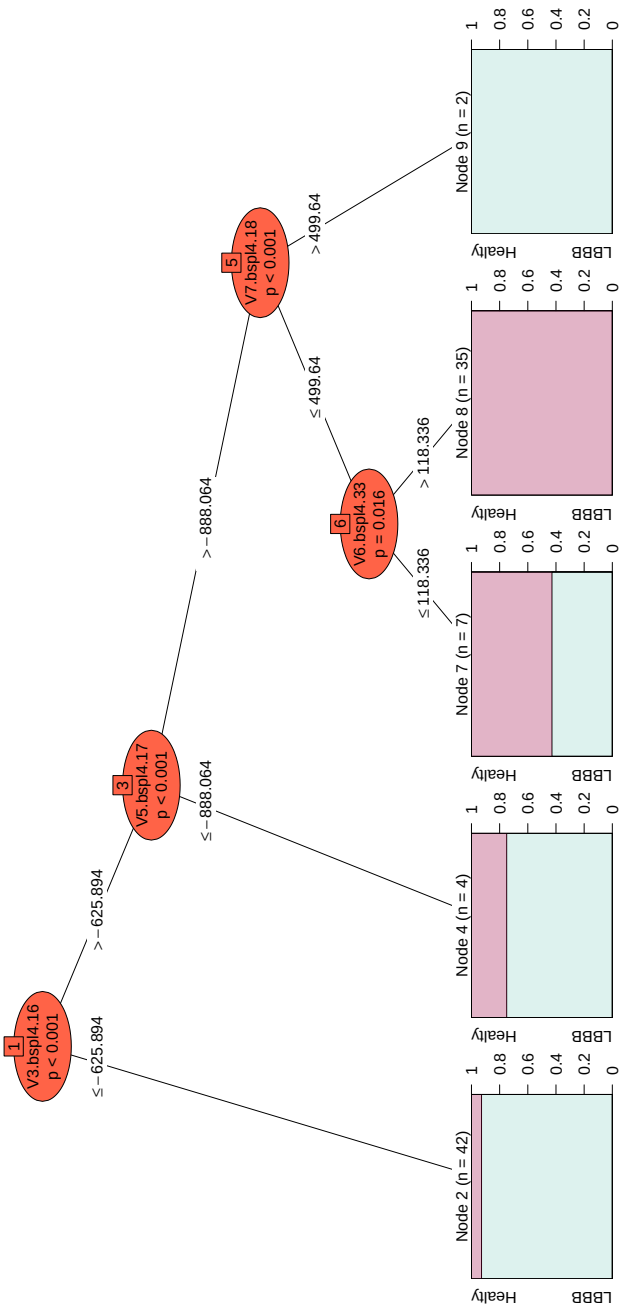

Figure 3.8: Output of regression energy tree for 8-lead ECG detected on 100 patients for 1 fold.

Chapter 4

Regression Energy Tree

In this chapter we define the general class of functional linear and we compare them with the proposed method of regression energy tree, modifying the procedure proposed in Chap. 3 when the response variable is continuous. We compare these methods measuring the prediction performance.

4.1 Functional Linear Model

The tree framework can be applied also when the dependent variable is continuous, $Y \in \mathbb{R}$, modifying partially the procedure proposed before. The natural competitor of regression tree for functional data is the functional linear model [21] defined as

$$Y_i = \beta_0 + \int X_i(t)\beta(t)dt + \epsilon_i \tag{4.1}$$

where β_0 is the intercept term, $X_i(t)$ is the functional covariate, $\beta(t)$ is the functional parameter and ϵ_i is the error term assumed to be normal with null mean and common variance σ^2. As usual in the functional context we can use the basis expansion for both terms

$$X(t) = \sum_{k=1}^{K1} c_k\phi_k(t) \qquad\qquad \beta(t) = \sum_{k=1}^{K2} v_k\psi_k(t) \tag{4.2}$$

