Politecnico di Milano

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
MSc of Mathematical Engineering

Classification Algorithms for Multivariate Functional Data

Advisor
Prof. Anna Maria Paganoni

Co-advisor
Dr. Andrea Ghiglietti

Candidate
Andrea Martino
Matr.818993

Academic Year 2015-2016

Politecnico di Milano:
www.polimi.it

School of Industrial and Information Engineering:
www.ingindinf.polimi.it

A Mamma, Papà,
Nonna e Zia,
per avermi sostenuto
e per aver creduto in me.

Contents

List of Figures

List of Tables

Listings

Sommario

In questa tesi proponiamo una tecnica di clustering funzionale multivariato basata su una distanza che generalizza la distanza di Mahalanobis al caso di dati funzionali generati da processi stocastici. Questo nuovo strumento matematico è ben definito in $L^2(I)$, con I intervallo compatto di $\mathbb{R}$, e considera tutte le infinite componenti in cui un processo può essere proiettato, pur mantenendo le stesse idee sulle quali è basata la distanza di Mahalanobis. Nella prima parte dell'elaborato, la distanza è stata adottata in un contesto inferenziale per costruire test sulla media di processi gaussiani. Nella seconda parte, viene usata in una procedura di classificazione basata su un k-means funzionale multivariato. Per valutare la robustezza della procedura di clustering presentiamo dapprima qualche simulazione, confrontando la distanza con altre distanze note e applicandola infine a un dataset composto da segnali elettrocardiografici.

Parole chiave: Dati Funzionali, Distanze in L^2, Processi Gaussiani, Inferenza sulla media, k-means funzionale multivariato, Segnali elettrocardiografici

Abstract

In this work we propose a multivariate functional clustering technique based on a distance which is a generalization of Mahalanobis distance that extends the usual multivariate one to functional data generated by stochastic processes. This new mathematical tool is well defined in $L^2(I)$, where I is a compact interval of $\mathbb{R}$, and considers all the infinite components of data basis expansion while keeping the same ideas on which is based the Mahalanobis distance. In the first part of the work it is adopted in an inferential context to construct tests on the mean of Gaussian processes. In the second part, we use it for a k-means clustering procedure. To test the robustness of our clustering procedure we first present some simulations, comparing the distance with other known distances, and eventually applying it to a dataset of reconstructed and registered ECGs.

Keywords: Functional Data, Distances in L^2, Gaussian Processes, Inference on the mean, Functional k-means clustering, Electrocardiograph signal

Introduction

The aim of the cluster analysis is to build homogenous groups (clusters) of observations representing realizations of some random process. Clustering is often used as a preliminary step for data exploration, the goal being to identify particular patterns in data that have some convenient intepretation for the user. In particular, k-means algorithm is a clustering procedure which is based on heuristic and geometric procedures.

Over the past few decades, in many scientific fields as economics, medicine and many other domains, there has been an increasing interest towards the study of data sets whose number p of features recorded for each statistical unit is much greater than the number n of units themselves. Each observed data can be seen as a random function generated by a continuous time stochastic process $X = \{X(t),\ t \in I\}$, lying in a suitable infinite dimensional Hilbert space, typically $L^2(I)$, with I compact interval of $\mathbb{R}$. *Large p - small n problems* is the term generally used to refer to such situations.

Functional Data Analysis (FDA) represents the natural framework to develop statistical models and tools which are useful for the study of this kind of data (see Ramsay and Silverman (2002), Ramsay and Silverman (2005), Ferraty and Vieu (2006), Horváth and Kokoszka (2012)). As highlighted in this literature, a central role in this context is carried out by the Functional Principal Component Analysis (FPCA), which is based on the Karhunen-Loève (KL) expansion, that decomposes a random function $X(t)$ in a sum of the mean $m(t)$ and a series of orthonormal functions $\vartheta_k(t)$, each one multiplied by zero-mean uncorrelated random variables $\sqrt{\lambda}_k Z_k$, where $(\lambda_k)_k$ are the eigenvalues of the covariance operator V of X while the orthonormal basis $(\vartheta_k)_k$ is composed by the eigenfunctions. As discussed in Benko *et al.* (2009), the analysis of the principal components seems to be more important in the functional context than in the

multivariate framework. In fact, it is one of the feasible ways to reduce data dimensionality. Moreover, in FPCA the principal components are intepreteted as the modes of variation of $X(t)$ along t, which is much more interpretable than in multivariate PCA. When the goal of the analysis consists in describing the shape of $X(t)$, the first K principal components $\vartheta_k, k = 1, \ldots, K$, usually contain all the information needed to represent data. Nevertheless, when the goal consists in making inference on infinite dimensional objects, as the mean function $m(t)$ or classifying curves in different groups, considering a fixed number of components may lead to lose some information on the distribution of $X(t)$ and to provide meaningless results.

Despite of the great interest in the FPCA, many inferential procedures adopted in the multivariate PCA have not been extended yet to the functional case. For example, in the multivariate context the inference on the mean is typically based on the Mahalanobis distance, since it is the best way to measure the distance between elements because it takes into account the correlation among the variables and their variability. However, when data belongs to an infinite dimensional space, as $L^2(I)$, the Mahalanobis distance is not generally well defined and the inference is usually realized by considering only the first K principal components. This is an approach which is pretty far from the idea of the Mahalanobis distance, which weights the distances along all the components with the inverse of their variability. In this work, we use a generalization of Mahalanobis distance that extends the usual multivariate one to functional data generated by stochastic processes. This new metric has been obtained after noticing a quite unconventional way to derive the classical Mahalnobis distance.

Clustering functional data generally can be a difficult task because of the infinite dimensional space that data belong to. The lack of a definition for the probability density of a functional random variable, the definition of distances or estimation from noisy data are some examples of such difficulties. Different approaches have been proposed along years. The most popular approach consists of reducing the infinite dimensional problem to a finite one by approximating data with elements from some finite dimensional space. Then, clustering algorithms for finite dimensional data can be performed. On the other hand, there have been defined nonparametric methods for clustering which consist generally in defining specific distances or dissimilarities for functional data and then apply clustering algorithms.

In this work, we perform a k-means algorithm clustering procedure by using a generalization of Mahalanobis distance. There are many different implementations of the functional k-means algorithm in the literature on functional data analysis, among which some procedures integrate registration in the classification steps (e.g. the k-means alignment algorithm that is described in Sangalli *et al.* (2010), the core shape modeling approach in Boudaoud *et al.* (2010), the non-parametric time-synchronized iterative mean updating technique in Liu and Müller (2003) or the simultaneously aligning and cluster K-centres model in Liu and Yang (2009)). However, in our real case study, the clustering procedure is performed on data that have been already registered.

Outline

In the first chapter we present all the theoretical background of our work, starting from Functional Data Analysis and all its related tools. Then we present all the distances used in our work, with a particular focus on the new distance named d_p. We show that it is well defined in $L^2(I)$ and achieves both the goals of considering all the infinite components of data basis expansion and of keeping the same ideas on which is based the Mahalanobis distance.

In the second chapter the new distance is adopted in an inferential context to construct tests on the mean of Gaussian processes for one and two populations. The tests are constructed assuming the covariance structure to be either known or unknown. The distance d_p which we adopt is tuned by a parameter that smoothly determines how to weight the contributions along all the fininite components of $L^2(I)$. As we will see in the simulations in the last part of the chapter, the parameter p plays a crucial role in the power of the tests.

In the third chapter we perform the multivariate functional k-means procedure based on the d_p distance for some values of p and we compare the results with the ones obtained using both the classical L^2-distance and a truncated version of the Mahalanobis distance. We present some simulation studies both in the univariate and multivariate functional case, showing that the results mainly depend on

the choice of the parameter p. In the last part of the chapter, the clustering procedure is carried out on a real dataset composed by a 8-dimensional sample of electrocardiographic (ECG) traces; hence, we discuss how the d_p distance works in comparison with the others, and proposing a new semi-automatic diagnostic procedure, based only on the ECG morphology, which could help in an early detection of heart failure.

All the analysis of our work have been carried out using the software R.

Chapter 1

Functional Data

This first chapter aims at introducing the theoretical background, starting from Functional Data Analysis and all the related tools with a particular focus on some distances which will be used to perform a functional k-means clustering, particularly the Generalized Mahalanobis Distance. We will introduce a new metric defined in the infinite dimensional space of the square integrable and differentiable real functions, defined on a compact interval of $\mathbb{R}$, that generalizes the Mahalanobis distance used in the multivariate context.

1.1 Functional Data Analysis

1.1.1 Functional Smoothing

One of the first steps in FDA consists in the functional smoothing, which is used for turning raw discrete data into smooth functions. Functional data are usually observed and recorded discretly as n pairs (t_j, y_j), where y_j is the value of the function x at time t_j. This procedure is needed because observed data are generally affected by the presence of observational error or a noise superimposed on the underlying signal by aspects of the measurement process. This assumption can be expressed formally as follows:

$$y_j = x(t_j) + \varepsilon_j, \quad j = 1, \ldots, n \tag{1.1}$$

where ε_j is the exogenous term which adds roughness to the raw data. One of the tasks in representing the raw data as functions consists in

the attempt of filtering out the noise as efficiently as possible. In this procedure, it is important to obtain a function that is smooth, in the sense of possessing a certain number of derivatives, that reproduces well the details of the observations but without the use of interpolation procedures, which would retain too much noise.

One of the main smoothing procedures is the representation by basis functions. A basic function system is a set of known functions ϕ_k that are mathematically independent of each other and that can approximate arbitrarily well any function x by taking a linear combination of a sufficiently large number of these functions. Basis function procedure represent a function x by a linear expansion

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t).$$

in terms of a certain number K of known basis functions ϕ_k. The best known basis expansion is perhaps provided by the *Fourier series*:

$$\hat{x}(t) = c_0 + c_1 \sin(\omega t) + c_2 \cos(\omega t) + c_3 \sin(2\omega t) + c_4 \cos(2\omega t) + \dots$$

defined by the basis $\phi_0(t) = 1, \quad \phi_{2r-1}(t) = sin(r\omega t), \quad \phi_{2r}(t) = cos(r\omega t)$, $r = 1, 2, \dots$ This basis is periodic and the parameter ω determines the period $2\pi/\omega$. If the values of t_j are equally spaced on the interval I and the period is equal to the length of interval I, then the basis is orthogonal. The Fast Fourier transform (FFT) makes it possible to find all the coefficients extremely efficiently when n is a power of 2 and the arguments are equally spaced in $O(n \log n)$ operations. For this reason, the Fourier series has been the traditional basis of choice for long time series, but newer techniques such as B-splines and wavelets can exceed this computational efficiency.

For what concerns the B-spline method, the first step in defining a spline is to divide the interval over which a function is to be approximated into J subintervals separated by values $\tau_j, j = 1, \dots, J-1$ that are called *knots*. Over each interval, a spline is a polynomial of specified order $m = n + 1$, with n order of the B-spline. The B-spline is then a continuous functions at the knots and its derivatives are also continuous up to the derivative of degree $n-1$. The degrees of freedom can be defined as d.o.f.$=m+J-1$. To construct a spline function, we have to specify a system of basis functions $\phi_k(t)$ which is itself a spline function. The most popular B-spline basis system has been developed by de Boor in 2001 and is of great importance

for efficient computation, thanks to the *compact support* property according to which an order m B-spline basis function is positive over no more than m intervals. A spline function $S(t)$ is then defined as

$$S(t) = \sum_{k=1}^{m+J-1} c_k B_k(t, \tau)$$

where $B(t, \tau)$ is the value at t of the $k-th$ B-spline basis function defined by the knot sequence τ with its coefficient c_k.
It is of great importance to know how to position the τ_j knots. As long as the data are equally spaced, it is good to choose an equal spacing. If they are not, it is maybe wiser to place a knot at every jth data point, where j is a number fixed in advance, such that it is possible to catch more characteristics of the function. Even though the two described smoothing procedures are very used, for the data which will be studied in the last chapter, in Indino (2015) it has been used the wavelet smoothing procedure, since it has given the best results.

Before introducing the smoothing procedure based on wavelet functions, we analyze the representation method of a function with orthonormal wavelet basis (see Pigoli and Sangalli (2012), Donoho and Johnstone (1995)). These basis are generated by dilations and translations of a single scaling function $\varphi(x)$ called *father wavelet*. We call V_0 the *reference* space generated by translations of the father wavelet $\varphi_{0,h} = \varphi(x-h)$, where h is an integer. The dilations $\varphi_{1,h} = \sqrt{2}\varphi(2x-h)$ form an orthonormal basis for the space $V_1 \supset V_0$. More generally, we have

$$\cdots \supset V_1 \supset V_0 \supset V_{-1} \supset \cdots$$

where each V_j is spanned by $\varphi_{j,k} = 2^{j/2}\varphi(2^j x - h)$. A function in V_{j+1} can be represented by a component in V_j plus the component in the orthogonal complement W_j of V_j to V_{j+1}, written as $V_{j+1} = V_j \oplus W_j$. The functions $\psi_(x-h)$ generated by the *mother wavelet* $\psi(x) = \varphi(2x) - \varphi(2x-1)$ form an orthonormal basis for W_0. Likewise $\psi_{j,k} = 2^{j/2}\psi(2^j x - h)$ form a basis for W_j. Now we can notice that $V_{j+1} = V_j \oplus W_j = V_{j-1} \oplus W_{j-1} \oplus W_j$, so we can write $V_J = V_0 \oplus W_0 \oplus W_1 \cdots \oplus W_{J-1}$. Since these spaces are orthogonal, all the basis functions are orthonormal. In fact, if the domain is discrete with 2^J time points, this is as far as we can go. At level j there are 2^j basis and adding up in W_J we have a total of $2^J - 1$ elements in W_j and

one in V_0. This structured orthonormal basis allows for a *multiresolution analysis*.

The procedure described above defines the *Discrete Wavelet Transform* (DWT) which is the first step of the smoothing procedure. The second and last step consists in the thresholding of the wavelet coefficients, which can be realized using several methods, as for example Universal threshold, SURE (Stein Unbiased Risk Estimator), Cross-validation threshold,...

Let us recall the assumption (1.1), then we can write the wavelet transform as follows:

$$d_i^* = d_i + \varepsilon_i$$

where $d^* = Wy, d = Wx$ and $\varepsilon = We$, where W is the orthonormal wavelet basis matrix evaluated at uniformly spaced observations. Thus, the greatest coefficients d_i^* represent the original signal with an added noise while the coefficients close to 0 only represent the noise.

1.1.2 Functional Registration

The values of two or more observations $x_i(t_j), i = 1, \ldots, n$ and $j = 1, \ldots, p$ can in principle differ because of two types of variation. The first one is the vertical variation, or *amplitude variation*, which is the difference between the function values at time points, although they have the same shape features at that time. However they may also exhibit *phase variation* in the sense that the value functions should not be compared at the same time t because they are not exhibiting the same behaviour. The procedure which reduces the phase variation without considering the amplitude variation is called *functional registration*. There are several registration methods, but we will be interested in the landmark registration since the dataset which will be used in this work naturally leads to this choice. A landmark of a curve is some feature that can be associated with a specific argument value t. These are typically maxima, minima or zero crossings of curves. The landmark registration process requires for each curve x_i the identification of the argument values $t_{il}, l = 1, ..., L$, associated with each of L landmarks. The goal is to construct a transformation h_i for each curve such that the registered curves with values

$$x^*(t) = x_i[h_i(t)]$$

have more or less identical argument values for any given landmark. The transformation h_i is called *time-warping function* of the i-th curve and it has to satisfy the following properties:

1. $h_i(0) = 0 \quad \forall i = 1, \ldots, n$
2. $h_i(T) = T \quad \forall i = 1, \ldots, n$
3. $h_i(t_{il}) = \bar{t}_l, \quad l = 1, ..., L \quad \forall i$ ($\bar{t}_l$ mean value of the landmarks over all the statistical units)
4. h_i strictly monotonic: $s < t$ implies that $h_i(s) < h_i(t)$.

1.1.3 Functional Principal Component Analysis

After the preliminary steps of smoothing, registering and displaying the data, when we have functional data we want to explore them to see the features characterizing typical functions. Some indications of complexity of the data is also required, in the sense of how many types of curves and characteristics are to be found. Functional principal component analysis (FPCA) has proved to be a key technique for the study of functional data. The basic tool which is used in the present work to perform the FPCA is the Karhunen-Loève (KL) expansion.

Let us consider a process X, with mean function $m(t) = \mathbb{E}[X(t)]$ and covariance operator V, i.e. V is a linear compact integral operator from $L^2(I)$ to $L^2(I)$ such that $(Va)(s) = \int_I v(s,t)a(t)dt \quad \forall a \in L^2(I)$, where v is the covariance function defined as $v = \mathbb{E}[(X(t) - m(t))(X(s) - m(s))]$. Then, denote with $\{\lambda_k, k \geq 1\}$ and $\{\vartheta_k, k \geq 1\}$ respectively the sequences of eigenvalues and eigenfunctions of v. The Karhunen-Loève expansion decomposes the process $X(t)$ in a sum of its mean $m(t)$ and the series of orthonormal functions $\vartheta_k(t)$, each one multiplied by zero-mean uncorrelated random variables $\sqrt{\lambda_k}Z_k$, $(\lambda_k > 0, \mathbb{V}ar(Z_k) = 1)$. Then we can write

$$X(t) = m(t) + \sum_{k=1}^{\infty} Z_k \sqrt{\lambda_k} \cdot \vartheta_k(t)$$

The orthonormal basis $(\vartheta_k)_k$ is composed by the eigenfunctions of the covariance operator V of X, while the coefficient variances $(\lambda_k)_k$ are its eigenvalues. The dynamic of the random function $(X(t) - m(t))$ can be fully described by the eigenstructure of V and the distribution of the

sequence $(Z_k)_k$, which will be considered in this work as i.i.d. standard normal variables.

1.2 Distances

In this section we present the main distances which are considered in this work, along with some preliminary concepts.

Definition 1.1. A **distance** on a given set E is a function $d : E \times E \to [0, \infty)$ such that for all $x, y, z \in E$ the following conditions are satisfied:

1. $d(x, y) \geq 0$ (non-negativity or separation axiom);
2. $d(x, y) = 0 \Leftrightarrow x = y$ (identity of indiscernibles);
3. $d(x, y) = d(y, x)$ (symmetry);
4. $d(x, z) \leq d(x, y) + d(y, z)$ (subadditivity or triangle inequality).

Definition 1.2. Let $(I, \mathcal{M}, \mu)$ be a given measurable space. $L^2(I)$ is the set of equivalence classes of measureable functions with respect to μ, $X : I \to \mathbb{R}$, such that

$$\int_R |X|^2 d\mu < \infty.$$

Definition 1.3. Given a compact interval I of $\mathbb{R}$, the L^2-distance between $X, Y \in L^2(I)$ is defined by

$$d_{L^2}(X, Y) = \sqrt{\int_I (X(t) - Y(t))^2 dt}$$

Our focus will be now on the usual Mahalanobis Distance in the multivariate framework, before generalizing it in the functional framework.

1.2.1 Mahalanobis distance in the multivariate framework

Let us consider a finite dimensional framework. Let $\mathbf{X} \in \mathbb{R}^K, K \in \mathbb{N}$, be a random vector with mean $\mathbf{m} \in \mathbb{R}^K$ and covariance matrix $V \in \mathbb{R}^{K \times K}$, with the eigenvectors $\boldsymbol{\vartheta}_1, ..., \boldsymbol{\vartheta}_K$ and the eigenvalues $\lambda_1, ..., \lambda_K$ such that $\lambda_1 \geq ... \geq \lambda_K > 0$. Given a couple of vectors $(\mathbf{u}, \mathbf{v})$ of elements of $\mathbb{R}^K$, the

usual metric is the L^2-distance $d_{L^2}(u,v) = \sqrt{\sum_{k=1}^{K}((\mathbf{u}-\mathbf{v})^T\boldsymbol{\vartheta}_k)^2}$, where $\mathbf{a}^T\mathbf{b} = \sum_{k=1}^{K} a_i b_i$ is the inner product in $\mathbb{R}^K$. However, when $\mathbf{u}$ and $\mathbf{v}$ are realization of $\mathbf{X}$, the Mahalanobis distance, i.e.

$$d_M(\mathbf{u},\mathbf{v}) = \sqrt{\sum_{k=1}^{K}\frac{((\mathbf{u}-\mathbf{v})^T\boldsymbol{\vartheta}_k)^2}{\lambda_k}},$$

could be more useful since it considers the correlations and the variability described by the covariance structure of $\mathbf{X}$. Now we highlight an interesting relation between these two distances. Let us introduce the function $f(\cdot;\mathbf{u},\mathbf{v}) : \mathbb{R}^+ \to \mathbb{R}^+$ defined as follows

$$f(c;\mathbf{u},\mathbf{v}) = \sum_{k=1}^{K} f_k(c;\mathbf{u},\mathbf{v}) = \sum_{k=1}^{K}((\mathbf{u}-\mathbf{v})^T\boldsymbol{\vartheta}_k)^2 \cdot \exp(-\lambda_k c).$$

For any fixed $c \in \mathbb{R}^+$, $\sqrt{f(c;\mathbf{u},\mathbf{v})}$ is a distance; in fact, it can be seen as the distance between the vectors $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$ which can be written as

$$\tilde{\mathbf{u}} := \sum_{k=1}^{K}((\mathbf{u}^T\boldsymbol{\vartheta}_k \exp(-\lambda_k c/2)) \cdot \boldsymbol{\vartheta}_k$$
$$\tilde{\mathbf{v}} := \sum_{k=1}^{K}((\mathbf{v}^T\boldsymbol{\vartheta}_k \exp(-\lambda_k c/2)) \cdot \boldsymbol{\vartheta}_k.$$

Moreover, we can remark three properties for the function f:

1. $f(c;\mathbf{u},\mathbf{v})$ is non increasing in c;
2. $\lim_{c\to\infty} f_k(c;\mathbf{u},\mathbf{v}) = 0$, $\forall\lambda_k > 0$;
3. $f(0;\mathbf{u},\mathbf{v}) = d^2_{L^2}(\mathbf{u},\mathbf{v})$.

