UNIVERSIDAD AUTONOMA DE MADRID

Escuela Politécnica Superior

18|19

Degree work

Functional data analysis: registration,interpolation and nearest neighbors in scikit-fda

Pablo Marcos Manchón

Campus Internacional excelencia UAM CSIC

www.uam.es

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Degree as Mathematics and Computer Science

DEGREE WORK

Functional data analysis: registration,interpolation and nearest neighbors in scikit-fda

Author: Pablo Marcos Manchón
Advisor: Alberto Suárez González

junio 2019

Francisco Tomás y Valiente, nº 1
Madrid, 28049
Spain

Pablo Marcos Manchón
Functional data analysis: registration,interpolation and nearest neighbors in scikit-fda

Pablo Marcos Manchón

PRINTED IN SPAIN

Lo peor es cuando has terminado un capítulo
y la máquina de escribir no aplaude.

Orson Welles

Agradecimientos

En primer lugar me gustaría dar las gracias a todas las personas junto a las que he trabajado durante este proyecto.

A Alberto, por darme la oportunidad de formar parte del equipo y guiarme a lo largo de este trabajo, dándome libertad para enfocarme en aquellas partes que más me motivaban. A Carlos, por su dedicación y sus exhaustivas revisiones, sin las cuales este proyecto no sería lo que es a día de hoy y a Jose Luís, por sus inestimables aportaciones matemáticas. Finalmente a Pablo y Amanda, junto a los cuales he tenido el placer de compartir este trabajo a lo largo del año.

Además me gustaría agradecer el apoyo de mi familia y amigos, el cual ha sido fundamental en esta etapa de mi vida. En especial a mis padres, sin los cuales nada de esto habría sido posible, a Sergio, por haber estado desde el inicio de los tiempos, y a Lucía, por compartir conmigo este camino y sin la cual todo habría sido muy diferente.

Resumen

El análisis de datos funcionales, también denominado FDA, es una rama de la estadística encargada del estudio de variables aleatorias de naturaleza funcional, como puede ser un conjunto de series temporales o de curvas en el espacio.

En las últimas dos décadas las técnicas empleadas para este análisis han evolucionado rápidamente, así como sus aplicaciones en gran cantidad de campos como medicina, bioinformática o ingeniería.

Sin embargo, debido a la relativamente reciente aparición de este campo, no hay una gran diversidad de soluciones software que engloben estas técnicas. Por esta razón en el año 2017 surgió el proyecto de código abierto llamado *scikit-fda*, con el objetivo de crear un paquete en Python que diera soporte a este estudio, junto con una comunidad que contribuyera a su desarrollo.

Con este trabajo se pretende contribuir al crecimiento de este proyecto, cuyo objetivo a largo plazo es convertirse en una referencia en FDA que contenga una amplia variedad de herramientas para el estudio de estos datos.

En concreto, durante este trabajo se han realizado contribuciones en diferentes áreas de este campo, entre las que se encuentran el registro de los datos, etapa en la cual se estudia la variabilidad debida a su escala interna, como puede ser el tiempo en el que evoluciona un proceso; o su representación, en la cual se estudian los problemas asociados su tratamiento y almacenamiento.

Palabras clave

Análisis de datos funcionales, interpolación, registro, vecinos próximos, python

Abstract

Functional data analysis, also called FDA, is a branch of statistics that deals with the study of random variables of a functional nature, such as temporal series or curves in space.

In the last two decades, techniques used for this analysis have evolved quickly, as well as their applications in many fields such as medicine, bioinformatics or engineering.

However, due to the relatively recent appearance of this field, there is not a great variety of software solutions that encompass these techniques. For this reason, in 2017 the open-source project called *scikit-fda* arose, with the aim of creating a Python package which would support this study, along with a community that would contribute to its development.

This work aims to contribute to the growth of this project, whose long-term goal is to become a reference in FDA, containing a wide variety of tools for the study of these data.

Specifically, during this work contributions have been made in different areas of this field, including the registration of data, a stage in which the variability due to its internal scale is studied, such as the time in which a process evolves; or its representation, in which the problems associated with its treatment and storage are studied.