The functional regression term can be written

$$\int X_i(t)\beta(t)dt = X_i^* J_{\phi,\psi} B_i^* \tag{4.3}$$

Where $X_i^* = [c_{i1}, \ldots, c_{iK1}]$ and $B_i^* = [v_{i1}, \ldots, v_{iK2}]$ and $J_{\phi,\psi} = \int \phi(t)\psi'(t)dt$. Coefficients and number of bases can be chosen with methods already introduced in Chapter 1 via the bias-variance trade-off or the roughness penalization and we can estimate parameters using the multivariate regression framework. Note that if we

choose the same basis for data and coefficients the integral $J_{\phi,\psi}$ is the identity matrix.
In [6] is introduced a functional linear model using principal components method for parameter estimation, assuming that functional data and functional parameter can be written respectively as

$$X_i(t)\sum_{k=1}^{\infty}\gamma_{ik}\nu_k \qquad\qquad \beta(t)=\sum_{k=1}^{\infty}\beta_k\nu_k \tag{4.4}$$

where $\gamma_{ik}=\langle X_i(t),\nu_k\rangle$ and $\beta_k=\langle\beta(t),\nu_k\rangle$ and $\langle\cdot\rangle$ denotes the inner product. We can estimates $\beta(t)$ using few principal components obtained via spectral decomposition of the covariance function. The number of chosen principal components k_n lead to a truncation and the direct consequence is that $\beta_k=0$ for $k>k_n$. We can approximate the integral in Eq 4.1 by

$$\hat{Y}=\langle X,\beta\rangle=\sum_{k=1}^{k_n}\gamma_{ik}\hat{\beta}_k \tag{4.5}$$

where $\hat{\beta}_1=\frac{\gamma_{\cdot 1}^T Y}{n\lambda_1},\ldots,\hat{\beta}_{k_n}=\frac{\gamma_{\cdot k_n}^T Y}{n\lambda_{k_n}}$, λ_i are the eigenvalues of the principal components. The choice of number of components is done by cross validation that minimized the predictive power defined as

$$PCV(k_n)=\frac{1}{n}\sum_{i=1}^{n}\Big(Y_i-\langle X_i,\beta_{(i,k_n)}\rangle\Big)^2 \tag{4.6}$$

4.2 Regression Energy Tree Algorithm

We build the regression tree using the same framework of decision tree in Chap.3. The first step is to transform and smooth data, then we measure the association between the response variable Y and the functional transformed and smooth covariates $\hat{f}^J(t)$ using distance correlation test introduced in Chap.2.2, where distance correlation is computed in our case as

$$\mathcal{R}_n^*(Y,\hat{f}_k^J(t)) \tag{4.7}$$

where $\sqrt{\nu-1}\mathcal{R}_n^*$ with $\nu=\frac{n(n-3)}{2}$ asymptotically follow a standard normal distribution. Repeating this procedure for each of the covariates we have the usual sequence of p-values, which must be correct to avoid the multiplicity problem with the Bonferroni correction. Again, if the minimum adjusted p-value given by the test is less than the significance level α we reject the hypothesis of independence between the response variable Y and the functional covariates $\hat{f}_k^J(t)$. Of course we use the

distances computed for the response variable and the distances for the functional covariates such as

$$d_{ij}^1(Y) = ||Y_i - Y_j||^2 \tag{4.8}$$

$$d_{ij}^2(\hat{f}_k^J(t)) = ||f_{ik}(t) - f_{jk}(t)||^2 = \left(\frac{1}{\int_a^b w(t)dt} \int_a^b |f_{ik}(t) - f_{jk}(t)|^2 w(t)dt \right)^{\frac{1}{2}} \tag{4.9}$$

to test the association between the response variable Y and the covariates. In order to binary split data we use the same method used in 3.3. We choose the coefficient of the basis expansion for split data, recomputing the distance correlation test, searching for an association between the response variable and one of the coefficients. In other words we compute

$$\mathcal{R}_n^*(Y, \hat{\theta}_j) \text{ for } j = 1, \ldots, J \tag{4.10}$$

Obviously we still have to consider the multiplicity problem and we correct the obtained p-values with Bonferroni correction. We select coefficient with minimum adjusted p-value on condition that is less than α, In other words we take $\hat{\theta}_{j^*k^*}$ associated with the $p_{j^*k^*}^{adj} = \min_{j=1,\ldots J} P_{jk^*}^{adj}$ and on condition that $p_{j^*k^*}^{adj} \leq \alpha$. The final step is to choose a value between the observed n values of coefficient $\hat{\theta}_{j^*k^*}$ to split data in two subsets and in contrast with the classification method (for the nature of the response variable), we choose the value that minimizes the variances of response variable Y in the two subsets. Eventually we recursively apply the procedure in the obtained subsets, stopping it when we accept the hypothesis of independence. In the following list we summarize the procedure:

1. Transform data

$$(X_{ik}(t_1), \ldots, X_{ik}(t_m)) \to \left(\hat{\theta}_{ik1}, \ldots \hat{\theta}_{ikm}\right) \text{ for } i = \quad 1, \ldots, n$$
$$k = \quad 1, \ldots, p$$

2. Smooth data

$$\hat{f}_{ik}^J = \sum_{j=1}^{J} \hat{\theta}_{ijk}\phi_j(t), \text{ for } \quad i = 1, \ldots, n \quad k = 1, \ldots, p$$

3. Global test if there is association between Y and $\{X_k\}_{k=1,\ldots,p}$.
   Distance Correlation test with distances computed with Simpson's rule for the functional covariates and Euclidean for the dependent variable Y

4. Choose the functional covariates $\hat{f}^J_{ik^*}$ associated the minimum adjusted P-value using Bonferroni correction conditionally that is less than α

5. Choose coefficient $\hat{\theta}_{j^*k^*}$ associated with the minimum adjusted P-value if less than α

6. Partition data for the observed coefficient value that minimizes the variances of Y in the subsets

7. Iterate the procedure on the two obtained subsets

4.3 Simulation Study

We simulate regression functional data according to synthetic data showed in [23] from the regression model

$$Y_{ij} = f(t_{ij}) + \epsilon_{ij} \tag{4.11}$$

with $j = 1, \ldots, m$ and $t_{jm} = j/m$ The regression functions for f are

$$F_1(t) = \left(\frac{2-5t}{2}\right) \wedge \left(\left(\frac{5t-2}{3}\right)^2 + \sin\frac{5\pi t}{2}\right) \tag{4.12}$$

$$F_2(t) = -F_1(t) \tag{4.13}$$

$$F_3(t) = \cos 2\pi t \tag{4.14}$$

$$F_4(t) = -F_3(t) \tag{4.15}$$

The error term ϵ_{ij} is assumed to be normal with zero mean and variance σ^2. We simulate 50 observations for each of the four curve shapes F_1,F_2,F_3 and F_4 over $m = 50$ design points. The response variable is taken as $Y_i = \mu_i(F(t_1), \ldots, F(t_m))$ with F taking one of the four shapes. The 200 non constant curves are showed in Fig.4.1. We repeat the simulation framework used in the classification setting, generating $M = 100$ datasets for different noise level. We execute the regression energy tree and the functional regression method presented in this chapter and as comparison for the fitted model we compute the mean square error prediction (MEP) in according with [1] defined as

$$MEP = \left(\sum_{i=1}^{n} \frac{(Y_i - \hat{Y}_i)^2}{n}\right) / Var(Y) \tag{4.16}$$

In Fig.4.2 we show the output for one of the simulated data and we can see in the terminal nodes that the recursive partitioning seems to partition data in crescent order and with a low variance in each terminal node.

In Tab4.1 the results of obtained MEP for each methods and for different noise levels are showed, and the Regression Energy Tree seems to perform better with respect the functional linear models proposed. When the noise levels increases, The MEP grows and the difference between methods performance is reduced.

Table 4.1: Mean and standard deviation of mean square error of prediction (MEP) for 100 samples of simulated regression dataset for different values of σ and for different functional regression methods

	$\sigma = 0.5$		$\sigma = 1$		$\sigma = 1.5$		$\sigma = 2$	
Method	Mean MEP	SD MEP	Mean MEP	SD MEP	Mean MEP	SD MEP	Mean MEP	SD MEP
Regression Energy Tree	0.1507	0.0279	0.1546	0.0308	0.2632	0.0403	0.3262	0.0511
Functional Linear Model	0.1717	0.0241	0.1745	0.0234	0.2657	0.0265	0.3237	0.0287
Functional Linear Model PC	0.2800	0.0244	0.2834	0.0244	0.3313	0.0293	0.3730	0.0337