Then for any fixed $c \in \mathbb{R}^+$, $\sqrt{f(c;\mathbf{u},\mathbf{v})}$ represents a distance which is less or equal to the L^2-distance between $\mathbf{u}$ and $\mathbf{v}$. In particular, $f_k(c;\mathbf{u},\mathbf{v})$ is the contribution to this distance along the component $\boldsymbol{\vartheta}_k$. As c increases the term $f_k(c;\mathbf{u},\mathbf{v})$ gets smaller. The decreasing rate of $f_k(c;\mathbf{u},\mathbf{v})$ is ruled by λ_k: the greater it is, the faster $f_k(c;\mathbf{u},\mathbf{v})$ vanishes. The term $\exp(-\lambda_k c)$ penalizes the contribution of the L^2-distance along the k-th component and this penalization is strong for high values of λ_k and irrelevant for low values of λ_k.
In the Mahalanobis distance, a similar behaviour is obtained by rescaling

with respect to λ_k; the components with low λ_k become stronger while the components with high λ_k become weaker. Then, integrating the function $f_k(c; \mathbf{u}, \mathbf{v})$ over c, we can measure how fast it vanishes:

$$\int_0^\infty f_k(c; \mathbf{u}, \mathbf{v})dc = \frac{((\mathbf{u}-\mathbf{v})^T \boldsymbol{\vartheta}_k)^2}{\lambda_k}$$

The Mahalanobis distance can be seen as the square root of the area below the function $f(c; \mathbf{u}, \mathbf{v})$:

$$d_M(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_{k=1}^{K} \frac{((\mathbf{u}-\mathbf{v})^T \boldsymbol{\vartheta}_k)^2}{\lambda_k}} = \sqrt{\int_0^\infty f(c; \mathbf{u}, \mathbf{v})dc} \tag{1.2}$$

1.2.2 Generalized Mahalanobis Distance in the functional framework

Now we want to extend these ideas to a functional framework. Specifically we present the generalized Mahalanobis distance in infinite dimensional spaces as it has been proposed in Ghiglietti and Paganoni, (2014). We consider now the realizations y and w of a stochastic process $X \in L^2(I)$, where I is a compact interval of $\mathbb{R}$. Let $\langle a, b\rangle = \int_I a(t)b(t)dt$ be the usual inner product in $L^2(I)$, so the natural generalization of the Mahalanobis distance in the functional framework would be

$$d_M(y, w) = \sqrt{\sum_{k=1}^{\infty} \frac{(\langle y-w, \vartheta_k)^2}{\lambda_k}}.$$

It is well known that, since it could be undefined because the series can diverge for some $y, w \in L^2(I)$, d_M is not a proper distance in $L^2(I)$. For this reason, the Mahalanobis distance is usually considered in its truncated version, fixing an integer $K \in \mathbb{N}$ and summing up only the first K components, so that we can write:

$$d_M(y, w) = \sqrt{\sum_{k=1}^{3} \frac{(\langle y-w, \vartheta_k)^2}{\lambda_k}}.$$

However, when this approach is used to measure the entire space $L^2(I)$, we can remark two main drawbacks:

1. the contribution given by the projections in the space orthogonal to $\vartheta_1, ..., \vartheta_K$ is not considered in the distance; then, for any choice of K, we may have $y, w \in L^2(I)$ such that the truncated Mahalanobis distance is arbitrarily small and the L^2-distance is arbitrarily large, which seems unresonable;

2. all the contributions of the $L^2(I)$ distance are basically multipled by $1/\lambda_k \cdot \mathbf{1}_{\{\lambda_k \geq \lambda_K\}}$ which is not decreasing in λ_k. This is incoherent with the idea of penalizing the $L^2(I)$ distance with a term that is inversely proportional to the size of the corresponding eigenvalue λ_k.

Now our goal is to solve these two problems, extending the representation (1.2) of the Mahalanobis distance to the functional framework. For any couple of elements of $L^2(I)$ (y, w) we can define the function $f(\cdot; y, w)$ as

$$f(c; y, w) = \sum_{k=1}^{\infty} (\langle y - w, \vartheta_k \rangle)^2 \cdot \exp(-\lambda_k c).$$

The series is finite for any choice of $c \in \mathbb{R}^+$. As explained before, the Mahalanobis distance can be computed as $\sqrt{\int_0^\infty f(c; y, w) dc}$. However, in the infinite dimensional case, when the Mahalanobis distance is not defined the function $f(c; y, w)$ is not integrable in $\mathbb{R}^+$. To deal with this case, we introduce a function $g(\cdot; p)$, tuned by a parameter $p > 0$ such that $\int_0^\infty g(c; p) dc < \infty$. In this way, it is ensured that the function $f(c; y, w) \cdot g(c; p)$ is integrable for any $p > 0$, in fact

$$\begin{aligned}
\int_0^\infty f(c; y, w) \cdot g(c; p) dc &= \int_0^\infty \sum_{k=1}^{\infty} (\langle y - w, \vartheta_k \rangle)^2 \cdot \exp(-\lambda_k c) \cdot g(c; p) dc \\
&\leq \int_0^\infty \sum_{k=1}^{\infty} (\langle y - w, \vartheta_k \rangle)^2 \cdot g(c; p) dc \\
&= \left(\int_0^\infty g(c; p) dc \right) \cdot \sum_{k=1}^{\infty} (\langle y - w, \vartheta_k \rangle)^2 \\
&= \left(\int_0^\infty g(c; p) dc \right) \cdot d^2_{L^2}(y, w) < \infty
\end{aligned}$$

Without loss of generality we choose g such that $\int_0^\infty g(c;p)dc = p$. We are now able to construct a distance defined as

$$d_p(y,w) := \sqrt{\int_0^\infty f(c;y,w)\cdot g(c;p)dc} = \\ = \sqrt{\int_0^\infty \sum_{k=1}^\infty (\langle y-w,\vartheta_k\rangle)^2 \cdot \exp(-\lambda_k c)\cdot g(c;p)dc} \tag{1.3}$$

that is finite for any couple of functions y and w. We can prove that $d_p(y,w)$ is a distance highlighting that $d_p(y,w)$ is the L^2 - distance between the following two elements:

$$\tilde{y} = \sum_{k=0}^\infty \langle y,\vartheta_k\rangle \cdot \sqrt{\int_0^\infty \exp(-\lambda_k c)\cdot g(c;p)dc}\cdot\vartheta_k$$
$$\tilde{w} = \sum_{k=0}^\infty \langle w,\vartheta_k\rangle \sqrt{\int_0^\infty \exp(-\lambda_k c)\cdot g(c;p)dc}\cdot\vartheta_k$$

where $\tilde{y}$ and $\tilde{w}$ are elements of $L^2(I)$ since $\int_0^\infty \exp(-\lambda_k c)\cdot g(c;p)dc \leq \int_0^\infty g(c;p)dc < \infty$. If the function g is such that for any $p > 0$

- $g(c;p)$ is a non increasing and non negative function in c,
- $g(0;p) = 1$,

the two functions $f\cdot g$ and f satisfy the same properties; then $f(0;y,w)\cdot g(0;p) = d^2_{L^2(I)}(y,w)$ and, for any fixed $c\in\mathbb{R}^+$, $\sqrt{f(c;y,w)\cdot g(c;p)}$ is a distance between the elements y and w dominated by the L^2-distance. Moreover, for any fixed $c\in\mathbb{R}^+$, we assume that g satisfies the following properties:

- $g(c;p)$ is a non decreasing and non negative function of p;
- $\lim_{p\to\infty} g(c;p) = 1$.

As a consequence, the greater the value of p, the greater the distance $d_p(y,w)$. Moreover, even if $d_p(y,w)$ is finite for any couple of functions y and w, when p goes to infinity, $d_p(y,w)$ can diverge since it tends to the Mahalanobis distance that can diverge in the infinite dimensional case. We need now to write the distance $d_p(y,w)$ in a more suitable and useful way; in order to do so, we define the function

$$h_k(p) := \int_0^\infty \lambda_k \exp(-\lambda_k c)\cdot g(c;p)dc$$

and so we obtain

$$d_p(y,w) = \sqrt{\sum_{k=1}^{\infty} d^2_{M,k}(y,w) \cdot h_k(p)} \tag{1.4}$$

where $d_{M,k}(y,w) = \sqrt{(\langle y-w, \vartheta_k\rangle)^2/\lambda_k}$ is the term representing the contribution of the Mahalanobis distance along the k^{th} component.

Finally, we can prove the following results:

- $\lim_{k\to\infty} h_k(p) = 0$;
- $\lim_{p\to\infty} h_k(p) = 1$ for any k such that $\lambda_k > 0$.

In fact,

$$\begin{aligned}\lim_{k\to\infty} h_k(p) &= \lim_{k\to\infty} \int_0^\infty \lambda_k \exp(-\lambda_k c)\cdot g(c;p)dc \\ &= \left(\lim_{k\to\infty}\lambda_k\right)\cdot\left(\int_0^\infty \lim_{k\to\infty}\exp(-\lambda_k c)\right)\cdot g(c;p)dc \\ &= \left(\lim_{k\to\infty}\lambda_k\right)\cdot\left(\int_0^\infty g(c;p)dc\right) = 0\end{aligned}$$

and

$$\begin{aligned}\lim_{p\to\infty} h_k(p) &= \lim_{p\to\infty} \int_0^\infty \lambda_k \exp(-\lambda_k c)\cdot g(c;p)dc \\ &= \int_0^\infty \lambda_k \exp(-\lambda_k c)\cdot\left(\lim_{p\to\infty} g(c;p)\right) \\ &= \int_0^\infty \lambda_k \exp(-\lambda_k c)dc = 1\end{aligned}$$

Moreover, the distance d_p and the usual distance in L^2 (d_{L^2}) are equivalent since, for any $y, w \in L^2(I)$

$$\left(\frac{h_1(p)}{\lambda_1}\right) d_{L^2}(y,w) \le d_p(y,w) \le p \cdot d_{L^2}(y,w)$$

Among the several ways to choose g, some meaningful examples are:

- $g(c;p) = \mathbf{1}_{\{c\le p\}}$. In this case $h_k(p) = (1-\exp(-\lambda_k p))$;
- $g(c;p) = \exp(-c/p)$. In this case $h_k(p) = \lambda_k/(\lambda_k + 1/p)$.

Remark 1.4. It is worth noting that the generalized Mahalanobis distance d_p can be also written for multivariate random functions $\boldsymbol{X} = (X_1, .., X_h)^T$, with $h \geq 2$, where $X_i \in L^2(I)$ for any $i \in \{1, .., h\}$, with I compact interval of $\mathbb{R}$. In that case, the mean $\boldsymbol{m} = \mathbb{E}[\boldsymbol{X}]$ is defined as a vector of functions in $L^2(I)$ such that $m_l = \mathbb{E}[X_l]$ for any $l = \{1, .., h\}$, and the covariance kernel $v(s,t) = \boldsymbol{Cov}\left[\boldsymbol{X(s)} \otimes \boldsymbol{X(t)}\right]$ is defined as a $h \times h$ matrix of functions such that $v_{l_1 l_2}(s,t) := \boldsymbol{Cov}\left[X_{l_1}(s), X_{l_2}(t)\right]$ for any $l_1, l_2 = \{1, .., h\}$. The scalar product between two elements y and w of $L^2(I)$ with values in $\mathbb{R}^h$ is defined as follows

$$\langle y, w \rangle = \sum_{l=1}^{h} \int_T y_l(t) w_l(t) dt.$$

The eigenvalues $\{\lambda_{kl}, k \geq 1, l = 1, \ldots, h\}$ and the eigenfunctions $\{\vartheta_{kl}, k \geq 1, l = 1, \ldots, h\}$ of v are the elements solving $\sum_{l_2}^{h} \langle v_{l_1 l_2}, \vartheta_{k l_2} \rangle = \lambda_k \vartheta_{k l_1}$ for any $l_1 = \{1, .., h\}$. The generalized Mahalanobis distance d_p can be defined as in (1.4) using these quantities and the contribution of the Mahalanobis distance along the k^{th} component is now the following:

$$d_{M,k}(y, w) = \sqrt{\sum_{l=1}^{h} \frac{\int_T (y_l(t) - w_l(t)) \cdot \vartheta_{kl}(t) dt}{\lambda_{kl}}}.$$

The results which will be presented hold as well for multivariate random functions.

We will now focus our work on using and comparing the distance d_p in a statistical inference framework.

Chapter 2

Inference on the mean of a Gaussian process

Before showing the results obtained with the functional k-means, we construct testing procedures on the means of Gaussian processes, discussing their statistical properties. We will propose critical regions based on the generalized Mahalanobis distance d_p presented in Subsection 1.2.2.

2.1 Inference on the mean of a Gaussian process with known covariance function

Let $X_1, \ldots, X_L$ be $L \geq 1$ Gaussian processes with probability laws $P_{X_1}, \ldots, P_{X_L}$, and denote with $m_1, \ldots, m_L \in L^2(I)$ the corresponding means. Assume that $P_{X_1}, \ldots, P_{X_L}$ have the same covariance function v, and denote with $(\lambda_k)_k$ the sorted eigenvalues in ascending order of v, with $(\vartheta_k)_k$ the associated eigenfunctions. For any $l = 1, \ldots, L$, let $X_{1,l}, \ldots, X_{n_l,l}$ be n_l i.i.d. realizations of P_{X_l}, and denote with $\bar{X}_l$ the pointwise sample mean: $(X_{1,l} + \ldots + Xn_l, l)/n_l$. In this section we propose critical regions associated to the following inferential problems:

1. testing the mean of a Gaussian process against an arbitrary function in $L^2(I)$;

2. comparing the means of two Gaussian processes with the same covariance function.

For a deeper study of these problems, see Ghiglietti and Paganoni (2014).

Part 1:
Fix $l = 1, \dots, L$, $m_0 \in L^2(I)$ and consider the following hypotesis test

$$H_0 : m_l = m_0 \qquad \text{vs} \qquad H_1 : m_l \neq m_0 \tag{2.1}$$

We want to construct a critical region of level α for test (2.1), so we consider the Karhunen-Loève decomposition of $X_{i,l}, i = 1, \dots, n_l$:

$$X_{i,l}(t) = m_l(t) + \sum_{k=1}^{\infty} Z_{ki,l}\sqrt{\lambda_k}\vartheta_k(t),$$

where $(Z_{ki,l})_{k=1}^{\infty}$ is a sequence of independent standard normal variables, since P_{X_l} is Gaussian. We have that

$$\left(\sqrt{n_l} \cdot \frac{\langle \bar{X}_l - m_l, \vartheta_k \rangle}{\sqrt{\lambda_k}}\right)_k = \left(\frac{1}{\sqrt{n_l}} \sum_{i=1}^{n_l} Z_{ki,l}\right)_k \overset{i.i.d.}{\sim} \mathcal{N}(0,1).$$

Hence, the sequence $(n_l \cdot d^2_{M,k}(\bar{X}_l))_k$ is composed by i.i.d. chi-squared random variables with 1 degree of freedom, so that

$$n_l \cdot d_p^2(\bar{X}, m_l) = n_l \cdot \sum_{k=1}^{\infty} d^2_{M,k}(\bar{X}_l, m_l) h_k(p) \sim \sum_{k=1}^{\infty} \chi^2_{1,k} h_k(p)$$

where $(\chi^2_{1,k})_{k=1}^{\infty}$ is a sequence of i.i.d. chi-squared variables with 1 degree of freedom. We denote with $\xi^2_{\alpha,p}$ the 1-α quantile of the distribution of $\sum_{k=1}^{\infty} \chi^2_{1,k} h_k(p)$. Then, we construct the following critical region of level α

$$R^1_\alpha = \{n_l \cdot d_p^2(\bar{X}_l, m_0) > \xi^2_{\alpha,p}\} \tag{2.2}$$

for any $p > 0$ and $n_l \in \mathbb{N}$. The quantiles $\xi^2_{\alpha,p}$ can be obtained from the distribution of $\sum_{k=1}^{\infty} \chi^2_{1,k} h_k(p)$ computed in simulation, noting that the probability law of $\sum_{k=1}^{\infty} \chi^2_{1,k} h_k(p)$ depends on the whole sequnce of eigenvalues $(\lambda_k)_k$, on the choice of g and on the parameter p.
We now investigate the power of test (2.1) based on the critical region (2.2). First, we can note that under the alternative hypotesis H_1 we can write:

$$\begin{aligned} d^2_{M,k}(\bar{X}_l, m_0) &= \left(\frac{\langle \bar{X}_l - m_0, \vartheta_k \rangle}{\sqrt{\lambda_k}}\right) \\ &= \left(\frac{\langle \bar{X}_l - m_l, \vartheta_k \rangle}{\sqrt{\lambda_k}} + \frac{\langle m_l - m_0, \vartheta_k \rangle}{\sqrt{\lambda_k}}\right)^2 \sim \chi^2_{1,k}(\nu_k) \end{aligned}$$

where $\chi^2_{1,k}(\nu_k)$ is a non-central chi-squared with 1 degree of freedom and $\nu_k = d^2_{M,k}(m_l, m_0)$ is the non centrality parameter. Hence, we have that

$$n_l \cdot d^2_p(\bar{X}_l, m_0) \sim \sum_{k=1}^{\infty} \chi^2_{1,k}(\nu_k) h_k(p), \qquad \nu_k = n_l \cdot d^2_{M,k}(m_l, m_0)$$

where $(\chi^2_{1,k}(\nu_k))_{k=1}^{\infty}$ are independent. Then, the power of (2.2) can be obtained as

$$\begin{aligned}
\beta_1 &= P_{m_l \neq m_0}(R^1_\alpha) \\
&= P_{m_l \neq m_0}(n_l \cdot d^2_p(\bar{X}_l, m_0) > \xi^2_{\alpha,p}) \\
&= P_{m_l \neq m_0}\left(\sum_{k=1}^{\infty} \chi^2_{1,k}(\nu_k) h_k(p) > \xi^2_{\alpha,p} \right)
\end{aligned}$$

with $\nu_k = n_l \cdot d^2_{M,k}(m_l, m_0)$. We can note that the power β_1 tends to one as n_l increases.

Remark 2.1. It is worth noting that this test does not have a low power when m_l and m_0 are arbitrarily distant in $L^2(I)$. This problem occurs when the inference is computed only with the first components. To see this, consider the quantity $d^2_p(\bar{X}_l, m_0)$ used to compute the power β_1 of the test (2.2). It is possible to show that

$$\mathbb{E}[d^2_p(\bar{X}_l, m_0)] = \sum_{k=1}^{\infty} \left[\frac{1}{n_l} + \left(\frac{\langle m_l - m_0, \vartheta_k \rangle}{\sqrt{\lambda_k}} \right)^2 \right] h_k(p) \geq C_1 \cdot d^2_{L^2}(m_l, m_0),$$

$$\begin{aligned}
\mathbb{V}ar(d^2_p(\bar{X}_l, m_0)) &= \sum_{k=1}^{\infty} \left[\frac{2}{n_l^2} + \frac{4}{n_l} \left(\frac{\langle m_l - m_0, \vartheta_k \rangle}{\sqrt{\lambda_k}} \right)^2 \right] h_k(p)^2 \\
&\leq C_2 \cdot (1 + p \cdot d^2_{L^2}(m_l, m_0)),
\end{aligned}$$

where $C_1, C_2 > 0$ are constants independent of m_l and m_0. From this result, we have that $\mathbb{E}[d^2_p(\bar{X}_l, m_0)] / \sqrt{\mathbb{V}ar(d^2_p(\bar{X}_l, m_0))}$ goes to infinity as $d^2_{L^2}(m_l, m_0)$ diverges. Hence, the power β_1 of the test (2.2) tends to one when $d^2_{L^2}(m_l, m_0)$ increases.

Part 2:

Fix $l_1, l_2 = 1, \ldots, L, \quad (L \geq 2), l_1 \neq l_2$ and consider the following hypotesis test

$$H_0 : m_{l_1} = m_{l_2} \qquad \text{vs} \qquad H_1 : m_{l_1} \neq m_{l_2}. \tag{2.3}$$

The assumption that X_{l_1} and X_{l_2} have the same covariance function can be tested using various inferential procedures presented in literature (see for example Benko *et al.* (2009), Fremdt *et al.* (2013), Pigoli *et al.* (2014)). Under the null hypotesis, using the Karhunen-Loève decomposition of $\bar{X}_{l_1}$ and $\bar{X}_{l_2}$ we have that

$$\left(\frac{\langle \bar{X}_{l_1} - \bar{X}_{l_2}, \vartheta_k \rangle}{\sqrt{\lambda_k}}\right)_k \overset{i.i.d}{\sim} \mathcal{N}\left(0, \frac{1}{n_{l_1}} + \frac{1}{n_{l_2}}\right)$$

and then

$$\left(\frac{1}{n_{l_1}} + \frac{1}{n_{l_2}}\right)^{-1} \cdot d_p^2(\bar{X}_{l_1}, \bar{X}_{l_2}) \sim \chi^2_{1,k} h_k(p)$$

where $(\chi^2_{1,k})_{k=1}^{\infty}$ are all independent. As a consequence, the following critical region

$$R^2_\alpha = \left\{ \left(\frac{1}{n_{l_1}} + \frac{1}{n_{l_2}}\right)^{-1} \cdot d_p^2(\bar{X}_{l_1}, \bar{X}_{l_2}) > \xi^2_{\alpha,p} \right\} \tag{2.4}$$

is of level α for any $p > 0$ and $n_{l_1}, n_{l_2} \in \mathbb{N}$.
We finally want to investigate the power of test (2.3). Following similar arguments used in case 1, we obtain that

$$\left(\frac{1}{n_{l_1}} + \frac{1}{n_{l_2}}\right)^{-1} \cdot d_p^2(\bar{X}_{l_1}, \bar{X}_{l_2}) \sim \sum_{k=1}^{\infty} \chi^2_{1,k}(\nu_k) h_k(p)$$

$$\nu_k = \left(\frac{1}{n_{l_1}} + \frac{1}{n_{l_2}}\right)^{-1} \cdot d^2_{M,k}(m_{l_1}, m_{l_2})$$

where $(\chi^2_{1,k})_{k=1}^{\infty}$ are all independent. Then, the power of (2.3) can be obtained as follows

$$\begin{aligned}
\beta_2 &= P_{m_{l_1} \neq m_{l_2}}(R^2_\alpha) \\
&= P_{m_{l_1} \neq m_{l_2}}\left(\left(\frac{1}{n_{l_1}} + \frac{1}{n_{l_2}}\right)^{-1} \cdot d_p^2(\bar{X}_{l_1}, \bar{X}_{l_2}) > \xi^2_{\alpha,p} \right) \\
&= P_{m_{l_1} \neq m_{l_2}}\left(\sum_{k=1}^{\infty} \chi^2_{1,k}(\nu_k) h_k(p) > \xi^2_{\alpha,p} \right)
\end{aligned}$$

with $\nu_k = \left(\frac{1}{n_{l_1}} + \frac{1}{n_{l_2}}\right)^{-1} \cdot d^2_{M,k}(m_{l_1}, m_{l_2})$. When n_{l_1} and n_{l_2} go to infinity, the power β_2 tends to one.

2.2 Inference on the mean of a Gaussian process with unknown covariance function

The inferential procedures presented in the previous section will now be extended to the case in which the covariance structure is unknown. The tests will be similar to (2.1) and (2.3) but the covariance operator with the related eigenvalues and eigefunctions are estimated from data.

For any $l = 1, \ldots, L$ we introduce

$$\hat{v}_{l,n_l} := \frac{1}{n_l - 1} \sum_{i=1}^{n_l} (X_{i,l}(s) - \bar{X}_l(s))(X_{i,l}(t) - \bar{X}_l(t))$$

as the estimator of the covariance operator v, computed using the n_l realizations of $P_{X_l} : X_{1,l}, \ldots, X_{n_l,l}$. If we call $N = n_1 + \ldots + n_l$ the total number of realizations from $X_1, \ldots, X_L$, we can define the pooled estimator of v computed using all data as

$$\hat{v}_N := \frac{1}{N - L} \sum_{l=1}^{L} (n_l - 1)\hat{v}_{l,n_l} \tag{2.5}$$

Let us denote with $(\lambda_k)_k$ the ordered eigenvalues of $\hat{v}_N$ and $(\hat{\vartheta}_k)_k$ the associated eigenfunctions. Since $\hat{\lambda}_k = 0 \quad \forall k \geq N$, the eigenfunctions $(\hat{\vartheta}_N, \hat{\vartheta}_{N+1}, \ldots)$ can be arbitrarily chosen such that $(\hat{\vartheta}_k)_k$ is an orthonormal basis of $L^2(I)$.

We must notice that the inferential procedures presented for the case of unknown covariance function are asymptotic; the nominal level of the tests is in fact achieved when the size of the sample used to estimate v is large. This could be different from the number of data involved in the estimation of the means to be test. All the asymptotic results hold for $N \to \infty$, which does not imply the divergence of all the sample sizes $n_1, \ldots, n_L$ (when $L \geq 1$). Since $N = n_1 + \ldots + n_L$, at least one among $n_1, \ldots, n_L$ must go to infinity if $N \to \infty$, but this is not necessarily the size of the sample drawn from the processes used to estimate the means considered in the hypothesis test.

Since the covariance structure, which is required to compute the generalized Mahalanobis distance d_p, is supposed unknown, in this section the critical regions will be constructed with statistics based on estimators of

the distance $\hat{d}_{p,N}$. The estimator of d_p based on the covariance structure $\hat{v}_N$ is defined as follows

$$\hat{d}^2_{p,N}(y,w) := \sum_{k=1}^{N-1} \hat{d}^2_{M,k}(y,w) \cdot \hat{h}_k(p) + p \sum_{k=N}^{\infty} \left(\langle y - w, \hat{\vartheta}_k \rangle \right)^2, \tag{2.6}$$

with $y, w \in L^2(I)$, where $\hat{d}^2_{M,k}(\cdot,\cdot)$ and $\hat{h}_k(p)$ represent the quantities $d^2_{M,k}(\cdot,\cdot)$ and $h_k(p)$, with $(\lambda_k)_k$ and $(\vartheta_k)_k$ replaced by $(\hat{\lambda}_k)_k$ and $(\hat{\vartheta}_k)_k$. If we compare the definition of $\hat{d}_{p,N}$ in (2.6) and d_p in (1.4), we notice how the first $N-1$ terms are similar while the terms $k \geq N$ are different because $\hat{\lambda}_k = 0 \quad \forall k \geq N$, so $\hat{d}_{M,k}$ would be undefined for $k \geq N$. For this reason, in (2.6) it has been introduced a correction in order to make the estimate of $d_{p,N}$ as close as possible to d_p. In particular, since $\lambda_k^{-1} h_k(p) \to p$ as $\lambda_k \to 0$, in (2.6) it has been redefined $\hat{d}_{M,k}\hat{h}_k(p) := p \cdot \left(\langle y - w, \hat{\vartheta}_k \rangle \right)^2$ for any $k \geq N$. Before constructing the tests based on the distance $\hat{d}_{p,N}$ with the same structure of (2.2) and (2.4), we need some auxiliar results.

Theorem 2.2. *For any $N, n_0 \in \mathbb{N}$, let $j_N = (j_{N,1}, \ldots, j_{N,n_0})$ be a vector of integers $(j_{N,1}, \ldots, j_{N,n_0} \in \mathbb{N})$ and let $(Y_{j_N}, W_{j_N})_N$ be a couple of stochastic processes, independent of the sequence $(\hat{v}_k)_k$ and such that*

$$\sup_{N \geq 1} \mathbb{E}[||Y_{j_N} - W_{j_N}||^2] < \infty. \tag{2.7}$$

Then, we have that

$$\mathbb{E}\left[|\hat{d}^2_{p,N}(Y_{j_N}, W_{j_N}) - d^2_p(Y_{j_N}, W_{j_N})| \right] \underset{N \to \infty}{\overset{D}{\to}} 0. \tag{2.8}$$

(2.8) is an important result proved through Slutsky's Theorem that the statistics based on $\hat{d}_{p,N}$ and those based on d_p converge to an asymptotic distribution. We highlight this in the following result:

Corollary 2.3. For any $l = 1, \ldots, L$ and for any $m_0 \in L^2(I)$ we have that

$$n_l \cdot \hat{d}^2_{p,N}(\bar{X}_l, m_0) \underset{N \to \infty}{\overset{D}{\to}} \sum_{k=1}^{\infty} \chi^2_{1,k}(\nu_k) h_k(p) \tag{2.9}$$

where $\nu_k = n_l \cdot d^2_{M,k}(m_l, m_0)$. Moreover, when $L \geq 2$, for any $l_1, l_2 = 1, \ldots, L$, we have that

$$\left(\frac{1}{n_{l_1}} + \frac{1}{n_{l_2}} \right)^{-1} \cdot \hat{d}^2_{p,N}(\bar{X}_{l_1}, \bar{X}_{l_2}) \underset{N \to \infty}{\overset{D}{\to}} \sum_{k=1}^{\infty} \chi^2_{1,k}(\nu_k) h_k(p), \tag{2.10}$$

where $\nu_k = \left(\frac{1}{n_{l_1}} + \frac{1}{n_{l_2}} \right)^{-1} \cdot d^2_{M,k}(m_{l_1}, m_{l_2})$.

Corollary 2.3 ensures that, for the hypothesis test (2.1) and (2.3), the critical regions based on the covariance estimator have the same structure of those in (2.2) and (2.4), respectively, where the covariance function is assumed to be known. However, the asymptotic distribution $\sum_{k=1}^{\infty} \chi^2_{1,k}(\nu_k) h_k(p)$ depends on the eigenvalues of v, which are unknown here. To compute the power function of the tests, we need this further asymptotic result:

Theorem 2.4. *Let $(\chi^2_{1,k})_k$ be a sequence of i.i.d. chi-squared processes with 1 d.o.f., independent of $\hat{v}_N$. Let $\hat{\xi}^2_{\alpha,p}$ be the 1-α quantile of the conditional distribution of $\sum_{k=1}^{\infty} \chi^2_{1,k} \hat{h}_k(p)$ given $(\hat{\lambda}_k)_k$. Then, we have that*

$$\hat{\xi}^2_{\alpha,p} \underset{N\to+\infty}{\to} \xi^2_{\alpha,p} \tag{2.11}$$

Using now (2.3) and (2.4) we can construct the critical regions for hypothesis tests (2.1) and (2.3) when the covariance operator is unknown.

Part 1:
Consider the hypothesis test (2.1). From Slutsky's Theorem and equations (2.9) and (2.11), we have that the following critical region is asymptotically in N of level α:

$$R^3_\alpha = \left\{ n_l \cdot \hat{d}^2_{p,N}(\bar{X}_l, m_l) > \hat{\xi}^2_{\alpha,p} \right\}. \tag{2.12}$$

Following the same arguments of the previous section, the power of test (2.12) can be written as follows:

$$\begin{aligned} \beta_3 &= P_{m_l \neq m_0}(R^3_\alpha) \\ &= P_{m_l \neq m_0}\left(n_l \cdot \hat{d}^2_p(\bar{X}_l, m_0) > \hat{\xi}^2_{\alpha,p} \right) \\ &\underset{N\to+\infty}{\to} P\left(\sum_{k=1}^{\infty} \chi^2_{1,k}(\nu_k) h_k(p) > \xi^2_{\alpha,p} \right) \end{aligned}$$

with $\nu_k = n_l \cdot d^2_{M,k}(m_l, m_0)$.

Part 2:
Assume $L \geq 2$ and consider the hypothesis test (2.3). From Slutsky's Theorem, equations (2.10) and (2.11) , we have that the following critical region is asymptotically in N of level α:

$$R^4_\alpha = \left\{ \left(\frac{1}{n_{l_1}} + \frac{1}{n_{l_2}} \right)^{-1} \cdot \hat{d}^2_{p,N}(\bar{X}_{l_1}, \bar{X}_{l_2}) > \hat{\xi}^2_{\alpha,p} \right\} \tag{2.13}$$

Following the same arguments of the previous section, the power of test (2.12) can be written as follows:

$$\begin{aligned}\beta_4 &= P_{m_{l_1}\neq m_{l_2}}(R^4_\alpha)\\ &= P_{m_{l_1}\neq m_{l_2}}\left(\left(\frac{1}{n_{l_1}}+\frac{1}{n_{l_2}}\right)^{-1}\cdot \hat{d}^2_p(\bar{X}_{l_1},\bar{X}_{l_2}) > \xi^2_{\alpha,p}\right)\\ &\underset{N\to+\infty}{\to} P_{m_{l_1}\neq m_{l_2}}\left(\sum_{k=1}^{\infty}\chi^2_{1,k}(\nu_k)h_k(p) > \xi^2_{\alpha,p}\right)\end{aligned}$$

with $\nu_k = \left(\frac{1}{n_{l_1}}+\frac{1}{n_{l_2}}\right)^{-1}\cdot d^2_{M,k}(m_{l_1},m_{l_2})$.

2.3 Simulation Studies

Starting from the tests presented in the previous sections, now we discuss some inferential properties concerning them in a simulation setting, focusing our attention on three main aspects:

1. general properties of the tests power function (Subsection 2.3.1);
2. dependence of the test power function on the choice of the parameter p (Subsection 2.3.2 and Subsection 2.3.3);
3. properties of the tests with unknown covariance function for large sample sizes (Subsection 2.3.4).

2.3.1 Simulations on the power function

In this first simulation study, our goal is to show some general properties regarding the power function presented in the previous section. We wish to test $H_0 : m_1 = m_0$ vs $H_1 : m_1 \neq m_0$, with the critical region R^1_α in (2.2). We are considering a Gaussian process X_1 in $L^2([0,1])$ with probability law P_{X_1}, mean supposedly unknown $m_1(t)$ and covariance function $v(s,t)$. The sample curves are generated through the Karhunen-Loève expansion, using $K = 100$ components and a grid of 500 equispaced points in $I = [0,1]$ as follows:

$$X_{i,1}(t) = m_{i,1}(t) + \sum_{k=1}^{\infty} Z_{ki,1}\sqrt{\lambda_k}\vartheta_k(t), \quad t\in[0,1], i\in\{1,\dots,n\}$$

where we set:

- the sample size $n = 10$;
- the random variables $\{Z_{ki,1}; k \geq 1, 1 \leq i \leq n\}$ as i.i.d. Gaussian processes with law P_{X_1};
- $\{\lambda_k; k \geq 1\}$ is the sequence of eigenvalues of the covariance function v defined as follows:

$$\lambda_k = \begin{cases} \frac{1}{k+1} & \text{if } k \in \{1,2,3\} \\ \frac{1}{(k+1)^4} & \text{if } k \geq 4 \end{cases} \tag{2.14}$$

- $\{\vartheta_k; k \geq 1\}$ is the sequence of eigenfunctions of the covariance function v defined as follows:

$$\vartheta_k = \begin{cases} \mathbf{1}_{\{t\in[0,1]\}} & \text{if } k = 1 \\ \sqrt{2}\sin(k\pi t)\mathbf{1}_{\{t\in[0,1]\}} & \text{if } k \geq 2,\ k \text{ even} \\ \sqrt{2}\cos((k-1)\pi t)\mathbf{1}_{\{t\in[0,1]\}} & \text{if } k \geq 3,\ k \text{ odd} \end{cases}$$

We consider now the hypothesis test (2.1) with $m_0(t) = t(1-t)\mathbf{1}_{\{t\in[0,1]\}}$ as mean function in H_0 and set $\alpha = 0.05$ as significance level. To better understand the power of R^1_α in (2.2), we compute it for different mean functions in H_1, defined as $m_{1k}(t) = m_0(t) + 0.03 \cdot \vartheta_k(t)$, for $k = 1, \ldots, 10$. We can notice that all the possible mean functions m_{1k} have the same L^2-distance from the tested mean m_0: $d_{L^2}(m_{1k}, m_0) = 0.03 \quad \forall k = 1, \ldots, 10$. The generalized Mahalanobis distance d_p used in R^1_α has been computed choosing $h_k(p) = (1-\exp(-\lambda_k p))$, which comes from the choice $g(c;p) = \mathbf{1}_{\{c\leq p\}}$. Since in this section we aren't focusing on the role of p, we fix $p = 10^3$. The empirical power is represented in Fig. 2.1 and has been computed by realizing 10^3 times the test (2.2), using each time $n = 10$ independent Gaussian processes generated from P_{X_1}. We can notice from Fig. 2.1 that the power is strictly increasing in k, even if $d_{L^2}(m_{1k}, m_0) = 0.03 \quad \forall k = 1, \ldots, 10$. This is due to the fact that the test statistics is constructed using the distance d_p, which is able to distinguish the differences of $(m_{1k} - m_0)$ along the principal components ϑ_k and to weight them according to the variability of X along the components λ_k, which is the idea underlying the Mahalanobis distance. Naturally, since the Mahalanobis distance obtained with $p \to \infty$ is not defined in infinite dimensional spaces, it is needed to fix the parameter p which rules how

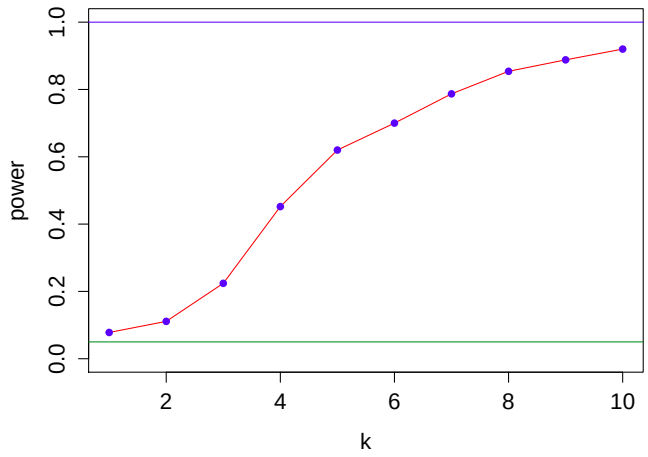


Figure 2.1: The empirical power of test (2.2) for $m_{1k}(t) = m_0(t) + 0.03 \cdot \vartheta_k(t)$, for $k = 1, \dots, 10$, realized with 10^3 simulations, $n = 10$ and $p = 10^3$.

much we can distinguish different low variances. However, the important thing now is that for any choice of $p > 0$, the test really considers all the components without any truncation.
We will show now that the parameter p plays an important statistical role in the inferential properties of the tests and that setting p arbitrarily large is not in general the right choice for the analysis.

2.3.2 Simulations on the test power as function of p

The next goal is to discuss how the choice of the parameter p affects the tests presented in the previous sections. Withouth loss of generality, we only consider the hypothesis test $H_0 : m_1 = m_0$ vs $H_1 : m_1 \neq m_0$, with the critical region R^1_α expressed in (2.2), since the discussion on the role of p for the other critical regions is analogous. In the first section of the chapter, we derived the analytic expression of the function power for the test (2.2) as

$$\beta_1 = P\left(\sum_{k=1}^{\infty} \chi^2_{1,k}(\nu_k) h_k(p) > \xi^2_{\alpha,p} \right)$$

with $\nu_k = n \cdot d^2_{M,k}(m_1, m_0)$.

We compute now the power β_1 for different choices of the parameter $p > 0$, when the true mean $m_1(t)$ is one of the following:

1. $m_1(t) = m_0(t)$, the true mean $m_1(t)$ coincides with the function in the null hypothesis $m_0(t) \Rightarrow d_{M,k}(m_1, m_0) = 0 \quad \forall k \geq 1$;

2. $m_1(t) = m_0(t) + \sqrt{\lambda_1} \cdot \vartheta_1(t)$, the true mean $m_1(t)$ and $m_0(t)$ only differs in the first component $\Rightarrow d_{M,1}(m_1, m_0) = 1$, $d_{M,k}(m_1, m_0) = 0 \quad \forall k \neq 1$;

3. $m_1(t) = m_0(t) + \sqrt{\lambda_5} \cdot \vartheta_5(t)$, the true mean $m_1(t)$ and $m_0(t)$ only differs in the fifth component $\Rightarrow d_{M,5}(m_1, m_0) = 1$, $d_{M,k}(m_1, m_0) = 0 \quad \forall k \neq 5$;

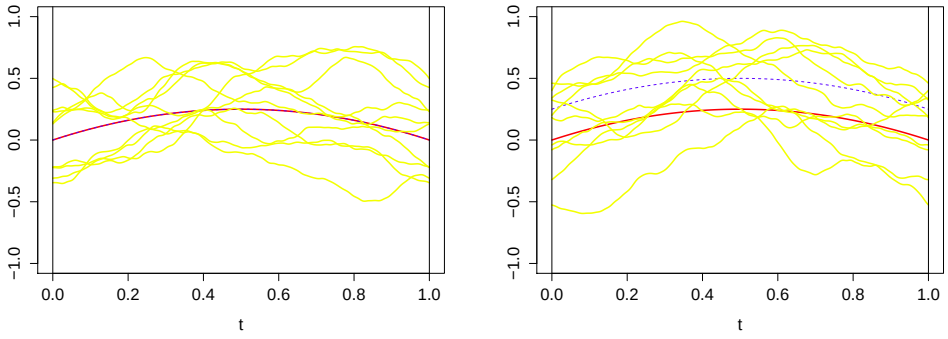
4. $m_1(t) = m_0(t) + \sum_{k=10}^{\infty} \sqrt{\lambda_k} \cdot \vartheta_k(t)$, the true mean $m_1(t)$ and $m_0(t)$ differs in all the component but the first nine $\Rightarrow d_{M,k}(m_1, m_0) = 1 \quad \forall k \geq 10$, $d_{M,k}(m_1, m_0) = 0 \quad \forall k < 10$.

For each one of the previous cases, the power β_1 has been computed by repeating 10^3 times the test (2.2), each time using $n = 10$ independent Gaussian processes generated from the law P_{X_1}. In Fig. 2.2 the four cases are separetaly reported. We can see an example of the functional sample $X_1, \ldots, X_{10}$ (yellow lines), the real mean m_1 (red line) and the tested mean m_0 (blue dotted line). On Fig. 2.3 we have instead the empirical power of R^1_α (2.2) (blue dots) for each of the value of the parameter $p \in \{10^{-2}, 10^{-1}, \ldots, 10^8\}$. We have also depicted the empirical power of the test (2.1) when we use the truncated version of the Mahalanobis, summing up only the first $K = 3$ components (blue line), which explained the most of the variability. In this case, the critical region becomes:

$$\tilde{R}^1_\alpha = \{n \cdot \sum_{k=1}^{3} d^2_{M,k}(\bar{X}_n, m_0) > \tilde{\xi}^2_\alpha\}. \tag{2.15}$$

where $\tilde{\xi}^2_\alpha = \sum_{k=1}^{3} \chi^2_{1,k}$. There are also dotted curves which reprent the same procedure realized with $n = 20$ (dotted orange curves) and with $n = 50$ (dotted yellow curves), showing that the power increases with the sample size n, for any $p > 0$ and for any mean $m_1 \in H_1$.

In case (a) the null hypothesis is true, then the proportion of rejection of both R^1_α (2.2) and $\tilde{R}^1_\alpha$ (2.15) is equal to the level of the test $\alpha = 0.05$, for any choice of $p > 0$.



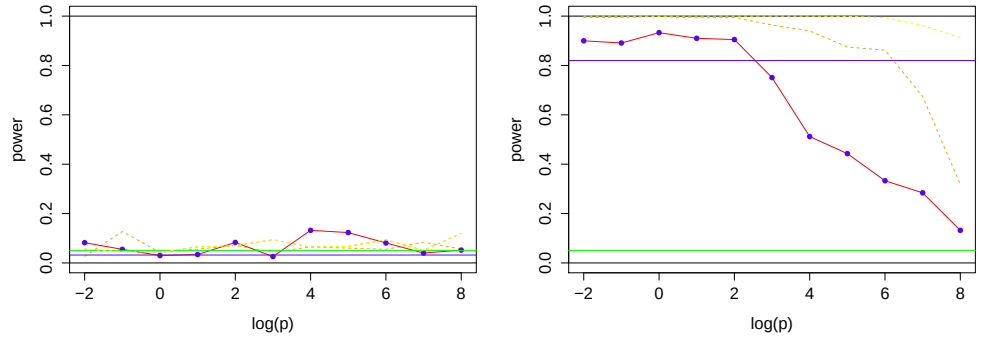
(a) $m_1(t) = m_0(t)$

(b) $m_1(t) = m_0(t) + \sqrt{\lambda_1} \cdot \vartheta_1(t)$

(c) $m_1(t) = m_0(t) + \sqrt{\lambda_5} \cdot \vartheta_5(t)$

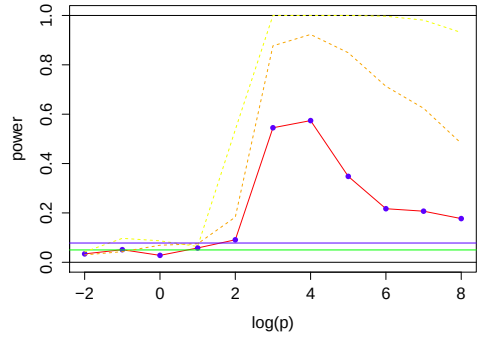
(d) $m_1(t) = m_0(t) + \sum_{k=10}^{\infty} \sqrt{\lambda_k} \cdot \vartheta_k(t)$

Figure 2.2: Functional sample $X_1, \ldots, X_{10}$ (yellow lines) along with the real mean $m_1(t)$ (red line) and the tested mean $m_0(t)$ (blue dotted line).

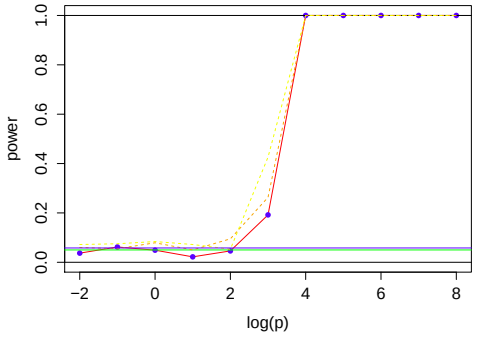


(a) $m_1(t) = m_0(t)$

(b) $m_1(t) = m_0(t) + \sqrt{\lambda_1} \cdot \vartheta_1(t)$



(c) $m_1(t) = m_0(t) + \sqrt{\lambda_5} \cdot \vartheta_5(t)$



(d) $m_1(t) = m_0(t) + \sum_{k=10}^{\infty} \sqrt{\lambda_k} \cdot \vartheta_k(t)$

Figure 2.3: For each case (a) - (b) - (c) - (d) we have the empirical power of R^1_α (2.2) for the values of the parameter $p \in \{10^{-2}, 10^{-1}, \ldots, 10^8\}$ realized with 10^3 simulations, with $n = 10$ (red line), $n = 20$ (orange line) and $n = 50$ (yellow line). It is also depicted the empirical power of the test (2.1) when we use the truncated version of the Mahalanobis distance (blue line) and the level of the test $\alpha = 0.05$ (green line).