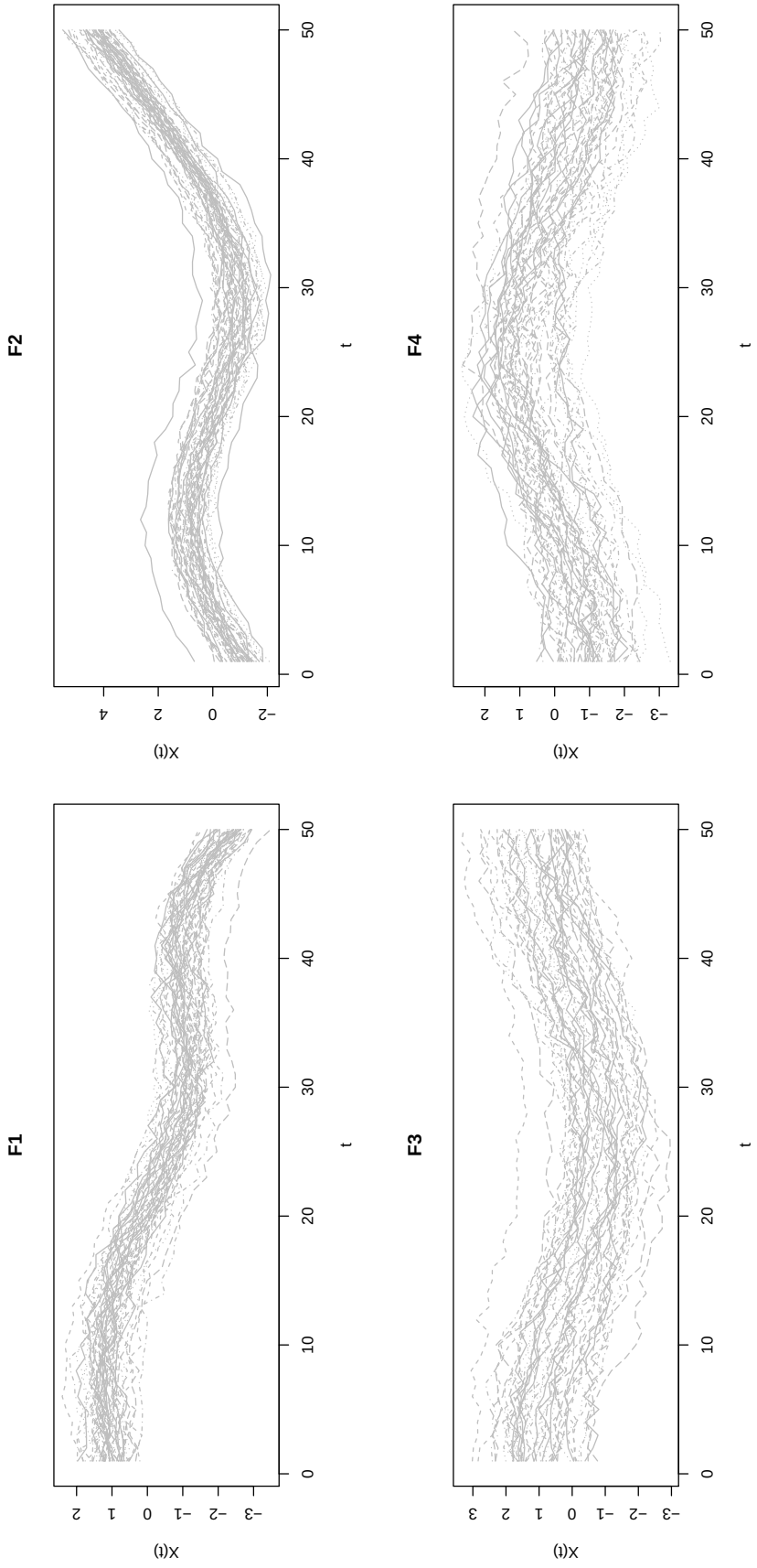

Figure 4.1: Simulated regression functional data for the four shapes defined in Eq. 4.12 for a noise level of $\sigma = 0.5$