We consider now case (b), where the first component is the only difference among $m_0(t)$ and $m_1(t)$. When p is small, the term $h_1(p)$ is very relevant with respect to $(h_k(p), k \geq 2)$; then, the non-centrality quantity $d_{M,1}(m_1, m_0) = 1$ multiplied by $h_1(p)$ makes the power higher. In this case, the empirical power with small p of R^1_α (2.2) is obviously equal to the empirical power of $\tilde{R}^1_\alpha$ (2.15). However, while the second one does not change since it does not depend on p, the more p increases, the more terms $(h_k(p), k = 1, 2, \ldots)$ become close to one; thus, the test statistics $d^2_p(\bar{X}_n, m_0)$ is now influenced by more components and the first one becomes less relevant. For this reason, since $m_0(t)$ and $m_1(t)$ only differ in the first component, $d_{M,k}(m_1, m_0) = 0 \quad \forall k \geq 2$ and the power decreases as p increases.

In case (c), the fifth component is the only difference among $m_0(t)$ and $m_1(t)$. In situations like this one, it is relevant why the d_p distance gives better results than the truncated one; in fact, the empirical power of the test related to the truncated distance is equal to the level $\alpha = 0.05$ since it can't detect the difference between $m_0(t)$ and $m_1(t)$. For what concerns the d_p distance, for small values of the parameter p we have $h_5(p) \simeq 0$, so the contribute of $d_{M,5}(m_1, m_0) = 1$ is irrelevant and the power is low, as it happens with the truncated distance. As p increases, $h_5(p)$ becomes closer to one and the power grows. However, since $d_{M,k}(m_1, m_0) = 0 \quad \forall k > 5$, when p increases a lot, there are too many irrelevant components influencing the test statistics $d^2_p(\bar{X}_n, m_0)$ and the power falls down.

In case (d), $m_0(t)$ and $m_1(t)$ have the same contributions in the first 9 components. For the same reason as the previous case, the empirical power of the test (2.15) is equal to the level of the test $\alpha = 0.05$. Regarding the test (2.2), in this case when p is small we have that $h_k(p) \simeq 0, k \geq 10$, so the test is unable to detect any difference and the power is low. As p increases, some $(h_k(p), k \geq 10)$ become close to one, the statistics $d^2_p(\bar{X}_n, m_0)$ is influenced by $d_{M,k}(m_1, m_0) = 1$ for $k \geq 10$ and the power grows. When $p \to +\infty$, almost all components of the statistics $d^2_p(\bar{X}_n, m_0)$ have a contribution $d_{M,k}(m_1, m_0) = 1$, so that the power tends to one.

2.3.3 Discussion on the role of p

As we have just discussed, the choice of the parameter $p > 0$ determines in which components the test R^1_α performs well in detecting possible

differences among the supposed mean $m_0(t)$ and the true mean $m_1(t)$. As we have seen, the test (2.1) can provide different results depending on the value of the parameter p.

When we set a small p, we obtain a test whose power of detecting differences beetween m_0 and the true mean m_1 does not discriminate very well the different components with their own variability, but is more suitable for looking at the shape of the function; the distance d_p is closer to the L^2-distance than the Mahalanobis one. For this reason, this can be a good choice when we are more interested in the general closeness between the supposed mean m_0 and the true man m_1 than the micro-structure's features. For instance, in case (b) m_0 is quite far from m_1 and the test with low p works well. At the contrary, in case (c) and (d), where the macro-structure between m_0 and m_1 is very similar, the power of the test with low p is low.

When we set a large value for the parameter p, the test explores a greater number of components, taking into account their own variability, since in the distance d_p many weights $h_k(p)/\lambda_k$ are close to $1/\lambda_k$. For this reason, the distance d_p between m_1 and m_0 acts much more like the Mahalanobis distance than like the classical L^2-distance. When we decide to set a large value of p, there are two aspects that we should consider:

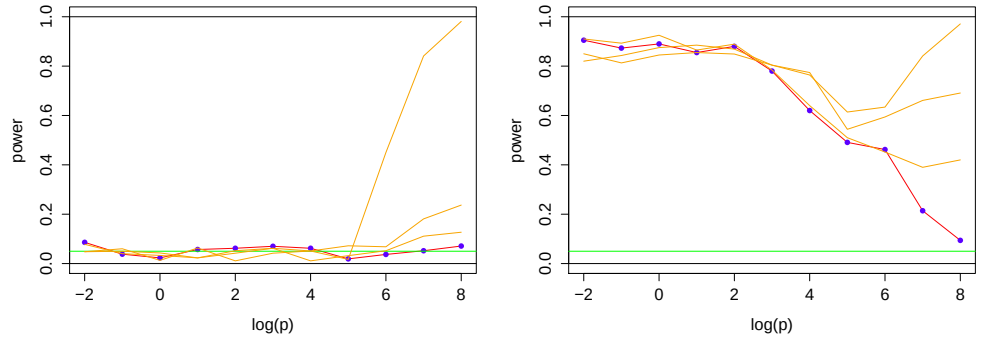
- each component is less relevant in the test statistics. For example, in the cases (b) and (c), $m_0(t)$ and $m_1(t)$ differ for only one component; then, when p is very large that component becomes negligible and the power of the test decrases. The opposite case is (d), in which the power tends to one for large p because the first nine components, which are in favour of H_0, become negligble;

- the test statistics less reflect the macro-structure of the function. For instance, in case (b) the curves $m_0(t)$ and $m_1(t)$ concerns the first component, which is the most relevant. However, for large values of p the test is more interested in looking at the average behaviour of many components and the distance easily visualized in the graphic is forgotten. The opposite case is (d), in which $m_0(t)$ and $m_1(t)$ seems very similar at first sight; in this case, the test does not care about the visualization of the curves and it is more concentrated on the average difference in many components, which leads to reject the null hypothesis H_0 with high probability.

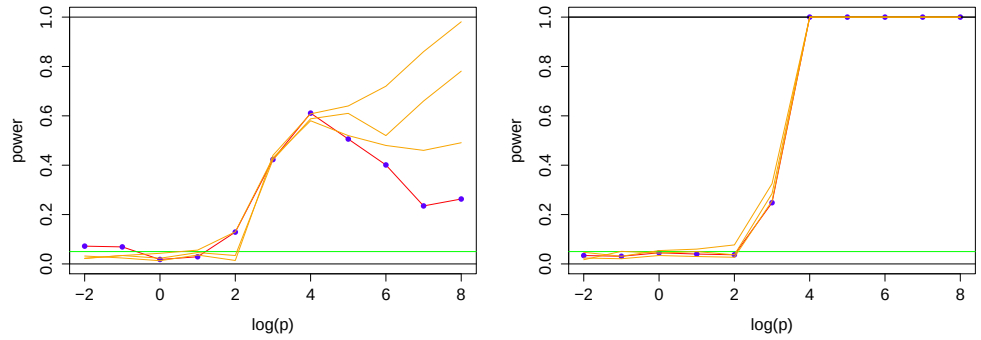
We can conclude that there is not an optimal choice for p: according to the main interest associated with the test, we should properly set the parameter p to detect differences related to a specific aim.

2.3.4 Simulations on tests with unknown covariance function

We want to show now that the critical regions based on the estimated covariance function present for large sample sizes the same power of the critical regions with known covariance structure. Starting from the hypothesis test $H_0 : m_1 = m_0$ vs $H_1 : m_1 \neq m_0$, with the critical region R^3_α expressed in (2.12), we adopt the same framework presented for the simulations when the covariance structure is known. In addition to the $n_1 = 10$ processes from P_{X_1} previously defined, we introduce n_2 i.i.d. Gaussian processes with probability law P_{X_2}, having the same covariance function $v(s,t)$ of P_{X_1}. These $N = n_1 + n_2$ processes provide the covariance estimate $\hat{v}_N$ as espressed in (2.5). For each $n_2 \in \{100, 500, 1000\}$ we compute the power of R^3_α as function of the parameter $p \in \{10^{-2}, 10^{-1}, \ldots, 10^8\}$. This procedure is repeated changing the true mean m_1 of P_{X_1}, as specified in the cases $(a) - (b) - (c) - (d)$. The power functions are represented in Fig. 2.4 with orange dotted lines. The blue dots indicate the power of test R^1_α adopted when the covariance operator v is known. As we can see from Fig. 2.4, the power functions of R^3_α tend to the power of R^1_α as N goes to infinity. Moreover, the convergence is faster when p is small and slower when p is large. We can conclude saying that all the results obtained in this chapter can be expanded for non-Gaussian processes, as it has been done in Ghiglietti *et al.* (2016).



(a) $m_1(t) = m_0(t)$ **(b)** $m_1(t) = m_0(t) + \sqrt{\lambda_1} \cdot \vartheta_1(t)$



(c) $m_1(t) = m_0(t) + \sqrt{\lambda_5} \cdot \vartheta_5(t)$ **(d)** $m_1(t) = m_0(t) + \sum_{k=10}^{\infty} \sqrt{\lambda_k} \cdot \vartheta_k(t)$

Figure 2.4: The empirical power of tests (2.2) (red line) and (2.12) (orange lines) for each case (a) - (b) - (c) - (d) for $p \in \{10^{-2}, 10^{-1}, \ldots, 10^{8}\}$, $n_1 = 10$, $n_2 \in \{100, 500, 1000\}$, realized with 10^3 simulations.

Chapter 3

Functional Clustering Methods

In the last part of this work, we want to extend the use of the d_p distance previously presented, adapting it to functional clustering techniques. In particular, the aim of the analysis is to develop a proper classification procedure in the functional framework based on the d_p distance. We first want to show through simulations the validity of the procedure, eventually applying it to a real case study.

3.1 Functional k-means

In Tarpey and Kinateder (2003) we can found a proper definition of the functional k-means procedure and an introduction to its consistency properties. The functional k-means clustering algorithm is an iterative procedure, alternating a step of *cluster assignement*, where all curves are assigned to a cluster, and a step of *centroid calculation*, where a relevant functional representative (the centroid) for each cluster is identified. More precisely, the algorithm is initialized by fixing the number of k clusters and by randomly selecting between the curves in the data set a set of k initial centroids $\{\varphi_1^{(0)}(t), \ldots, \varphi_k^{(0)}(t)\}$. As it is usually done in the multivaraite k-means algorithm, to select the initial mean curves we initialize the means by randomly selecting k different data. Given this initial choice, the algorithm iteratively repeats two basic steps. At the m^{th} iteration of the algorithm, $m > 0$, the two following steps are performed:

***Step 1 (cluster assignment step)*:** each curve is assigned to the cluster whose centroid at the $(m-1)^{th}$ iteration is the nearest according to the chosen distance d. This means that the choice of the m^{th} cluster assignment of the i^{th} statistical unit $C_i^{(m)}$, for $i = 1, \ldots, n$ is

$$C_i^{(m)} = \underset{l=1,\ldots,k}{\operatorname{argmin}}\, d\{\mathbf{F}_i(t), \psi_l^{(m-1)}(t)\}.$$

***Step 2 (centroid calculation step)*:** the identification of the centroids $\{\varphi_1^{(m)}(t), \ldots, \varphi_k^{(m)}(t)\}$ at the m^{th} iteration is performed by solving the optimization problem

$$\varphi_l^{(m)}(t) = \underset{\psi \in \Omega_d}{\operatorname{argmin}} \sum_{j:C_i^{(m)}=l} d\{\mathbf{F}_i(t), \varphi(t)\}^2,$$

where $C_i^{(m)}$ is the cluster assignment of the i^{th} statistical unit at the current iteration, d is a distance and Ω_d is the space where the chosen distance d is natural.

The solution to the infinite dimensional optimization problem that is expressed in the centroid calculation step obviously depends on the choice of the distance.

The algorithm is stopped when the same cluster assignments are obtained at two subsequent iterations, i.e. the set of cluster assignements $\{C_1^{(\bar{m})}, \ldots, C_n^{(\bar{m})}\}$ and the set of centroids $\{\varphi_1^{(m)}(t), \ldots, \varphi_k^{(m)}(t)\}$ are considered final solutions of the algorithm if we obtain $C_i^{(\bar{m}+1)} \equiv C_i^{(\bar{m})}$ for all $i = 1, \ldots, n$.

Obviously, the k-means clustering procedure depends not only on the choice of the distance, but also on the number of clusters k. Since the number of clusters is a priori unknown, we also consider a way to compute the optimal number of clusters k^* via silhouette values and a plot of the final classification, see Struyf *et al.* (1997). In particular, the silhouette plot of a classification consists of a bar plot of the *silhouette values* s_i, obtained for each statistical unit $i = 1, \ldots, n$ as

$$s_i = \frac{b_i - a_i}{max\{a_i, b_i\}},$$

where a_i is the average distance, according to one of the distances considered, of the ith statistical unit to all other ones assigned to the same

cluster, whereas

$$b_i := \underset{l=1,\dots,k;l\neq C_i}{\operatorname{argmin}} \frac{\sum_{j:C_j=l} d\{\mathbf{F}_i(t), \mathbf{F}_j(t)\}}{\#\{j : C_j = l\}}$$

is the minimum average distance of the ith statistical unit from another cluster and d is one of the distances considered. Clearly s_i always lies between -1 and 1, the former value indicating a misclassified statistical unit while the latter indicates a well-classified one. Note that a statistical unit who alone constitutes a cluster has silhouette value equal to 1 but is not considered in the silhouette plot for choosing k^*.

3.2 Simulations

3.2.1 Simulations in univariate functional framework

The main goal of this simulation study is to consider two samples of i.i.d. curves $X_i(t)$ and $Y_i(t), i = 1, \dots, n$, generated by independent stochastic processes with different means such that $X_i(t), Y_i(t) \in L^2(I)$, where I is a compact interval of $\mathbb{R}$. As we already did in Chapter 3, we use the Karhunen-Loève decomposition to generate the sample curves in the following way:

$$X_i(t) = m_0(t) + \sum_{k=1}^{\infty} Z_{ki,1}\sqrt{\lambda_k}\vartheta_k(t) \quad i = 1, \dots, n_1,$$

$$Y_i(t) = m_1(t) + \sum_{k=1}^{\infty} Z_{ki,2}\sqrt{\lambda_k}\vartheta_k(t) \quad i = 1, \dots, n_2,$$

where we set:

1. the time grid of 1000 equispaced points in $I = [0, 1]$;
2. truncation at $K = 200$ components;
3. the same sample sizes $n_1 = n_2 = 50$;
4. the mean of the first sample $m_0(t) = t(1 - t)$ while we set different values for the mean of the second sample;
5. the random variables $(Z_{ki,1})_{k=1}^{\infty}$ $(Z_{ki,2})_{k=1}^{\infty}$ are two sequences of independent standard normal variables;

6. $\{\lambda_k; k \geq 1\}$ is the sequence of eigenvalues of the covariance function v defined as follows:

$$\lambda_k = \begin{cases} \frac{1}{k+1} & \text{if } k \in \{1,2,3\} \\ \frac{1}{(k+1)^2} & \text{if } k \geq 4 \end{cases} \tag{3.1}$$

7. $\{\vartheta_k; k \geq 1\}$ is the sequence of eigenfunctions of the covariance function v defined as follows:

$$\vartheta_k = \begin{cases} \mathbf{1}_{\{t\in[0,1]\}} & \text{if } k = 1 \\ \sqrt{2}\sin(k\pi t)\mathbf{1}_{\{t\in[0,1]\}} & \text{if } k \geq 2,\ k \text{ even} \\ \sqrt{2}\cos((k-1)\pi t)\mathbf{1}_{\{t\in[0,1]\}} & \text{if } k \geq 3,\ k \text{ odd} \end{cases}$$

Starting from the eigenvalues $(\lambda_k)_k$ with the associated eigenfunctions $(\vartheta_k)_k$, we generated the curves in three different cases:

(a) $m_1(t) = m_0(t) + \sqrt{\lambda_1} \cdot \vartheta_1$

(b) $m_1(t) = m_0(t) + \sqrt{\lambda_5} \cdot \vartheta_5$

(c) $m_1(t) = m_0(t) + \sum_{k=10}^{\infty} \sqrt{\lambda_k} \cdot \vartheta_k$

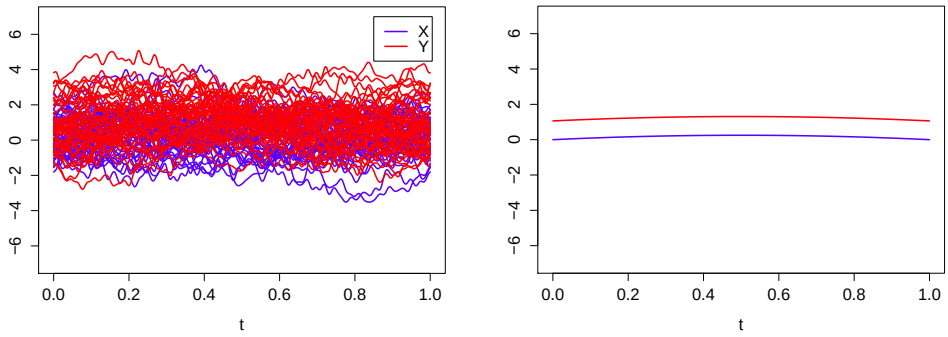
For each case we then computed a k-means algorithm based on three different distances between the two samples.

Starting from our two samples X and Y, we computed the empirical eigenvalues $(\hat{\lambda}_k)_k$ with the associated eigenfunctions $(\hat{\vartheta}_k)_K$ so that we could compute the d_p distance as follows:

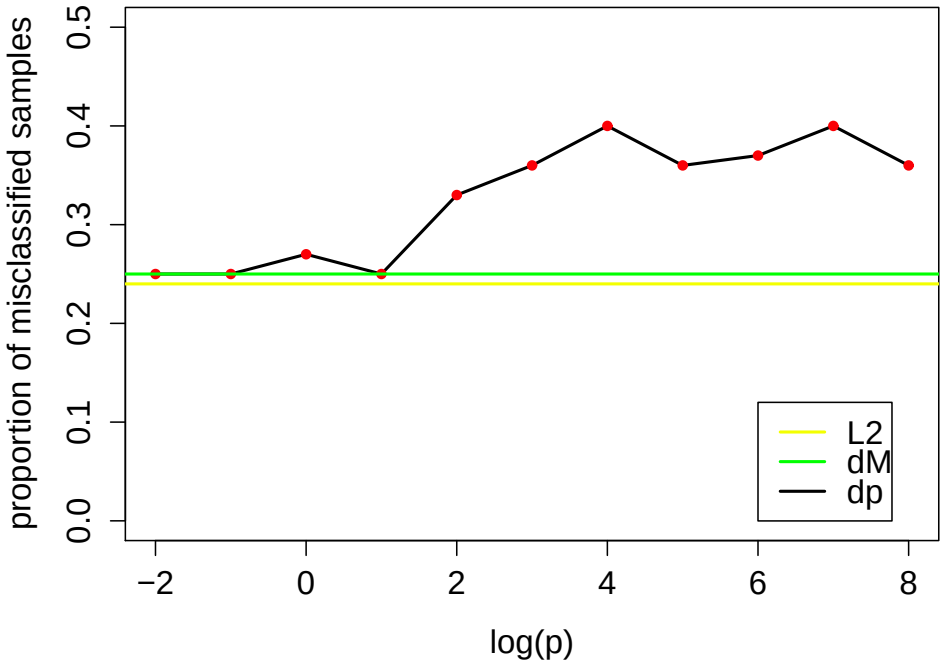
$$d_p(X,Y) = \sqrt{\sum_{k=1}^{\infty} d_{M,k}^2(X,Y) \cdot h_k(p)}$$

where $d_{M,k}(X,Y) = \sqrt{(\langle X - Y, \vartheta_k \rangle)^2/\lambda_k}$ is the term representing the contribution of the Mahalanobis distance along the k^{th} component and $h_k(p) = \lambda_k/(\lambda_k + 1/p)$. We considered other two competitors to compare our distances with: a truncated version of the Mahalanobis distances (summing up $K = 3$ components, which described the most of the varaibility) and the classical L^2-distance which can be written as:

$$d_M(X,Y) = \sqrt{\sum_{k=1}^{3} d_{M,k}^2(X,Y)}$$

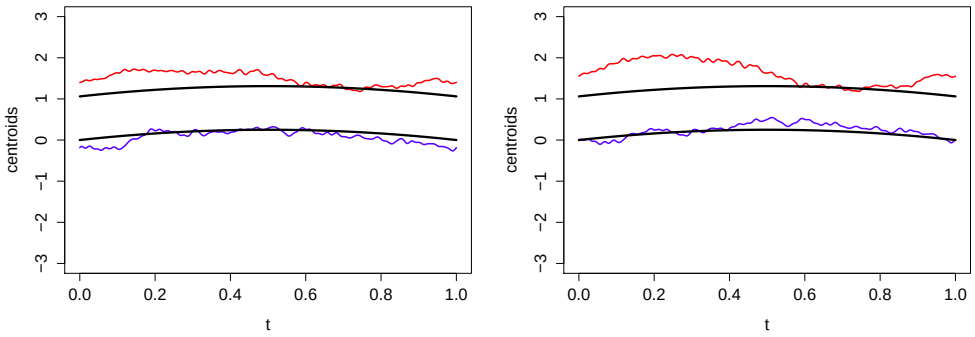


(a) (b)

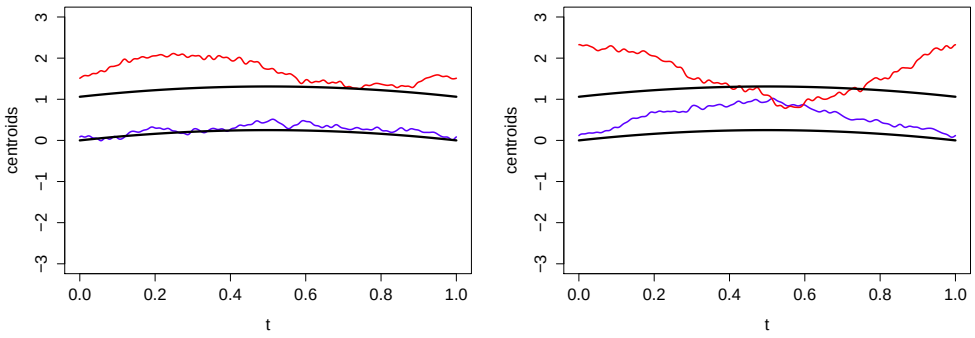


(c)

Figure 3.1: (a) Functional samples $X_1, \ldots, X_{50}$ (blue lines) and $Y_1, \ldots, Y_{50}$ (red lines).
(b) Means of the two samples $m_0(t)$ (blue line) and $m_1(t)$ (red line).
(c) Proportion of misclassified sample with the functional k-means for case (a) using the L^2-distance (yellow line), the truncated version of the Mahalanobis distance (green line) and the d_p distance (black line).



(a) d_{L^2} **(b)** d_M



(c) d_p, low p **(d)** d_p, high p

Figure 3.2: Real means of the functional sample (black lines) and centroids obtained with the k-means clustering procedures (blue and red lines) with different distances.

(a) d_{L^2}

Cluster	X	Y
1	36	10
2	14	40

(b) d_M

Cluster	X	Y
1	41	16
2	8	34

(c) d_p, low p

Cluster	X	Y
1	42	17
2	8	33

(d) d_p, high p

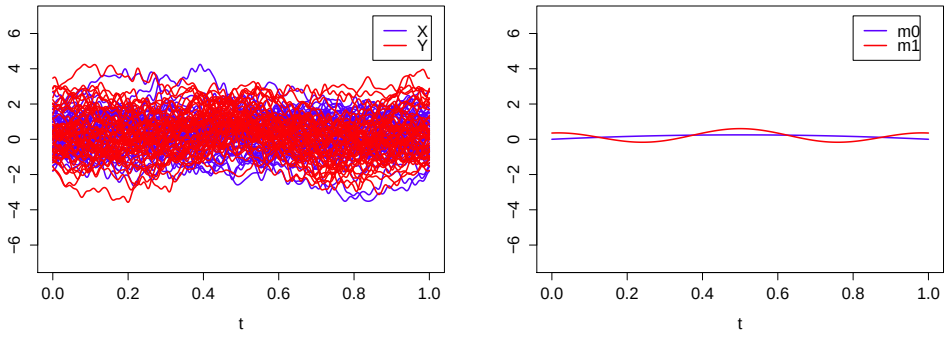
Cluster	X	Y
1	34	28
2	16	32

Table 3.1: Confusion matrices related to the functional k-means for the samples X and Y for case (a).

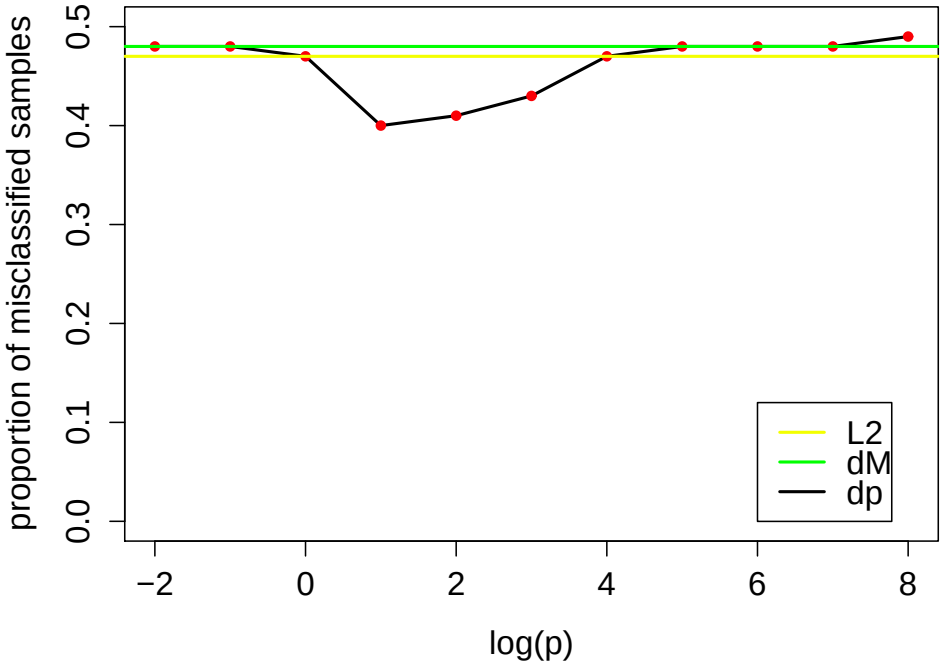
$$d_{L^2}(X,Y) = \sqrt{\int_I (X(t) - Y(t))^2 dt}$$

In Fig. 3.1(a) we can see the two samples X (blue lines) and Y (red lines) in case (a) and their means in Fig. 3.1(b). In this simulation, the two means differ only on the first component. Since it is just a traslation, the classical L^2-distance seems to work pretty well, assigning about the 70% of the data to the right group, as we can read in Table 3.1. For what concerns the other two distances, both the truncated version of the Mahalanobis distances and the generalized Mahalanobis distance with low value of the parameter p give pretty good results, similar to the L^2-distance. But when we set a higher value for p, the clustering procedure begins to explore a much greater number of components and many weights of the distance d_p become closer to $1/\lambda_k$. For this reason, all the variability of the other components adds noise to the variability of the first components and the distance d_p become closer to the Mahalanobis distance than the L^2-distance. As we can note in Fig. 3.1(c), the number of misclassified curves increases very much, making the choice of setting a large value for the parameter p very bad. In fact, looking at Fig. 3.2, we can see how the centroids obtained by the k-means procedure and the real means differ a lot when we use the d_p distance with high p.

The next simulation concerns the case (b), where the means of the two samples X and Y differs only on the fifth component. In Fig. 3.3(a) there

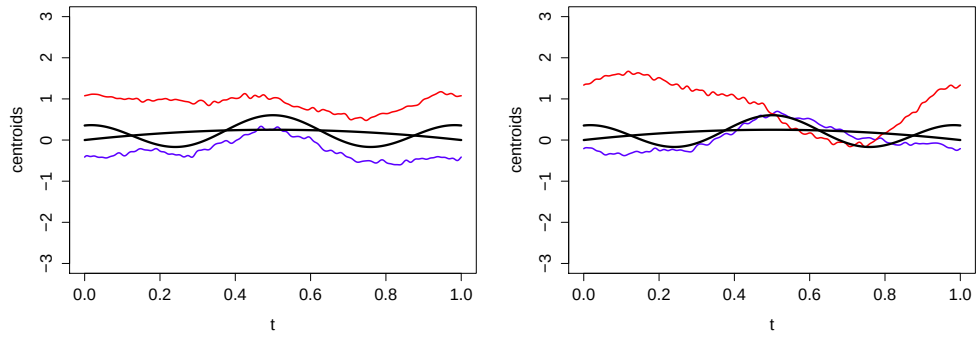


(a) **(b)**

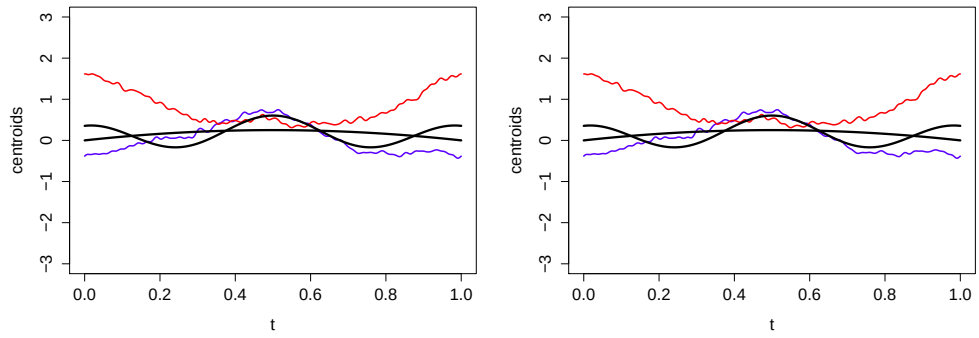


(c)

Figure 3.3: (a) Functional samples $X_1, \ldots, X_{50}$ (blue lines) and $Y_1, \ldots, Y_{50}$ (red lines).
(b) Means of the two samples $m_0(t)$ (blue line) and $m_1(t)$ (red line).
(c) Proportion of misclassified samples with the functional k-means for case (b) using the L^2-distance (yellow line), the truncated version of the Mahalanobis distance (green line) and the d_p distance (black line).



(a) d_{L^2} **(b)** d_M



(c) d_p, high p **(d)** d_p, medium p

Figure 3.4: Real means of the functional sample (black lines) and centroids obtained with the k-means clustering procedures (blue and red lines) with different distances.

(a) d_{L^2}

Cluster	X	Y
1	36	29
2	14	21

(b) d_M

Cluster	X	Y
1	35	29
2	15	21

(c) d_p high p

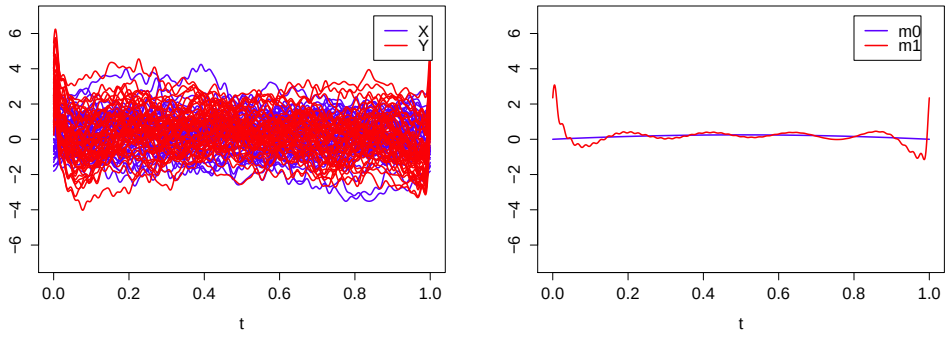
Cluster	X	Y
1	27	24
2	23	26

(d) d_p, medium p

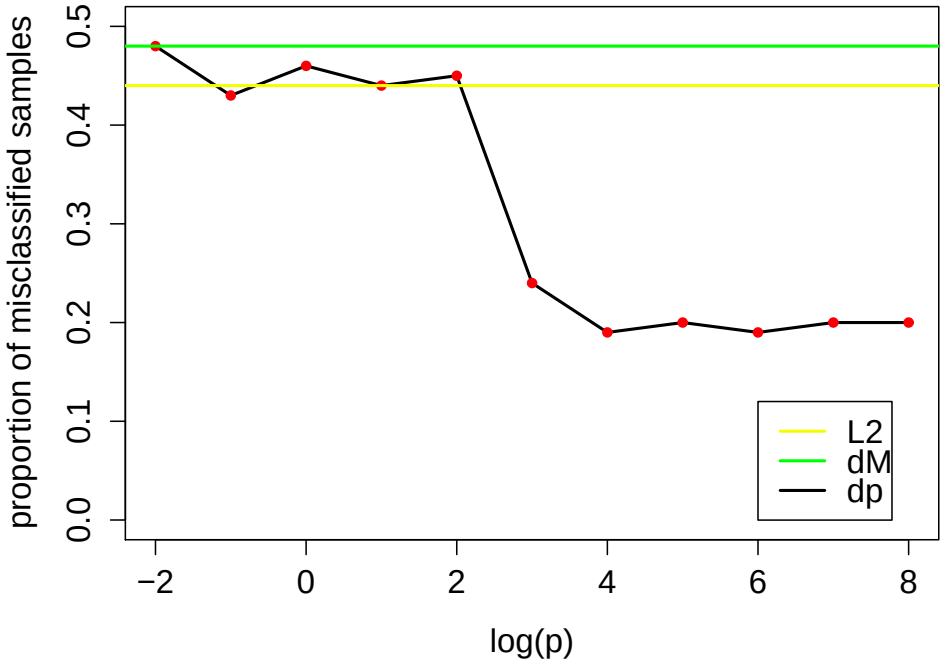
Cluster	X	Y
1	37	27
2	13	23

Table 3.2: Confusion matrices related to the functional k-means for the samples X and Y for case (b).

are the two samples X and Y along with their means in Fig. 3.3(b). In this case, it is not just a translation, in fact the macro-structure of the curve means is pretty similar. For this reason, the results obtained with the classical L^2-distance are not good at all, as it is shown in Table 3.2 and in Fig. 3.5(c). For what concerns the truncated version of the Mahalanobis distance, the result are similar to the L^2-distance. This is because we have considered only the fifth component, which cannot be seen from the d_M distance, since it is truncated up to the third component. Finally, we can see that in this case the d_p distance gives us some improvement. When we consider a low value of the parameter p the result is pretty similar to the d_M distance, since we are just giving almost all the weight to the first components, as the d_M distance does. When we increase the number of components considered, the d_p distance begins taking into account other components and the clustering procedure gives us better results. In fact, Fig. 3.3 shows us that for $\log(p) \in [0, 4]$, the number of misclassified curves decreases of about a 10%. At last, when the value of p is too large and almost all the values of the function $h_k(p)$ tend to 1, the distance d_p considers too many components and the method gives us again very bad results. Comparing the centroids obtained with the clustering procedure and the real means in Fig. 3.4, we can see that we never obtain very good results, but the d_p distance when $\log(p) \simeq 1$ definitely gives us the best ones.

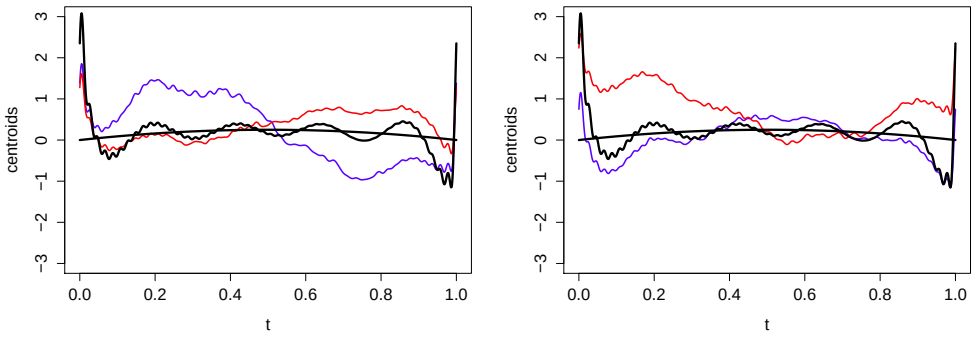


(a) (b)

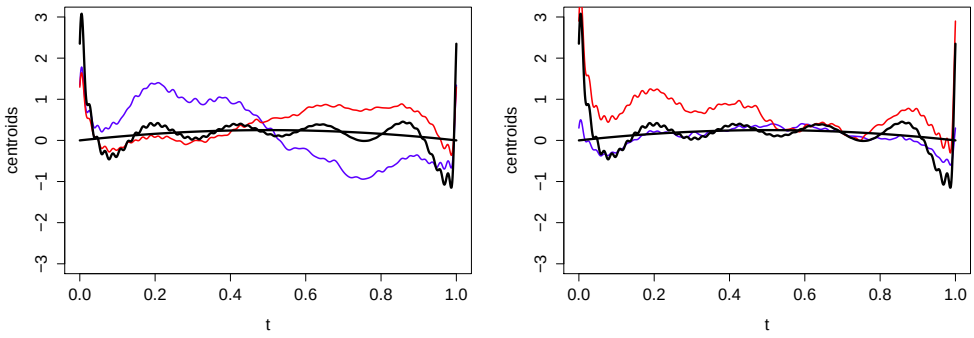


(c)

Figure 3.5: (a) Functional samples $X_1, \ldots, X_{50}$ (blue lines) and $Y_1, \ldots, Y_{50}$ (red lines).
(b) Means of the two samples $m_0(t)$ (blue line) and $m_1(t)$ (red line).
(c) Proportion of misclassified sample with the functional k-means for case (c) using the L^2-distance (yellow line), the truncated version of the Mahalanobis distance (green line) and the d_p distance (black line).



(a) d_{L^2} **(b)** d_M



(c) d_p, low p **(d)** d_p, high p

Figure 3.6: Real means of the functional sample (black lines) and centroids obtained with the k-means clustering procedures (blue and red lines) with different distances.

(a) d_{L^2}

Cluster	X	Y
1	35	29
2	15	21

(b) d_M

Cluster	X	Y
1	36	28
2	14	22

(c) d_p, low p

Cluster	X	Y
1	35	29
2	15	21

(d) d_p, high p

Cluster	X	Y
1	46	16
2	4	34

Table 3.3: Confusion matrices related to the functional k-means for the samples X and Y for case (c).

The last simulation in the univariate framework is given by case (c), where the two means differ for all the components except the first nine. Fig. 3.5(a) shows the two samples X and Y along with their means in Fig. 3.5(b). For the reasons that we have just explained, the L^2-distance and the truncated version of the Mahalanobis distance d_M do not work, because they cannot see the differences between the means; the same happens for what concerns the d_p distance with low p. When we set a large value for the parameter p, the distance begins to take into account all the components and it works pretty well, assigning about the 80% of the curves to the right group, as we can see in Table 3.3 and in Fig. 3.5(c). In fact, Fig. 3.6 shows us in this last case the centroids and the real means are more similar, so the procedure can catch the small differences in the micro-structure of the curves.

As we did in Chapter 2, we can conclude that also for what concerns the clustering procedures the d_p distance works in the same way and the choice of p is determined by our analysis. In fact, if we are comparing functional samples whose means have a very different macro-structure, it is probably best to use a distance which is similar to the L^2-distance and set a low value of the parameter p. When instead we want to compare samples whose means are very similar and with differences in the micro-structure, the L^2-distance begins not to work anymore and the choice of a high value for p is probably the best one.

3.2.2 Simulations in a multivariate framework

We want now to extend the results just presented in a multivariate framework. Let us consider two bivariate samples of i.i.d. curves $\mathbf{X}_i(t)$ and $\mathbf{Y}_i(t), i = 1, \ldots, n$, generated by independent stochastic processes with different means such that $\mathbf{X}_i(t), \mathbf{Y}_i(t) : L^2(I) \to \mathbb{R}^h$, where I is a compact interval of $\mathbb{R}$ and, without loss of generality, we choose $h = 2$.

Using the same framework of Subsection 3.2.1, we generate our samples through the Karhunen-Loève decomposition in the following way:

$$\mathbf{X}_i(t) = \mathbf{m}_0(t) + \sum_{k=1}^{\infty} \mathbf{Z}_{ki,1} \sqrt{\lambda_k} \vartheta_k(t) \quad i = 1, \ldots, n_1.$$

$$\mathbf{Y}_i(t) = \mathbf{m}_1(t) + \sum_{k=1}^{\infty} \mathbf{Z}_{ki,2} \sqrt{\lambda_k} \vartheta_k(t) \quad i = 1, \ldots, n_2.$$

where we set:

1. the time grid of 1000 equispaced points in $I = [0, 1]$;
2. truncation at $K = 200$ components;
3. the same sample sizes $n_1 = n_2 = 50$;
4. the mean of the first sample

$$\mathbf{m}_0(t) = \begin{pmatrix} t(1-t) \\ 4t^2(1-t) \end{pmatrix} \tag{3.2}$$

   while we set different values for the mean of the second sample;
5. the random variables $(\mathbf{Z}_{ki,1})_{k=1}^{\infty}$ $(\mathbf{Z}_{ki,2})_{k=1}^{\infty}$ are two sequences of bivariate standard normal variables with the same mean $\boldsymbol{\mu} = (0 \quad 0)^T$ and covariance operator

$$\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

6. $\{\lambda_k; k \geq 1\}$ is the sequence of eigenvalues of the covariance function v defined as follows:

$$\lambda_k = \begin{cases} \frac{1}{k+1} & \text{if } k \in \{1, 2, 3\} \\ \frac{1}{(k+1)^2} & \text{if } k \geq 4 \end{cases} \tag{3.3}$$

7. $\{\vartheta_k; k \geq 1\}$ is the sequence of eigenfunctions of the covariance function v defined as follows:

$$\vartheta_k = \begin{cases} \mathbf{1}_{\{t\in[0,1]\}} & \text{if } k = 1 \\ \sqrt{2}\sin(k\pi t)\mathbf{1}_{\{t\in[0,1]\}} & \text{if } k \geq 2,\ k \text{ even} \\ \sqrt{2}\cos((k-1)\pi t)\mathbf{1}_{\{t\in[0,1]\}} & \text{if } k \geq 3,\ k \text{ odd} \end{cases}$$

Starting from the eigenvalues $(\lambda_k)_k$ with the associated eigenfunctions $(\vartheta_k)_k$, we generated the curves in three different cases:

(a)

$$\mathbf{m}_1(t) = \mathbf{m}_0(t) + \begin{pmatrix} \sqrt{\lambda_1} \cdot \vartheta_1(t) \\ 4\sqrt{\lambda_1} \cdot \vartheta_1(t) \end{pmatrix} \tag{3.4}$$

(b)

$$\mathbf{m}_1(t) = \mathbf{m}_0(t) + \begin{pmatrix} \sqrt{\lambda_5} \cdot \vartheta_5(t) \\ 4\sqrt{\lambda_5} \cdot \vartheta_5(t) \end{pmatrix} \tag{3.5}$$

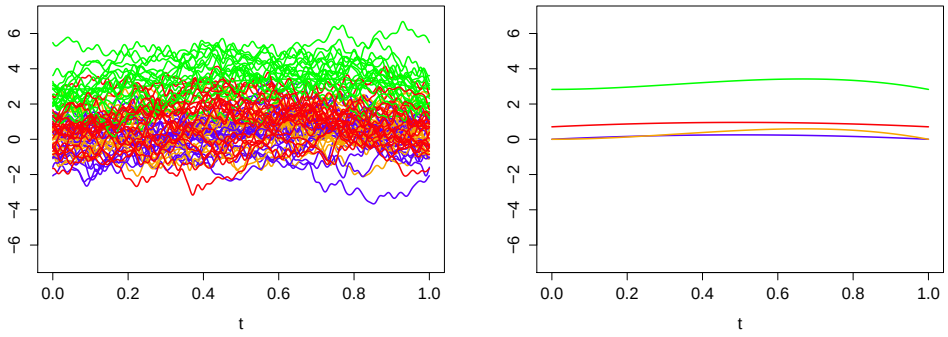
(c)

$$\mathbf{m}_1(t) = \mathbf{m}_0(t) + \begin{pmatrix} \sum_{k=10}^{\infty} \sqrt{\lambda_k} \cdot \vartheta_k(t) \\ 4\sum_{k=10}^{\infty} \sqrt{\lambda_k} \cdot \vartheta_k(t) \end{pmatrix} \tag{3.6}$$

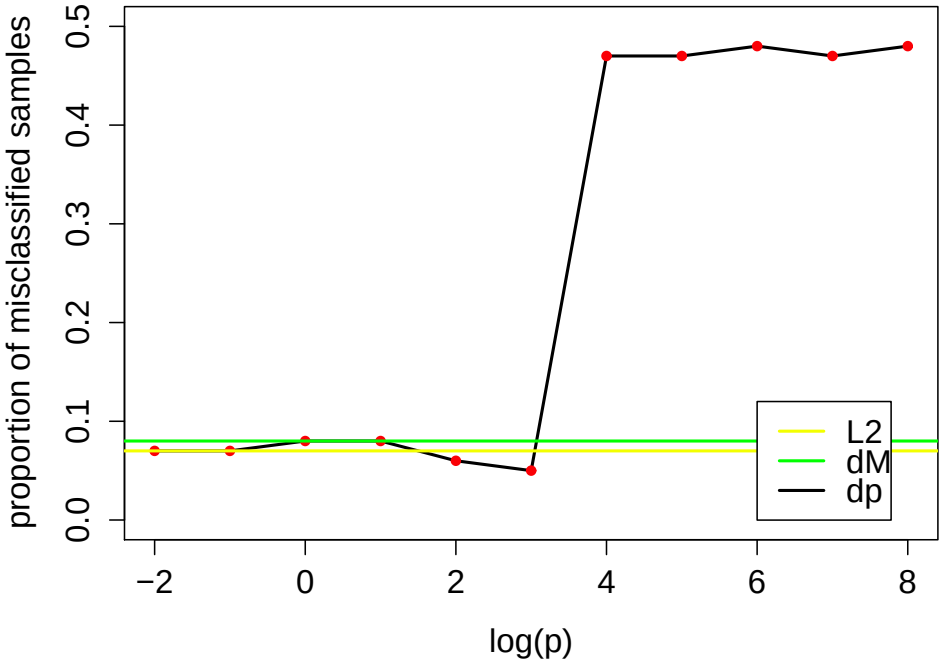
For each case we then computed a k-means algorithm based on the three different distances between the two samples used before. Since we are now in a bivariate framework, the distances will be different. Let us consider our two samples $\mathbf{X} = (X_1, X_2)^T$ and $\mathbf{Y} = (Y_1, Y_2)^T$. We used the empirical eigenvalues for each component $(\hat{\lambda}_k^{(1)})_k, (\hat{\lambda}_k^{(2)})_k$ with the associated eigenfunctions $(\hat{\vartheta}_k^{(1)})_k, (\hat{\vartheta}_k^{(2)})_k$ to compute the d_p distance as follows:

$$d_p(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{k=1}^{\infty} d_{M,k}^2(\mathbf{X}, \mathbf{Y}) \cdot h_k(p)}$$

where $d_{M,k}(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{h=1}^{2}(\langle X_h - Y_h, \hat{\vartheta}_k^{(h)}\rangle)^2/\hat{\lambda}_k^{(h)}}$ is the term representing the contribution of the Mahalanobis distance along the k^{th} component and $h_k(p) = \lambda_k/(\lambda_k + 1/p)$. As we did in the previous section, we consider other two competitors to compare our distances with: the truncated Mahalanobis distances (summing up $K = 3$ components, which