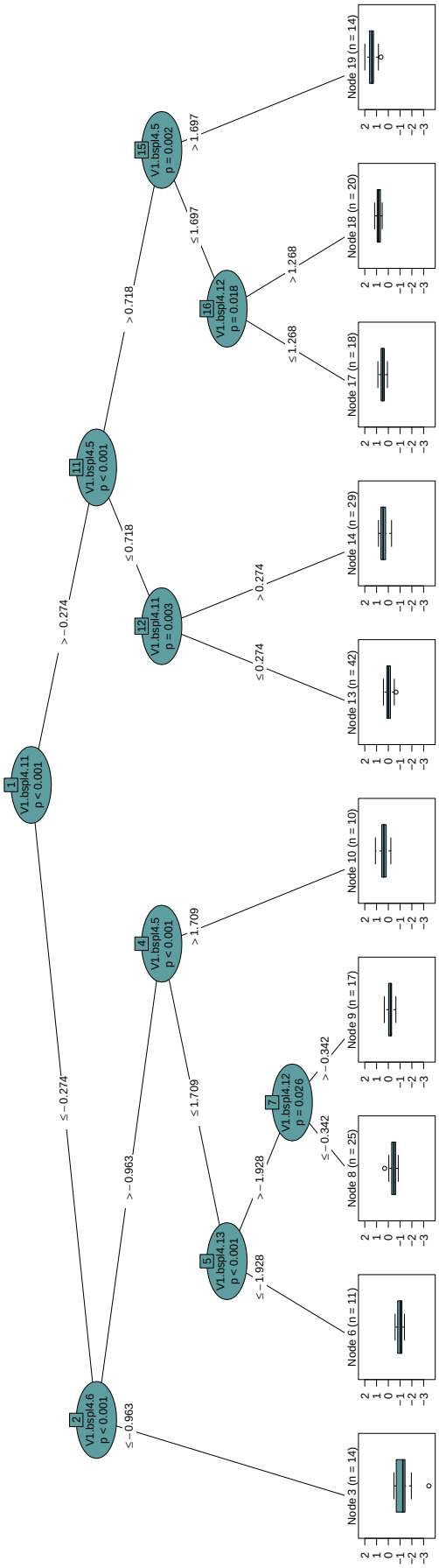

Figure 4.2: Output of regression energy tree for a simulated dataset with noise level of $\sigma = 0.5$

Conclusions and Additional Consideration

Nowadays the increasing information available requires to build more complicated model for more complicated data. The usual theory of multivariate analysis can lead to erroneous evaluations. The functional data analysis allow to treat data as functions taking account the dependence between observations on the same individual.

In this work we proposed a method for classification and regression using a tree-based model when the covariates are functions. The algorithm follows a procedure that select the functional variable more associated with the response variable that can be continuous or categorical. This association is measured with energy test such as distance correlation test and DISCO (an extension of analysis of variance in the energy framework). This step allow us to add a statistical evaluation at each stage of the algorithm, contrary to what happens in the most known method of decision tree CART.

The application of several simulated dataset, where functional covariates can be univariate or multivariate, shows the highest performance in predictive terms compared to other classification and regression methods for functional data, this can be explained by the use of the energy test that can capture the non-linear relationship between the response variable and the functional covariates.

Keep in mind the simplicity of the output that can be understood even by people who do not have sufficient knowledge of statistical insights.

Before concluding, we want to add some considerations. It could be interesting to use a functional split instead the numeric that we used. This can lead to increase the prediction performance for the proposed method. Due to the general nature of energy test, that can be used for a wide range of variables as we need only distances between observations, it is possible to have functional or even a multivariate functional dependent variable, conditionally that we use an homogeneity measure for the dependent variable. We can build an ensemble of tree using bagging or boosting to increase the prediction performance. Further development can be obtained by

considering other complicated covariates, such as graphs, provided that we identify a split value.

Bibliography

[1] G. Aneiros-Perez and P. Vieu. Semi-functional partial linear regression. *Statistics and Probability Letters*, 76(11):1102 – 1110, 2006.

[2] Y. Araki, Konishi S., Kawano S., and H. Matsui. Functional Logistic Discrimination Via Regularized Basis Expansions. *Communications in Statistics-Theory and Methods*, 2009.