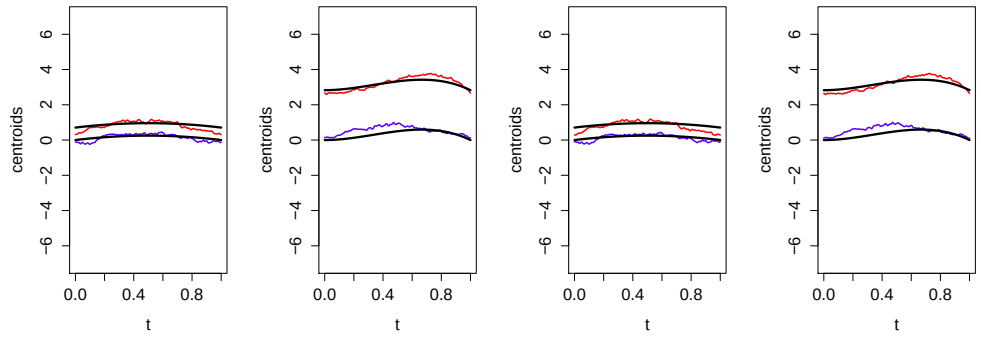


(a) **(b)**

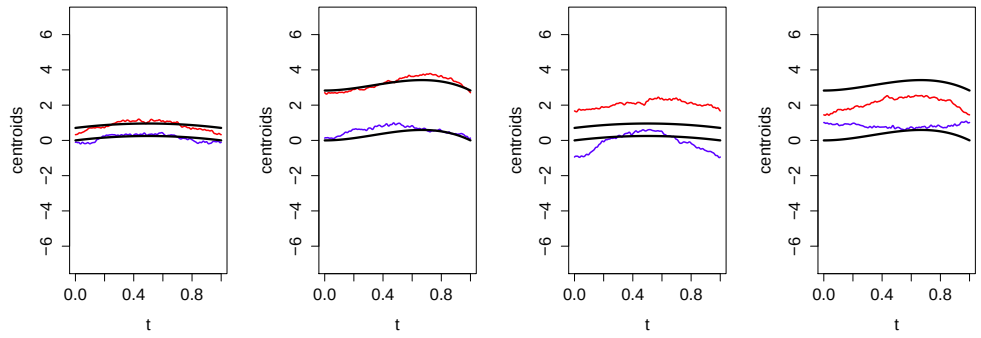


(c)

Figure 3.7: (a) Functional samples $\mathbf{X}_1, \ldots, \mathbf{X}_{50}$ (blue and orange lines) and $\mathbf{Y}_1, \ldots, \mathbf{Y}_{50}$ (red and green lines).
(b) Means of the two samples $\mathbf{m}_0(t)$ (blue and orange lines) and $\mathbf{m}_1(t)$ (red and green lines).
(c) Proportion of misclassified sample with the functional k-means for case (a) using the L^2-distance (yellow line), the truncated version of the Mahalanobis distance (green line) and the d_p distance (black line).



(a) d_{L^2} **(b)** d_M



(c) d_p, high p **(d)** d_p, medium p

Figure 3.8: Real means of the functional sample (black lines) and centroids obtained with the k-means clustering procedures (blue and red lines) with different distances.

(a) d_{L^2}

Cluster	X	Y
1	48	5
2	2	45

(b) d_M

Cluster	X	Y
1	48	6
2	2	44

(c) d_p, low p

Cluster	X	Y
1	48	4
2	2	46

(d) d_p, high p

Cluster	X	Y
1	27	24
2	23	26

Table 3.4: Confusion matrices related to the functional k-means for the samples $\mathbf{X}$ and $\mathbf{Y}$ for case (a).

described the most of the varaibility) and the classical L^2-distance, which can be written as:

$$d_M(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{k=1}^{3} d_{M,k}^2(\mathbf{X}, \mathbf{Y})}$$

$$d_{L^2}(\mathbf{X}, \mathbf{Y}) = \sqrt{\int_I (\mathbf{X}(t) - \mathbf{Y}(t))^2 dt}$$

In Fig. 3.7(a) there are the samples $\mathbf{X}$ (X_1 with blue lines, X_2 with orange lines) and $\mathbf{Y}$ (Y_1 with red lines, Y_2 with green lines) which differ only on the first component, along with their means in Fig. 3.7(b). In Fig. 3.7(c) we depicted instead a comparison among the different results obtained using the three distances. We can note that the results obtained in the multivariate framework confirm and strenghten the ones obtained in the univariate framework. In this case, the difference between the means is just a translation and the classical L^2-distance works very well, assigning almost the 95% of the data to the right group, as we can see in Table 3.4. For what concerns the other two distances, both the d_M distance and the d_p distance with low p work as well as the d_{L^2} distance. Setting a higher value for p become now a worst choice than in the univariate framework, since we are considering too many components and our algorithm can't distinguish the two samples at all. In Fig. 3.7, it is shown how the number of misclassified curves increase a lot, almost reaching the 50% of the curve

(a) d_{L^2}

Cluster	X	Y
1	31	20
2	19	30

(b) d_M

Cluster	X	Y
1	34	26
2	16	24

(c) d_p, high p

Cluster	X	Y
1	27	15
2	13	35

(d) d_p, $\log(p) = 0$

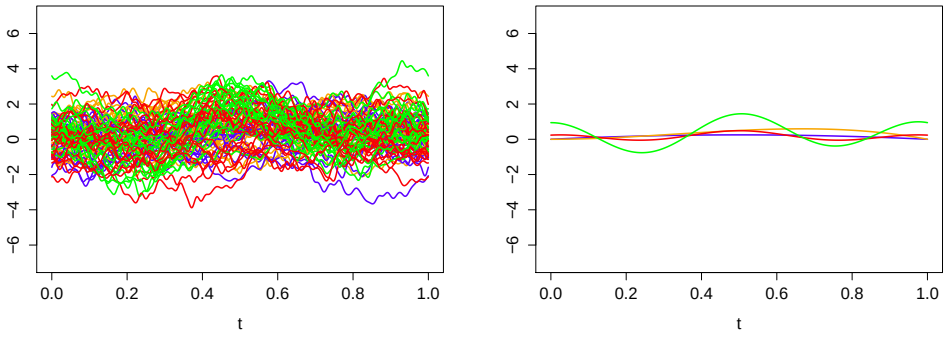
Cluster	X	Y
1	40	18
2	10	32

Table 3.5: Confusion matrices related to the functional k-means for the samples $\mathbf{X}$ and $\mathbf{Y}$ for case (b).

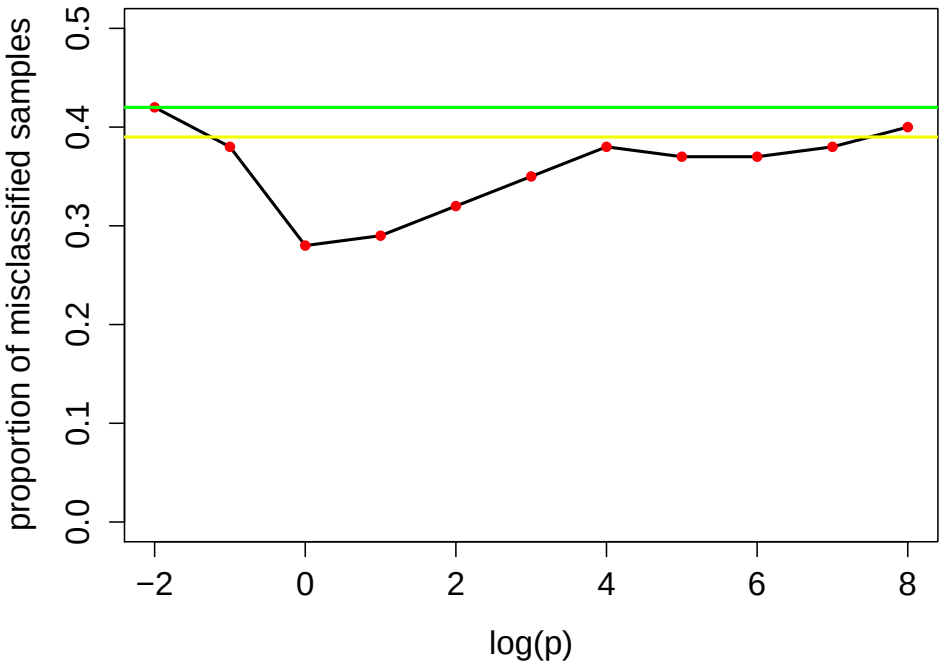
samples. The centroids in Fig. 3.8 are in fact the ones that differ the most from the real means of the samples.

The second simulation in the multivariate framework concerns the case (b), where the means of the two samples $\mathbf{X}$ and $\mathbf{Y}$ differ only for the fifth component. This difference cannot be seen by the d_M distance, since it is truncated up to the third one. In Fig. 3.9(a) we depicted the two samples along with their means in Fig. 3.9(b), which look pretty similar. For this reason, the results obtained for the d_{L^2} and the d_M distance are not very good (see Table 3.5). Comparing the Fig. 3.3(c) with Fig. 3.9(c), we can see that the proportion of misclassified samples when p becomes higher has a similar trend but in the multivariate framework it is generally lower; in fact, a higher number of curves is assigned to the right cluster. Even though for low values of the parameter p the distance d_p gives the same results obtained with the d_{L^2} and the d_M distance, when $\log(p) \in [-1, 3]$ the number of misclassified curves decreases, since the distance begins taking into account more components. This is probably the case in which the centroids are more similar to the real means, as it is shown in Fig. 3.10. However, as in the univariate framework, when p becomes higher, too many components are considered and we obtain again bad results.

Finally, the last simulation concerns case (c), where the two means differ for all the components except the first nine. In Fig. 3.11(a) there are the two samples $\mathbf{X}$ and $\mathbf{Y}$ and their means in Fig. 3.11(b). For the same

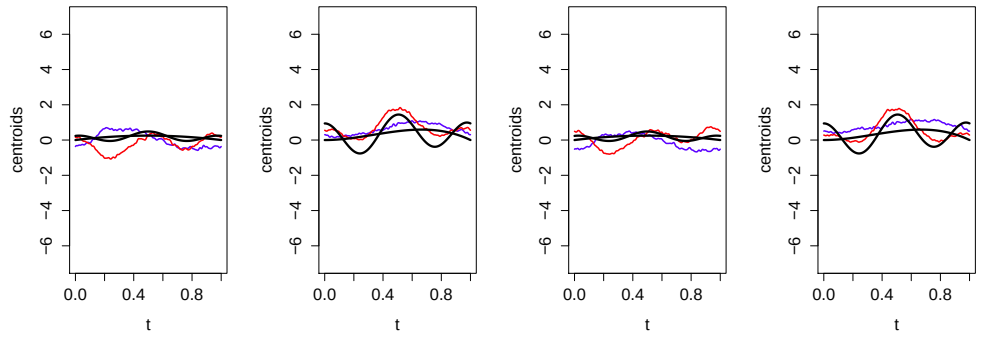


(a) **(b)**

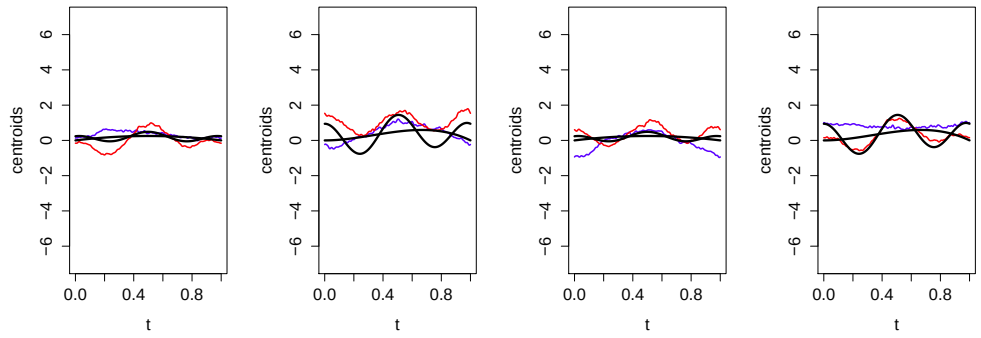


(c)

Figure 3.9: (a) Functional samples $\mathbf{X}_1, \ldots, \mathbf{X}_{50}$ (blue and orange lines) and $\mathbf{Y}_1, \ldots, \mathbf{Y}_{50}$ (red and green lines).
(b) Means of the two samples $\mathbf{m}_0(t)$ (blue and orange lines) and $\mathbf{m}_1(t)$ (red and green lines).
(c) Proportion of misclassified sample with the functional k-means for case (b) using the L^2-distance (yellow line), the truncated version of the Mahalanobis distance (green line) and the d_p distance (black line).



(a) d_{L^2} **(b)** d_M



(c) d_p, high p **(d)** d_p, medium p

Figure 3.10: Real means of the functional sample (black lines) and centroids obtained with the k-means clustering procedures (blue and red lines) with different distances.

(a) d_{L^2}

Cluster	X	Y
1	25	28
2	25	32

(b) d_M

Cluster	X	Y
1	30	22
2	20	28

(c) d_p, low p

Cluster	X	Y
1	28	21
2	22	29

(d) d_p, high p

Cluster	X	Y
1	50	1
2	0	49

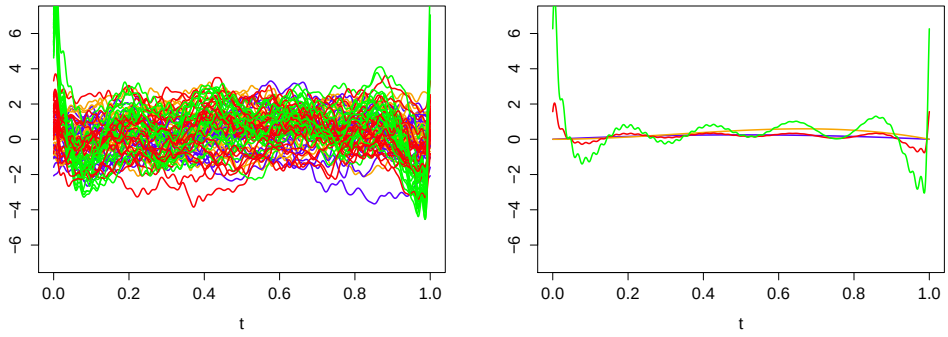
Table 3.6: Confusion matrices related to the functional k-means for the samples $\mathbf{X}$ and $\mathbf{Y}$ for case (c).

reason already explained for the univariate framework, when the value of p is low the clustering procedure do not work and the results are as bad as for the d_{L^2} and d_M distances. When the value of p increases, the procedure with the d_p distance gives us almost perfect results (see Fig. 3.11(c) since we are taking into account every component. In Fig. 3.12 we can see how the centroids are almost the same as the real means, when we use the d_p distance setting a large value for the parameter p.

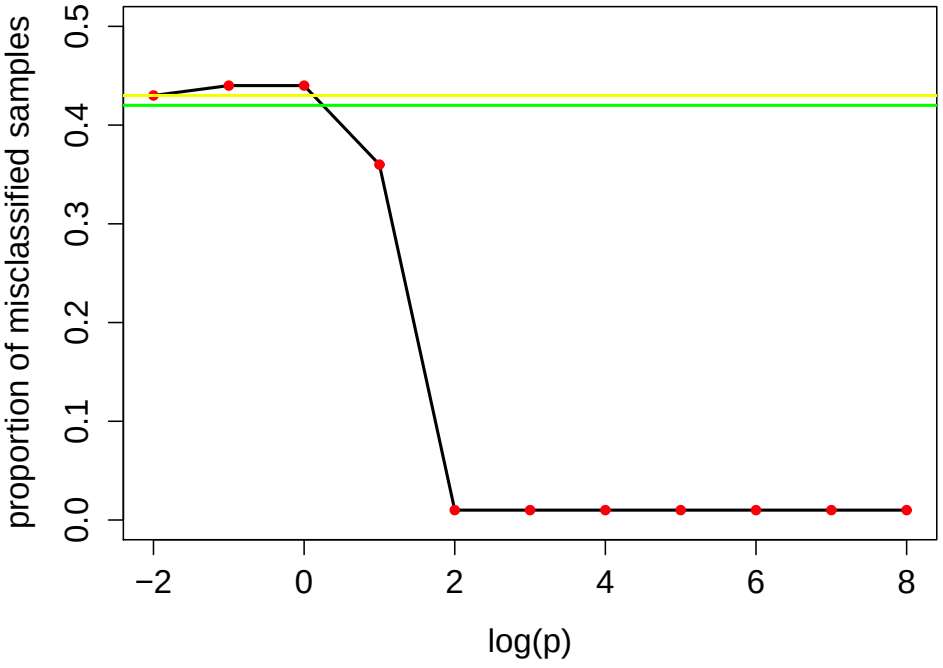
We have now showed that all the results obtained in the univariate framework also apply in the multivariate framework. We can even say that if we have multivariate functional data, the clustering procedure improves but the choice of p is again determined by our analysis. As in the univariate framework, if the functional samples have a very different macro-structure it is better to choose a low value for p, since we want a distance more similar to the classical L^2-distance. If the curves have instead pretty similar means but differences in the micro-structure, it is better to choose a high value for p, since we want a distance which is more similar to the Mahalanobis distance.

3.3 Applications on a real dataset

In this last section of the chapter we want to apply our functional k-means to a real case study on ECGs. We give some background theory

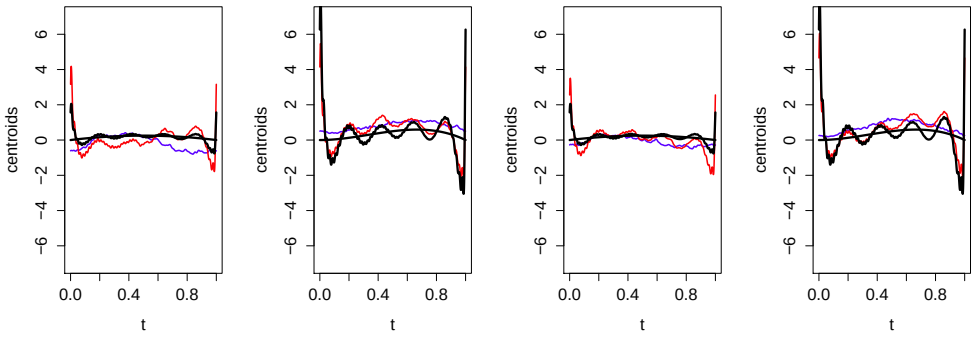


(a) (b)

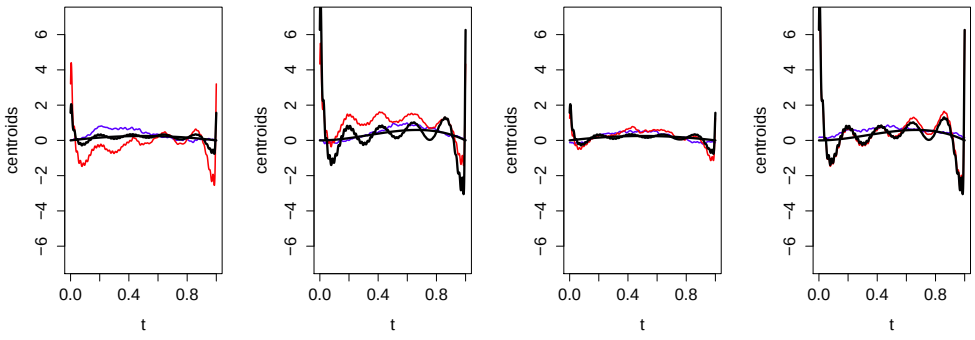


(c)

Figure 3.11: (a) Functional samples $\mathbf{X}_1, \ldots, \mathbf{X}_{50}$ (blue and orange lines) and $\mathbf{Y}_1, \ldots, \mathbf{Y}_{50}$ (red and green lines).
(b) Means of the two samples $\mathbf{m}_0(t)$ (blue and orange lines) and $\mathbf{m}_1(t)$ (red and green lines).
(c) Proportion of misclassified sample with the functional k-means for case (c) using the L^2-distance (yellow line), the truncated version of the Mahalanobis distance (green line) and the d_p distance (black line).



(a) d_{L^2} **(b)** d_M



(c) d_p,low p **(d)** d_p, high p

Figure 3.12: Real means of the functional sample (black lines) and centroids obtained with the k-means clustering procedures (blue and red lines) with different distances.

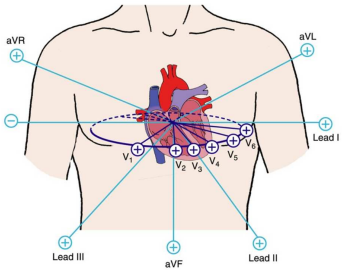


Figure 3.13: 12-lead ECG position

on ECG, before presenting the dataset and showing the obtained results.

3.3.1 Electrocardiography (ECG)

Electrocardiography is a transthoracic recording of the electrical activity of the heart over a period of time as it is captured and externally recorded through electrodes placed on the skin. These electrodes detect the tiny electrical changes that arise from the depolarization of the heart muscle during each heartbeat. The standard ECG registration is the 12-lead ECG system, in which the overall magnitude of the heart's electrical potential is measured from 12 different angles and is recorded over a period of time, see Einthoven (1908) and Einthoven *et al.* (1950). The first sets of leads are the standard limb leads, also known as the bipolar leads, denoted by I, II and III. There are then the augmented unipolar leads, denoted by aVR, aVL and aVF. The last ones are the chest leads denoted by V1, V2, V3, V4, V5 and V6. The placement of the electrodes for the leads can be seen in Fig. 3.13.

The first two sets of leads are derived from the same three measurement points, so we can omit the augmented unipolar leads. Furthermore, we can consider only two bipolar leads since, for the Einthoven's law, the third one can be obtained as the sum of the other two. Then, the ECG traces which are analyzed in the following chapters will consist of leads I, II, V1, V2, V3, V4, V5 and V6 only. The stylized shape of a physiological single beat is shown in Fig. 3.14. In the same figure, the main relevant points, segments and waves are highlighted. In the stylized signal, the deflections are listed alphabetically starting with letter P, which represents atrial depolarization; it follows then the QRS-complex which is caused by ventricular depolarization. At last, we have the ventricular repolarization

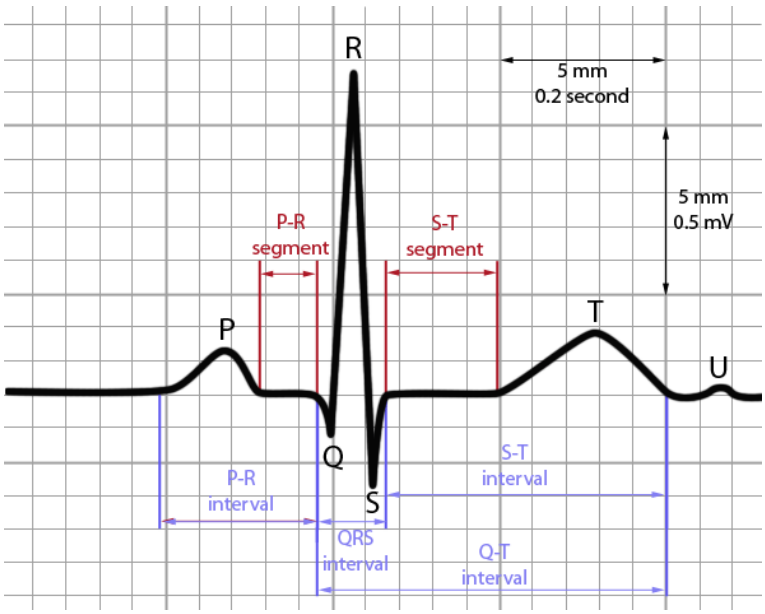


Figure 3.14: Stylized shape of a physiological single beat, recorded on ECG graph paper (the main relevant points, segments and waves are highlighted).

which generates the T-wave and mask the atrial repolarization (for further details, see Scher and Young (1957)). The direction of the wave of depolarization is named the *heart electrical axis*.

Bundle Branch Blocks

The heartbeat is triggered by electrical impluses that travel down a special pathway through the heart. The impulse starts in a small bundle of specialized cells located in the upper right atrium, called the sinoatrial node (which is the heart's natural pacemaker, 1 in Fig. 3.15). The electrical activity spreads through the walls of the left and right atria, summating at the atrioventricular node (2 in Fig. 3.15). From here, the impulse travels down the bundle of His (3 in Fig. 3.15) and divides itself into the right and left bundle branches (4 and 10 in Fig. 3.15). The right bundle branch contains one fascicle while the left bundle branch subdivides into two fascicles: the left anterior fascicle and the left posterior fascicle (6 and 5 in Fig. 3.15). Ultimately, the fascicles divide further into millions of fibers which send impulses to the individual cardiac cells, allowing for rapid,

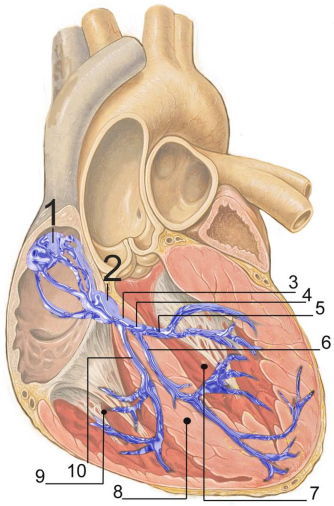


Figure 3.15: Conduction system of the heart: 1, sinoatrial node; 2, atrioventricular node; 3, bundle of His; 4, left bundle branch; 5, left posterior fascicle; 6, left anterior fascicle; 7, left ventricle; 8, ventricular septum; 9, right ventricle; 10, right bundle branch.

coordinated physiologic depolarization of the ventricles. Bundle branch injuries result in altered pathways for ventricular depolarization. As a result, there is a loss of ventricular synchrony, ventricular depolarization is prolonged, and there may be a corresponding drop in cardiac output (see Guyton and Hall (2006)). Depending on the anatomical location of the defect which leads to a bundle branch block, the blocks are further classified into right bundle branch block (RBBB) and left bundle branch block (LBBB).

From a clinical perspective, a RBBB typically causes prolongation of the last part of the QRS-complex, and may shift the heart's electrical axis slightly to the right. A LBBB widens the entire QRS-complex, and in most cases shifts the heart's electrical axis to the left. Another usual finding with BBBs is an appropriate T-wave discordance: this means that the T-wave will be deflected in the opposite direction to the terminal deflection of the QRS-complex.

What it is in our interest is that, from a statistical point of view, there are shape modifications that are induced on the ECG curves, and we shall investigate them only from a statistical perspective, without using clinical criteria.

3.3.2 Smoothing and registration of the ECGs

The dataset provided by Mortara-Rangoni S.r.l. contains, among others, the ECG signals of $n = 2102$ subjects, among which 1602 are healthy while 500 are affected by BBBs (224 RBBBs and 276 LBBBs). Every signal consists of 8 curves, one for each lead, so we have a multivariate functional dataset. The electrocardiographic signals have been registered without any processing, so they are just noisy and discrete observations of the function describing the ECG trace of each patient. Moreover, each patient has his own 'biological' time, i.e. the same event of the heart dynamics may occur at different times for different patients; that is why the morphological change due to this difference in timings is misleading from a statistical perspective. To address these two problems, which are common in functional data analysis, it has been necessary to perform the procedures presented in Chapter 1, the data *smoothing* and *registration*.

Among all the possible smoothing methods, the most suitable for the data was the wavelet basis method with universal threshold, which was able to capture strongly localized ECG features (peaks, oscillations,...). Thus, starting from the vectorial raw signal, it was estimated the vectorial function

$$\mathbf{f}_i(t) = (I_i(t), II_i(t), V1_i(t), V2_i(t), V3_i(t), V4_i(t), V5_i(t), V6_i(t)),$$

and its derivatives, for each patient $i = 1, ..., n$. As in most smoothing methods based on wavelet expansion, the grid which has been used consists of 2^J points, $J \in \mathbb{N}$. In this particular case, we have the time grid [0 ms, 550 ms] which consists of $2^{10} = 1024$ observation points.

As we have previously said, to face the second problem it has been used a registration procedure based on landmarks, which are points of the curve that can be associated with a specific biological time (see Ieva *et al.* (2013), Indino (2015)). For each patient $i = 1, ..., n$ there has been the identification of the P-wave($P^i_{onset}, P^i_{offset}$), the QRS-complex ($QRS^i_{onset}, QRS^i_{offset}$), the peak for the T-wave (T_{peak}) and the peak for the R wave (R_{peak}). These landmarks have been used to register all the leads and for each patient i it has been looked for a warping function h_i obtained by cubic spline interpolation.

3.3.3 k-means

Let us consider the time interval I=[0 ms, 550 ms] which consists of $2^{10} = 1024$ observation points and analyse the n patients according to a functional k-means clustering procedure, where we consider

$$\mathbf{F}(t) = \{F^r(t)\}_{r=1}^{8} = \{I(t), II(t), V1(t), V2(t), V3(t), V4(t), V5(t), V6(t)\}$$

such that the eight leads $\mathbf{F}_i(t) : I \to \mathbb{R}^8$, for patients $i = 1, \ldots, n$, are simultaneously clustered. We assume that $\mathbf{F}_i(t) \in L^2(I; \mathbb{R}^8)$. Since we consider all the eight leads simultaneously in the analysis, this clustering procedure can be called *multivariate functional k-means*.

For computational reasons, we focus our attention on a reduced number of curves. We consider $n = 700$ subjects, where 400 of them are healthy, 150 are affected by LBBBs and 150 are affected by RBBBs. We then compute the empirical eigenvalues for each lead $(\hat{\lambda}_k^r)_k$ with the associated eigenfunctions $(\hat{\vartheta}_k^r)_k$, with $k = 1, \ldots, n$ and $r = 1, \ldots, 8$. In this particular case, we can define the generalized Mahalanobis distance between two random ECG curves $\mathbf{F}_i(t)$ and $\mathbf{F}_j(t)$, with $i \neq j$, using the definition (1.4) where the term the contribution of the Mahalanobis distance along the k^{th} component can be written as follows:

$$d_{M,k}(F_i^r(t), F_j^r(t)) = \sqrt{\frac{[\int_I (F_i^r(t) - F_j^r(t)) \cdot \vartheta_k(t)]^2}{\lambda_k}}$$

and $h_k(p) = \lambda_k/(\lambda_k + 1/p)$. To perform comparisons and to test the robustness of our clustering procedure we have considered the same distances used in the simulation between a pair of ECG traces, d_M and d_{L^2}, which can be written in this case as follows:

$$d_M(\mathbf{F}_i(t), \mathbf{F}_j(t)) = \sqrt{\sum_{r=1}^{8} \sum_{k=1}^{3} d_{M,k}^2(F_i^r(t), F_j^r(t))dt} \tag{3.7}$$

$$d_{L^2}(\mathbf{F}_i(t), \mathbf{F}_j(t)) = \sqrt{\sum_{r=1}^{8} \int_I (F_i^r(t) - F_j^r(t))^2 dt} \tag{3.8}$$

The final silhouette plots obtained by clustering our sample of ECG traces according to a multivariate functional k-means procedure with the distance d_p, where $p = 10^4$ and $k = 2, 3, 4, 5$, are shown in Fig. 3.16. As we can see in the figure, the grouping structure that is obtained by setting

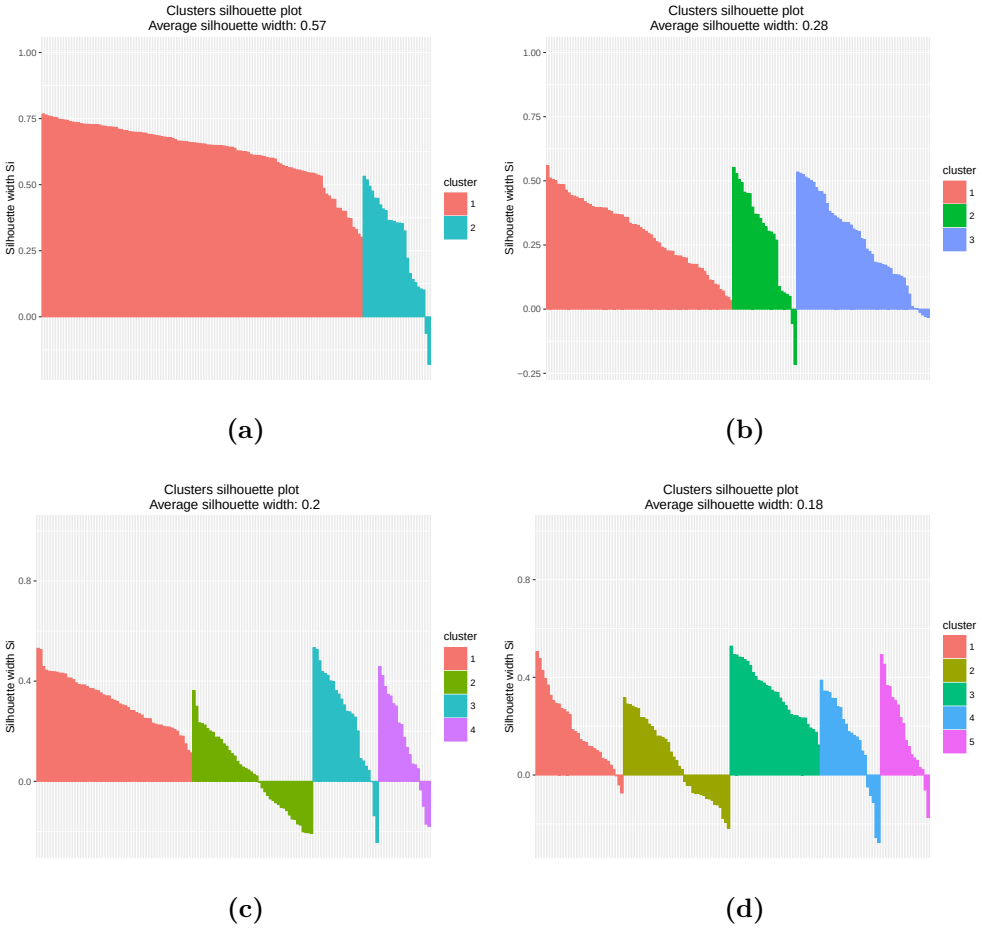


Figure 3.16: Silhouette plots of the clustering result obtained via the multivariate functional k-means procedure, setting (a) k=2, (b) k=3, (c) k=4 and (d) k=5 and with distance d_p: the data are ordered according to an increasing value of silhouette within each cluster and are coloured according to the cluster assignment

(a) L^2 distance

Cluster	Healthy	LBBB	RBBB
1	342	16	18
2	55	134	32
3	3	0	100

(b) d_M distance

Cluster	Healthy	LBBB	RBBB
1	336	20	10
2	61	130	36
3	3	0	104

(c) d_p distance, low p

Cluster	Healthy	LBBB	RBBB
1	343	16	17
2	54	134	33
3	3	0	100

(d) d_p distance, high p

Cluster	Healthy	LBBB	RBBB
1	357	19	14
2	40	131	28
3	3	0	108

Table 3.7: Confusion matrices related to the functional k-means for the ECG traces.

$k = 3$ seems the best, both in terms of silhouette profile and in terms of wrong assignments. A similar result is obtained by measuring the distance between curves with the d_M or the d_{L^2} distances; however, the procedure seems to detect the best grouping structure when we use the d_p distance and we set a high value of the parameter p. We thus set $k^* = 3$.

Because of the high computational cost due to the computation of the d_p distance, which have to take into account a great number of components, the code has been parallelized to greatly reduce the computational time, using the R-packages *doParallel* and *foreach* (for further details for both packages, see Revolution Analytics and Steve Weston (2015)). The obtained results of our 3-means multivariate clustering procedure, using the three distances, are showed in the confusion matrices in Table 3.7. We a posteriori identified the cluster with the greater number of physiological ECG traces as the one containing the healthy subjects. Subsequently, to distinguish the clusters corresponding to the pathological traces, we first selected the cluster containing the maximum number of pathological traces of the same kind and at last the remaining cluster.

Looking at the four confusion matrices, we can immediately note that we obtained pretty good results, which differ a little depending on the distance tested. As obtained in simulation (c) of the previous section and

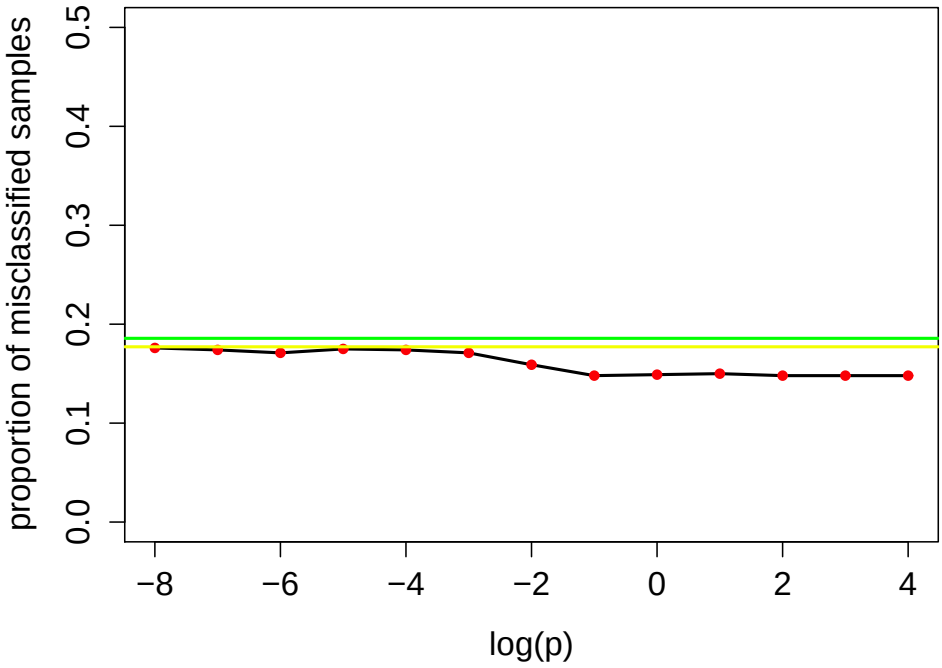


Figure 3.17: Proportion of misclassified samples with the functional k-means for the ECG dataset using the L^2 distance (yellow line), the truncated version of the Mahalanobis distance (green line) and the d_p distance (black line).

	d_{L^2}	d_M	d_p, low p	d_p, high p
Mean $cost_{CV}$	0.1281	0.1357	0.1205	0.0985
Standard deviation $cost_{CV}$	0.1244	0.1316	0.1286	0.1145

Table 3.8: Mean misclassification costs and standard deviations computed over 20 repetitions of the cross-validation procedure.

as we can see in Fig. 3.17, when the value of the parameter p in the d_p distance goes over a certain value, there is a change in the number of misclassified curves. In particular, in this case we go from about 81% of well-classified subjects to about 86%, which is a good improvement. Then, we can state that the generalized Mahalanobis distance with high value of p is probably the best choice, even though we have a higher computational cost. Remembering what we saw in the tests in the previous chapter and in the clustering simulations of the previous section, this is probably because of the difference in the micro-structure of the ECG curves, which is better identified by the d_p distance, as it is the best approximation of the Mahalanobis distance in a functional framework.

In Fig. 3.18 the ECG curves of the considered subjects are shown, for each one of the 8 leads, with a different color for each cluster (green for the healthy subjects, red for the RBBBs, blue for the LBBBs). Looking at the centroids in Fig. 3.19 we can highlight the main differences between the healthy subjects and the ones affected by Bundle Branch Blocks. The LBBBs have a larger QRS complex and a greater intensity of the R-peak. Futhermore, we can clearly see how in the centroid related to lead I, V1 and V2, there is an inversion of the T wave. For what concerns the RBBBs subjects, we can see in leads I, V1, V5 and V6 how the S wave is deeper than in the healthy subjects; in these cases the centroids show a larger QRS-complex.

We have now defined a semi-automatic diagnostic tool based on the multivariate functional k-means algorithm; in fact, after our clustering procedure we obtained a set of k centroids, representative of each cluster, which can be used as reference signals to compare a new ECG trace. Using these, we could have an immediate hint on a new patient's diagnosis by smoothing his ECG trace, registering it and finally assigning it to the group that is characterized by the nearest centroid.

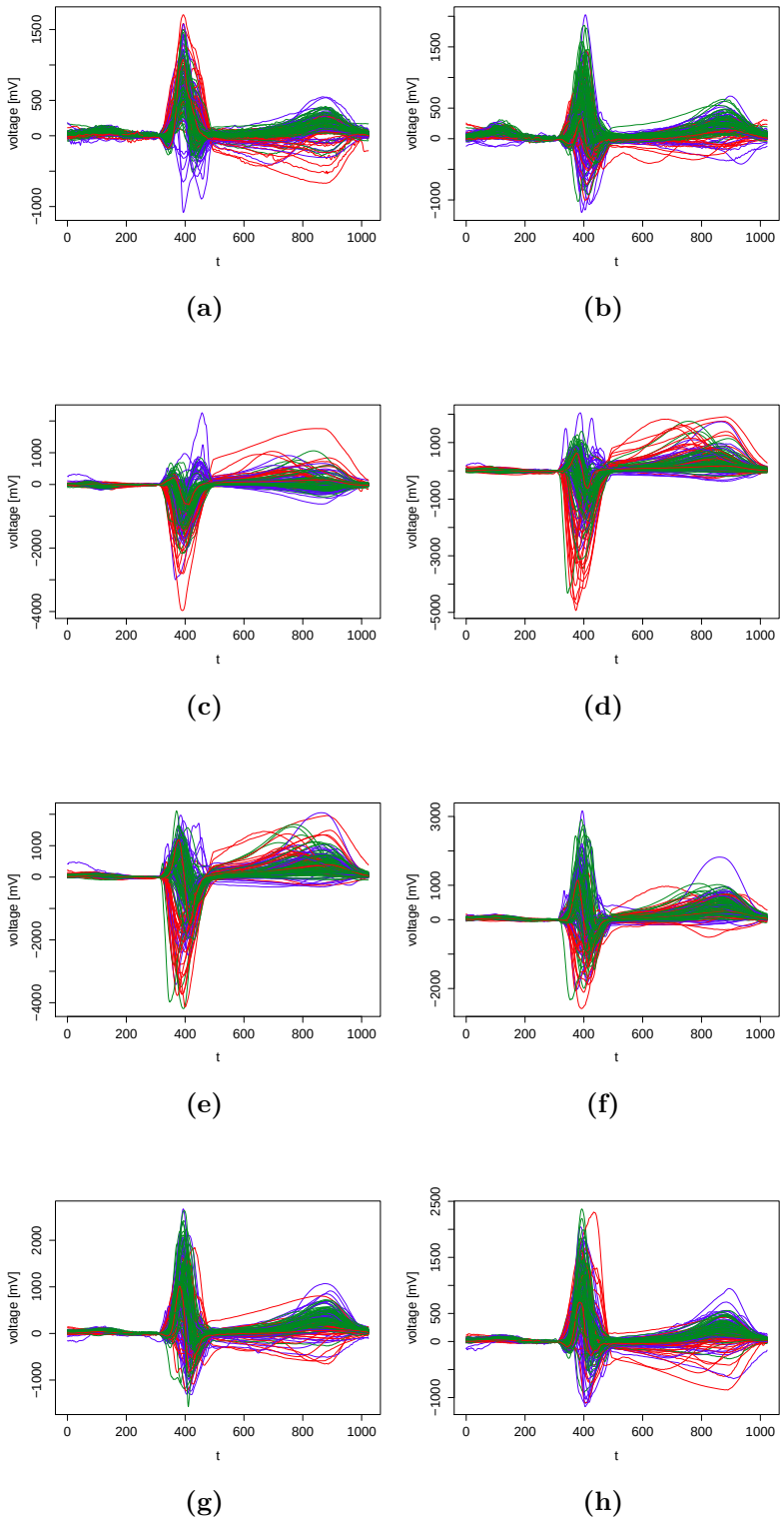


Figure 3.18: ECG leads assigned to each cluster (green for the healthy subjects, blue for the LBBBs, red for the RBBBs).