[3] S. Balakrishnan and D. Madigan. Decision trees for functional variables, 2006.

[4] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[5] L. Breiman et al. *Classification and Regression Trees*. Chapman & Hall, New York, 1984.

[6] H. Cardot, F. Ferraty, and P. Sarda. Functional linear model. *Statistics and Probability Letters*, 45(1):11–22, 1999.

[7] C. De Boor. *A Practical Guide to Splines*. Springer, 2001.

[8] M. Febrero-Bande and M.O. De La Fuente. Statistical computing in functional data analysis: The R package fda.usc. *Journal of Statistical Software*, 51(4):1–28, 2012.

[9] R.A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, (7):179–188, 1936.

[10] T. Gorecki, M. Krzysko, W. Ratajczak, and W. Wolynski. An extension of the classical distance correlation coefficient for multivariate functional data with applications. *Statistics in Transition*, 17(3):449–466, 2016.

[11] T. Hothorn, K. Hornik, and A. Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674, 2006.

[12] T. Hothorn and A. Zeileis. partykit: A modular toolkit for recursive partytioning in R. *Journal of Machine Learning Research*, 16:3905–3909, 2015.

[13] C.J. Huberty and S.F. Olejnik. Applied manova and discriminant analysis. 01 2006.

[14] G.M. James and T.J. Hastie. Functional Linear Discriminant Analysis for Irregularly Sampled Curves. *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 2001.

[15] G.V. Kass. An exploratory technique for investigating large quantities of categorical data, 1980.

[16] J.G. Liao and K.V. Chin. Logistic regression for disease classification using microarray data: model selection in a large p and small n case. *Bioinformatics*, 23(15):1945–1951, 2007.

[17] W.Y. Loh and Y.S. Shih. Split Selection Methods for Classification Trees. *Statistica Sinica*, 1997.

[18] R. Lyons. Distance covariance in metric spaces. *Ann. Probab.*, 41(5):3284–3305, 09 2013.

[19] S.H. Mousavi and H. Sorensen. Multinomial Functional Regression with Wavelets and LASSO Penalization. *Econometrics and Statistics*, 2017.

[20] H.G Muller, J.L. Wang, and J. Chiou. Functional Data Analysis. *Annual Review of Statistics and Its Application 3*, pages 257–295, 2016.

[21] J.O. Ramsey and B.W. Silverman. *Functional Data Analysis.* Springer, 2005.

[22] N. Saito. *Local feature extraction and its application using a library of bases.* PhD thesis, Yale University, 1994.

[23] N. Serban and L. Wasserman. Cats: Clustering after transformation and smoothing. *Journal of the American Statistical Association*, 100(471):990–999, 2005.

[24] J.P. Shaffer. Multiple hypothesis testing. *Annual Review of Psychology*, 46(1):561–584, 1995.

[25] J.A. Sonquist, E.L. Baker, and J.N. Morgan. Searching for Structure. 1973.

[26] H. Strasser and C. Weber. On the Asymptotic Theory of Permutation Statistics. *Mathematical Methods of Statistics*, 1999.

[27] G.J Székely and M.L. Rizzo. Testing for equal distributions in high dimension. *InterStat*, 2004.

[28] G.J. Székely and M.L. Rizzo. Disco analysis: A nonparametric extension of analysis of variance. *Ann. Appl. Stat.*, 4(2):1034–1055, 06 2010.

[29] G.J. SzéKely and M.L. Rizzo. The distance correlation t-test of independence in high dimension. *J. Multivar. Anal.*, 117:193–213, May 2013.

[30] G.J Székely and M.L. Rizzo. Energy Statistics: A Class of Statistics Based on Distances. *Journal of Statistical Planning and Inference*, 2013.

[31] G.J. Székely, M.L. Rizzo, and N.K. Bakirov. Measuring and testing dependence by correlation of distances. *Ann. Statist.*, 35(6):2769–2794, 12 2007.

[32] N. Tarabelloni, A. Arribas-Gil, F. Ieva, A.M. Paganoni, and J. Romo. *roahd: Robust Analysis of High Dimensional Data*, 2018. R package version 1.4.

[33] D. West. Neural network credit scoring models. *Computers & Operations Research*, 27(11):1131 – 1152, 2000.