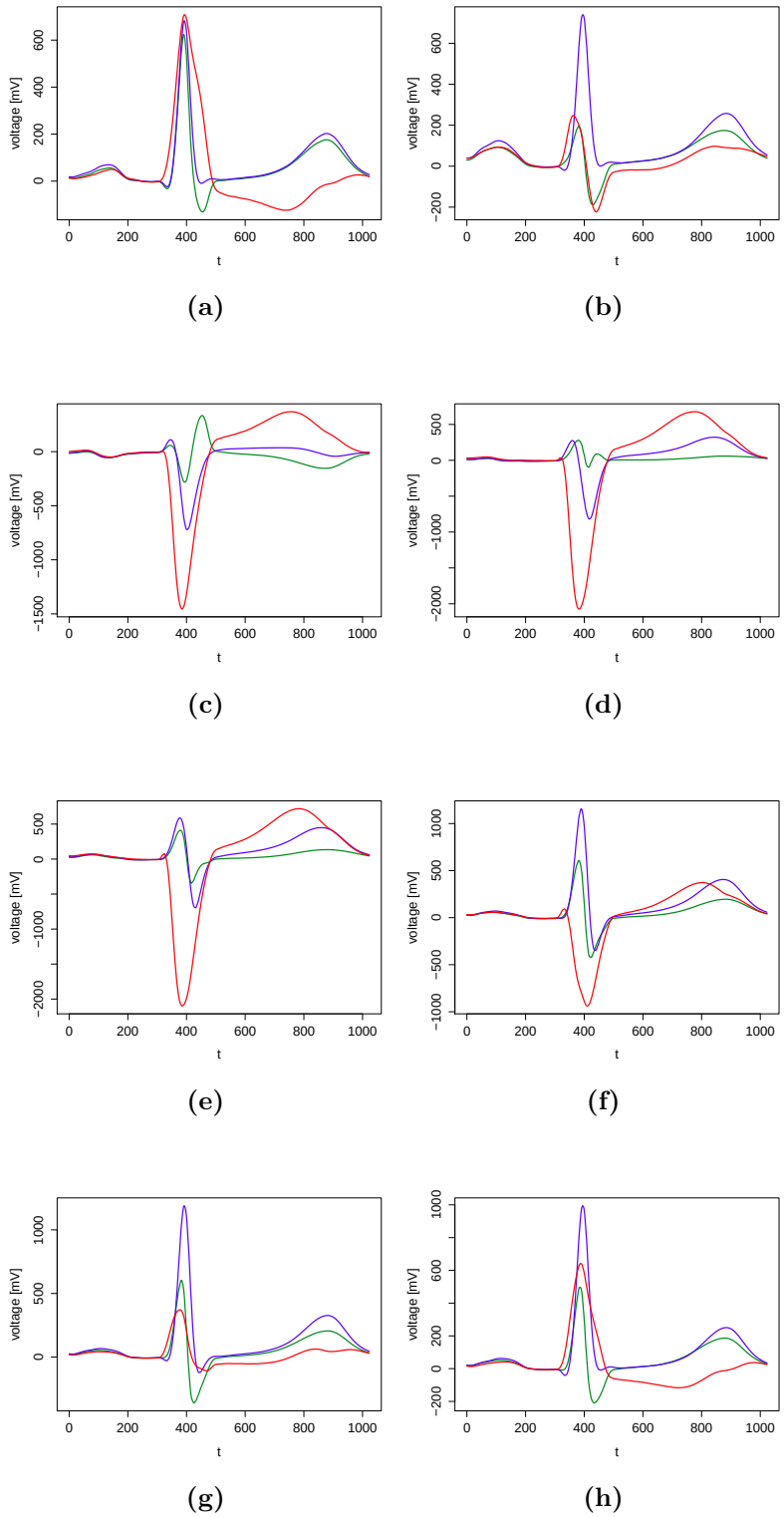


Figure 3.19: ECG centroids of the three clusters for each lead (green for the healthy subjects, blue for the LBBBs, red for the RBBBs).

(a) L^2 distance

Cluster	Healthy	LBBB	RBBB
1	68.4	3.2	3.6
2	11	26.8	6.4
3	0.6	0	20

(b) d_M distance

Cluster	Healthy	LBBB	RBBB
1	67.2	4	2
2	12.2	26	7.2
3	0.6	0	20.8

(c) d_p distance, low p

Cluster	Healthy	LBBB	RBBB
1	68.6	3.2	3.4
2	10.8	26.8	6.6
3	0.6	0	20

(d) d_p distance, high p

Cluster	Healthy	LBBB	RBBB
1	71.4	3.8	2.8
2	8	26.2	5.6
3	0.6	0	21.6

Table 3.9: Cross Validation results obtained on a test set of 120 patients, by application of the multivariate functional three-means clustering algorithm, with different choices of the distance between ECGs.

For this reason, it is important to evaluate the *misclassification cost* for this procedure, with the choice of the different functional distances. In order to do this, we perform a *cross-validation analysis* by randomly choosing among our $n = 700$ subjects, a training set of 320 healthy subjects, 120 RBBBs and 120 LBBBs, for a total of $n_{training} = 560$ curves and perform a multivariate functional three-means clustering on the training set selected. At last, we consider the remaining $n_{test} = 140$ curves and we assign each of them to the cluster whose centroid is the nearest, according to the three distances d_p, d_M and d_{L^2}. Given the patients' disease classification, we compute the misclassification cost by using the index

$$\text{cost}_{CV} = \frac{\lambda_1 \text{misc}_N + \lambda_2(\text{misc}_{RN} + \text{misc}_{LN}) + \lambda_3(\text{misc}_{RL} + \text{misc}_{LR})}{n_{test}} \tag{3.9}$$

where we denote by misc_N the number of healthy subjects assigned to a pathological cluster, while misc_{RN} and misc_{LN} are the number of patients respectively affected by RBBB and LBBB who are assigned to the cluster of healthy patients. misc_{RL} and misc_{LR} represent the number of patients whose ECGs are detected as pathological but are assigned to the wrong pathology. Finally, the parameters $\lambda_1, \lambda_2, \lambda_3$ are misclassification weights, which are chosen according to the suggestion of the clinicians,

who believe that assigning a BBB patient to the cluster of healthy patients is approximately four times more serious than treating a healthy subject as pathological, which is instead twice as serious as assigning a RBBB patient to the LBBB cluster and vice versa. To determine the values of the weights, it has introduced a further request: the cost_{CV} has to be equal to 1 in the worst case, when all healthy subjects are classified as BBB and all BBB subjects are classified as healthy. This choice leads to the following values: $\lambda_1 = 0.4270, \lambda_2 = 1.7079, \lambda_3 = 0.2135$.

The procedure has been repeated 20 times, computing each time the misclassification cost according to equation (3.9). Table 3.8 shows the mean and standard deviation computed along the 20 cross-validation repetitions. Once again, the distance d_p with higher p seems to give best results, confirming that is probably worth studying the micro-structures of ECG curves, looking at all the components of the functional data. We also show the average confusion matrices for each of the distances considered, which have been obtained by taking the mean of the confusion matrices along the cross-validation repetitions. The four average confusion matrices obtained for the different distances are shown in Table 3.9. We can see that the d_p distance with higher value of p gives us the best results, confirming the considerations that we have made while looking at the results shown in Table 3.7. .

Conclusions

In the first part of this work, we showed some inferential procedures to test difference between means of functional data. The inference is based on a suitable generalization d_p of the Mahalanobis distance to the Hilbert space L^2. We showed through simulations that the behaviour of d_p distance between a sample mean and a mean function $m(t) \in L^2(I)$ mostly depends on a tuning parameter p, which must be chosen depending on the type of analysis we want to perform.

In the second and main part of our work, we proposed a statistical framework for the analysis and classification of functional data. To analyse morphological information, we carried out several simulations using a functional k-means procedure, both in the univariate and multivariate framework, observing the same behaviour as in the testing procedures. We performed some comparisons of our clustering procedure considering two more distances and we can conclude that, depending on the value of the tuning parameter p, the results are not always the same. If the clusters have very different means, with a different macro-structure, it is convenient to choose a low value for the parameter p, since in this case the d_p distance is much similar to the L^2-distance. On the contrary, when the means differ only for the micro-structure's features, the d_p distance setting a high value of the parameter p gives the best results. In general, the parameter p must be chosen according to the type of analysis we want to make.

At last, we applied our clustering procedure to a real case study on ECGs. We first performed a measure of the goodness of the clustering results and in all cases considered the optimal number of clusters is set equal to 3. The confusion matrices resulting from our classification framework showed effective results, especially when we use the d_p distance and set a high value of p; then, we can conclude that it is worth to study the micro-structures of ECG curves, looking at all the components of the

functional data. Thus, we proposed a classification procedure which uses group centroids as reference signals; performing a cross-validation analysis to evaluate the misclassification cost, the procedure using the d_p distance with high value of p gave us again the best results.

Even though we obtained very good results using a multivariate functional k-means, it could be worth to implement other clustering techniques using the d_p distance. A more general way could be for instance to use an EM algorithm for Gaussian Mixtures. Alternatively, it could be good to try a different approach, using another clustering procedure as for example the hierarchical clustering. Besides, our clustering procedure could be adapted and applied to different types of functional data, as for example the trajectories of càdlàg processes, on which the procedures based on the classical L^2 distance do not work.

Appendix A

Code

Listing A.1: Code to perform test simulations based on the d_p distance.

```
iter <- 10^3
p <- 1000
Time <- 500
t <- seq(0,1,length=Time)
m0 <- t*(1-t)

integrale <- function(ascissa,integr){
  return(sum((integr[-1]+integr[-length(integr)])*diff(ascissa)/2))
}

K <- 100
m1<-NULL
h <- rep(0,K)

for (k in 1:10) {
  if (k%%2==0)
    theta <- sqrt(2)*sin(k*pi*t)
  else if (k%%2!=0 && k!=1)
    theta <- sqrt(2)*cos((k-1)*pi*t)
  else
    theta <- rep(1,Time)
  m1 <- rbind(m1,m0+0.03*theta)
}

lambda <- rep(0,K)
theta <- matrix(0,K,Time)
for (k in 1:K) {
  lambda[k] <- 1/(k+1)^4
  h[k]<- 1-exp(-lambda[k]*p)
  if (k%%2==0)
    theta[k,] <- sqrt(2)*sin(k*pi*t)
  else if (k%%2!=0 && k!=1)
    theta[k,] <- sqrt(2)*cos((k-1)*pi*t)
  else
    theta[k,] <- rep(1,Time)
}
chsq <- rep(0,K)
for (i in 1:K) {
  chsq[i] <- rchisq(K,1)%*%h
}
q.xi_sq <- quantile(chsq,0.95)
q.xi_sq

power <- rep(0,10)
cont <- rep(0,10)
for (l in 1:10) {
  d_p2 <- rep(0,iter)
  for (j in 1:iter) {
```

```
    u <- matrix(0,n,Time)
    for (i in 1:n) {
      s <- rep(0,Time)
      for(k in 1:K) {
        s <- s + sqrt(lambda[k])*(rnorm(1)*theta[k,])
      }
      u[i,] <-s
    }
    x<-matrix(0,n,Time)
    for (i in 1:n){
      x[i,] <- u[i,] + m1[l,]
    }
    x.mean <- colMeans(x)
    d_M <- rep(0,K)

    for (k in 1:K) {
      d_M[k] = integrale(t,(x.mean - m0) * theta[k,])^2/lambda[k]*h[k]
    }
    d_p2[j] <- sum(d_M)

    if (n*d_p2[j]>q.xi_sq)
      cont[l] = cont[l] + 1
  }
}
power <- cont/1000

# case (a)-(b)-(c)-(d)

m0 <- t*(1-t)
m1a <- m0 #case(a)
m1b <- m0 + sqrt(lambda[1])*theta[1,] #case (b)
m1c <- m0 + sqrt(lambda[5])*theta[5,] #case (c)

s <- 0
for (k in 10:K) {
  s <- s + sqrt(lambda[k])*theta[k,]
}
m1d <- m0 + s #case (d)
cont <- rep(0,11)
q.xi_sq <- rep(0,11)
power <- rep(0,10)
for (l in -2:8) {
  p <- 10^l
  chsq <- rep(0,K)
  chsqq <- rep(0,100)
  for (i in 1:K) {
    for (k in 1:K) {
      h[k] <- 1-exp(-lambda[k]*p)
      chsq[k] <- rchisq(1,1)*h[k]
    }
    chsqq[i] <- sum(chsq)
  }
  q.xi_sq[l+3] <- quantile(chsqq,0.95)
  q.xi_sq[l+3]

  d_p2 <- rep(0,iter)
  for (j in 1:iter) {
    u <- matrix(0,n,Time)
    for (i in 1:n) {
      s <- rep(0,500)
      for(k in 1:K) {
        s <- s + sqrt(lambda[k])*(rnorm(1)*theta[k,])
      }
      u[i,] <- s
    }

    x<-matrix(0,n,Time)
    for (i in 1:n){
      x[i,] <- u[i,] + m1a #(m1b for case (b), m1c for case (c), m1d for case (d))
    }
    x.mean <- colMeans(x)

    d_M <- rep(0,K)
    d_Mh <- rep(0,K)
    for (k in 1:K) {
      d_M[k] = integrale(t,(x.mean - m0) * theta[k,])^2/lambda[k]
```

```
      d_Mh[k] <- d_M[k]*h[k]
    }
    d_p2[j] <- sum(d_Mh)

    if(n*d_p2[j]>q.xi_sq[l+3])
      cont[l+3] = cont[l+3] + 1
  }
}
power <- cont/iter
```

Listing A.2: Code to perform functional k-means simulations with the d_p distance.

```
n1 <- 50
n2 <- 50
Time <- 1000
t <- seq(0,1,length=Time)
K <- 100
m1 <- NULL
h <- rep(0,K)

integrale <- function(ascissa,integr){
  return(sum((integr[-1]+integr[-length(integr)])*diff(ascissa)/2))
}

lambda <- rep(0,K)
theta <- matrix(0,K,Time)
for (k in 1:K) {
  lambda[k] <- 1/(k+1)^2
  if (k%%2==0)
    theta[k,] <- sqrt(2)*sin(k*pi*t)
  else if (k%%2!=0 && k!=1)
    theta[k,] <- sqrt(2)*cos((k-1)*pi*t)
  else
    theta[k,] <- rep(1,Time)
}
lambda[1] <- 1/2
lambda[2] <- 1/3
lambda[3] <- 1/4

m0 <- rbind(t*(1-t),4*t^2*(1-t))
s <- 0
for (k in 10:K) {
  s <- s + sqrt(lambda[k])*theta[k,]
}
m1 <- matrix(0,2,Time)
m1[1,] <- m0[1,] + s
m1[2,] <- m0[2,] + 4*s

set.seed(5454)
u11 <- matrix(0,n1,Time)
u12 <- matrix(0,n1,Time)
u21 <- matrix(0,n2,Time)
u22 <- matrix(0,n2,Time)

for (i in 1:n1) {
  s11 <- rep(0,Time)
  s12 <- rep(0,Time)
  z1 <- mvrnorm(K,c(0,0),matrix(c(1,0.4,0.4,1),nrow=2,ncol=2))
  for(k in 1:K) {
    s11 <- s11 + sqrt(lambda[k])*(z1[k,1]*theta[k,])
    s12 <- s12 + sqrt(lambda[k])*(z1[k,2]*theta[k,])
  }
  u11[i,] <- s11
  u12[i,] <- s12
}

for (i in 1:n2) {
  s21 <- rep(0,Time)
  s22 <- rep(0,Time)
  z2 <- mvrnorm(K,c(0,0),matrix(c(1,0.4,0.4,1),nrow=2,ncol=2))
  for(k in 1:K) {
    s21 <- s21 + sqrt(lambda[k])*(z2[k,1]*theta[k,])
```

```
61      s22 <- s22 + sqrt(lambda[k])*(z2[k,2]*theta[k,])
62    }
63    u21[i,] <- s21
64    u22[i,] <- s22
65  }
66
67  x11 <- matrix(0,n1,Time)
68  x12 <- matrix(0,n1,Time)
69  x21 <- matrix(0,n2,Time)
70  x22 <- matrix(0,n2,Time)
71  for (i in 1:n1) {
72    x11[i,] <- u11[i,] + m0[1,]
73    x12[i,] <- u12[i,] + m0[2,]
74  }
75  for (i in 1:n2) {
76    x21[i,] <- u21[i,] + m1[1,]
77    x22[i,] <- u22[i,] + m1[2,]
78  }
79
80  x11.mean <- colMeans(x11)
81  x12.mean <- colMeans(x12)
82  x21.mean <- colMeans(x21)
83  x22.mean <- colMeans(x22)
84
85  par(mfrow=c(1,1))
86  x1 <- rbind(x11,x12)
87  x2 <- rbind(x21,x22)
88
89  pca1 <- prcomp(x1)
90  pca2 <- prcomp(x2)
91  lambda_capp1 <- pca1$sdev
92  theta_capp1 <- pca1$rotation
93  lambda_capp2 <- pca2$sdev
94  theta_capp2 <- pca2$rotation
95
96  #K-MEANS d_p DISTANCE #
97  prop_p <- rep(0,11)
98  h1 <- NULL
99  h2 <- NULL
100 for (b in -2:8) {
101   p <- 10^(b)
102   for (k in 1:K)
103     h1[k]<- lambda_capp1[k]/(1/p+lambda_capp1[k])
104   for (k in 1:K)
105     h2[k]<- lambda_capp2[k]/(1/p+lambda_capp2[k])
106   k <- 2
107   n <- dim(x1)[[1]]
108   p <- sample(n, k)
109   C1 <- x1[p,]
110   C2 <- x2[p,]
111   iter.max <- 9
112   cluster <- numeric(n)
113   d_M1 <- rep(0,K)
114   d_M2 <- rep(0,K)
115   d_p1 <- rep(0,n)
116   d_p2 <- rep(0,n)
117
118   par(mfrow = c(3,3))
119
120   for(i in 1:iter.max)
121   {
122     for (l in 1:n) {
123       for (j in 1:K) {
124         d_M1[j] = (integrale(t, ((x1[l,] - C1[1,]) * theta_capp1[,j])))^2/lambda_
              capp1[j] * h1[j] + (integrale(t, ((x2[l,] - C2[1,]) * theta_capp2[,j])))
              ^2/lambda_capp2[j] * h2[j]
125         d_M2[j] = (integrale(t, ((x1[l,] - C1[2,]) * theta_capp1[,j])))^2/lambda_
              capp1[j] * h1[j] + (integrale(t, ((x2[l,] - C2[2,]) * theta_capp2[,j])))
              ^2/lambda_capp2[j] * h2[j]
126       }
127       d_p1[l] <- sqrt(sum(d_M1))
128       d_p2[l] <- sqrt(sum(d_M2))
129     }
130
131     for(j in 1:n)
132       cluster[j] <- which.min(c(d_p1[j],d_p2[j]))
```

```
133
134       C1 <- NULL
135       C2 <- NULL
136       for(l in 1:k) {
137         C1 <- rbind(C1, colMeans(x1[cluster == l,]))
138         C2 <- rbind(C2, colMeans(x2[cluster == l,]))
139       }
140     }
141     print(cluster)
142   }
```

Bibliography

[1] Benko, M., Hardle, W. and Kneip, A. (2009). Common functional principal components. *Annals of Statistics*, **54**, 1–34.

[2] Boudaoud, S., Rix, H. and Meste, O. (2010). Core Shape modeling of a set of curves. *Computational Statistical Data*, **54**, 308–325.

[3] Revolution Analytics and Weston, S. (2015). doParallel: Foreach Parallel Adaptor for the 'parallel' Package. R package version 1.0.10, *http://CRAN.R-project.org/package=doParallel*.

[4] Revolution Analytics and Weston, S. (2015). foreach: Provides Foreach Looping Construct for R. R package version 1.4.3., *http://CRAN.R-project.org/package=foreach*.

[5] Donoho, D.L. and Johnstone, I.M. (1995). Adapting to Unknown Smoothness via Wavelet Shrinkage. *Journal of the American Statistical Association*, **90**, Issue 432.

[6] Einthoven, W. (1908). Weiteres über das Elektrokardiogram. *Pflüg. Arch.*, **122**, 517-548

[7] Einthoven, W., Fahr, G. and de Waart, A. (1950) On the direction and manifest size of the variations of potential in the human heart and on the influence of the position of the heart on the form of the electrocardiogram. *American Heart Journal*, **40**, 163–211

[8] Ferraty, F. and Vieu, P. (2006). *Nonparametric Functional Data Analysis: Theory and Practice*. Springer Series in Statistics, Berlin.

[9] Fremdt, S., Steinebach, J.G., Horváth, L. and Kokoszka, P. (2013). Testing the Equality of Covariance Operators in Functional Samples. *Scandinavian Journal of Statistics*, **40**, 138–152

[10] Galeano, P., Joseph, E. and Lillo, R. (2015). The Mahalanobis Distance for Functional Data With Applications to Classification. *Technometrics*, **57** (2).

[11] Ghiglietti, A. and Paganoni, A.M. (2014). Statistical inference for functional data based on a generalization of Mahalanobis distance. *Mox Report 39/2014*, Department of Mathematics, Politecnico di Milano.

[12] Ghiglietti, A., Ieva, F. and Paganoni, A.M. (2016). Statistical inference for stochastic processes: two sample hypothesis test. *Journal of Statistical Planning and Inference*, In Press.

[13] Guyton, A.G. and Hall, J.E. (2006). *Medical Physiology*, 11th edition

[14] Horváth, L. and Kokoszka, P. (2012). *Inference for Functional Data with Applications*. Springer, New York.

[15] Ieva F., Paganoni A.M., Pigoli D. and Vitelli V. (2013). Multivariate functional clustering for the morphological analysis of electrocardiograph curves *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **19**, 1937-1944.

[16] Indino F.S. (2015). Analisi statistica di dati ad alta dimensionalità: un'applicazione ai segnali elettrocardiografici.

[17] Liu, X. and Müller, H.-G. (2003). Modes and clustering for time-warped gene expression profile data. *Bioinformatics*, **19**, 1937-1944.

[18] Liu, X. and Yang, M. (2009). Simultaneous curve registration and clustering for functional data. *Computational Statistics and Data Analysis*, **53**, 1361–1376.

[19] Pigoli, D. and Sangalli, L.M. (2012). Wavelets in functional data analysis: estimation of multidimensional curves and their derivatives. *Computational Statistics and Data Analysis*, **96**, 1482-1498

[20] Pigoli, D., Aston J.A.D., Dryden, I.L. and Secchi, P. (2014). Distances and inference for covariance operators. *Biometrika*, **101**, 409–422

[21] R Core Team (2015). *R: A language and environment for statistical computing* . R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[22] Ramsay, J. and Silverman, B. (2002). *Applied Functional Data Analysis*. New York, Springer.

[23] Ramsay, J. and Silverman, B. (2005). *Functional Data Analysis*. New York, Springer.

[24] Sangalli, L.M., Secchi, P., Vantini, S. and Vitelli, V. (2010). k-mean alignment for curve clustering *Computational Statistics and Data Analysis*, **54**, 1219–1233.

[25] Scher, A.M. and Young, A.C. (1957). Ventricular depolarization and the genesis of the QRS. *Annals of the New York Academy of Sciences*, **65**, 768-778.

[26] Struyf, A., Hubert, M. and Rousseeuw, P. (1997). Clustering in an objected-oriented environment. *Journal of Statistical Software*, **1**, no. 4, 1–30.

[27] Tarpey, T. and Kinateder K.K.J. (2003). Clustering Functional data. *Journal of Classification*, **20**, 93-114.

[28] Yao, F., Muller, H.G. and Wang, J.L (2005). Functional data analysis for sparse longitudinal data. *Journal of American Statistical Association*, **100**, 577–590.