Keywords

Functional data analysis, interpolation, registration, nearest neighbors, python

Table of Contents

Lists

List of equations

List of figures

1

Introduction

Functional data analysis (FDA), is a branch of the statistic which deals with the study of random variables of functional nature, such as time series or curves in the space. It is a relatively recent field, whose first references began in the 1950s, with various articles related to the study of stochastic processes. However, it was not until 1982 with the publication by J.O. Ramsay of *When the data are functions* [1] when the term FDA began to be used to denote so this field.

Since then, and especially in the last two decades, techniques used for this analysis have evolved quickly, as well as their applications in a wide range of fields such as medicine, bioinformatics or engineering.

Although there were some software solutions to support this field, such as *fda* [2], powered by J.O. Ramsay [3]; or *fda.usc* [4], powered by the University of Santiago de Compostela. But none of them had been implemented under the philosophy of a open-source project, encouraging open collaboration. In addition all implementations were based on R or Matlab, and there were no Python packages for this purpose, although its use has become widespread in recent years in statistics and machine learning.

In 2017, the *scikit-fda* project came into being, under the name *fda* [5], as part of a degree work made by Miguel Carbajo at the UAM. The aim of this work was to initiate the creation of a Python package to give support to this field, under the philosophy of a open-source project, along with the creation of a community that contributes to development and maintenance. Currently the project is being driven by the machine learning group of the UAM (GAA-UAM) together with the contributions of several degree works, including this one.

Although the project is in an early stage of development, the package is already in use by some researchers and was presented in the III International Workshop on Advances in Functional Data Analysis [6].

1.1. Goals and scope

The aim of this work is to continue the *scikit-fda* project, collaborating with in a general way in its development, but especially in the registration of functional data, one of the areas of FDA, in which it is studied the variability of the data due to its internal scale, such as time in which a process unfolds.

Due to the extent of this area, it is not possible to cover all techniques in this field, therefore the purpose of this work was not delimited from the beginning in a clear way. It was decided to start with techniques exposed in the book *Functional data analysis* [7], which makes a general overview of different aspects in FDA, but without a clear goal on the specific aspects to be dealt with.

For this reason, weekly meetings were scheduled throughout the year with all the team involved in the project, to review the scope of this work and the next steps to follow in its development, with the goal in mind to contribute as much as possible to the growth of this project.

1.2. Document Structure

The main part of the present document is divided into four chapters, aside from this introduction part. In the Chapter 2, it is exposed the state of the art, which gives an overview of the mathematical framework of the functionalities developed to the package.

Chapter 3 summarizes the analysis and design of the functionalities added to the package, as well as the methodology and technologies used during the development of the work. To finish the main part, Chapter 4 presents the conclusions of the work done along with a brief discussion about future work.

Besides, this document appends three annexes. In Annex A there are more detailed descriptions of the different algorithms used, along with proofs of some mathematical results set out in Section 2.3, which have been moved to the annexes for the sake of clarity.

Finally, Annex B includes a series of Python notebooks showing examples of the use of the functionalities incorporated, which have been thought to show in a pleasant way how to use the package to a new user. The documentation of the software developed has been include in the Annex C.

2

State of the art

In FDA, the analyzed data might take the form of temporal series, curves, surfaces or anything else varying over a continuum. The observations that will make up our data will come from random variables, whose realizations will take values in functional spaces, that is to say, they will be realizations of stochastic processes. For this reason, under this framework, we will consider them functions.

One may wonder what makes FDA different from multivariate analysis, considering the fact that data is generally collected and stored in the form of discrete point sets (t_j, y_j) [8]. In multivariate statistics one works with the vector y_j, applying statistical methods and vector calculus to its study. However, FDA keeps the association of values y_j and t_j, taking into account the functional nature of the data, allowing the study of the data using functional calculus.

In this chapter we will make an overview, from a theoretical point of view, of different FDA topics covered in this work.

2.1. Functional data representation

Since our observations will be functions, dependent on one or several continuous parameters, such as the time in which a process takes place, it will be impossible to measure and store all the points which they are defined for.

In practice, the information of a functional datum $x(t)$ is observed and recorded as pairs $x(t_j) = (t_j, y_j)$, where y_j is a snatshoot of the function x at t_j, possibly blurred by measurement error [7].

Although in FDA is also treated the case in which our data are multivariate functions, from $\mathbb{R}^n$ to $\mathbb{R}^m$, in general, as a way of simplification, we will assume that our data are univariate functions.

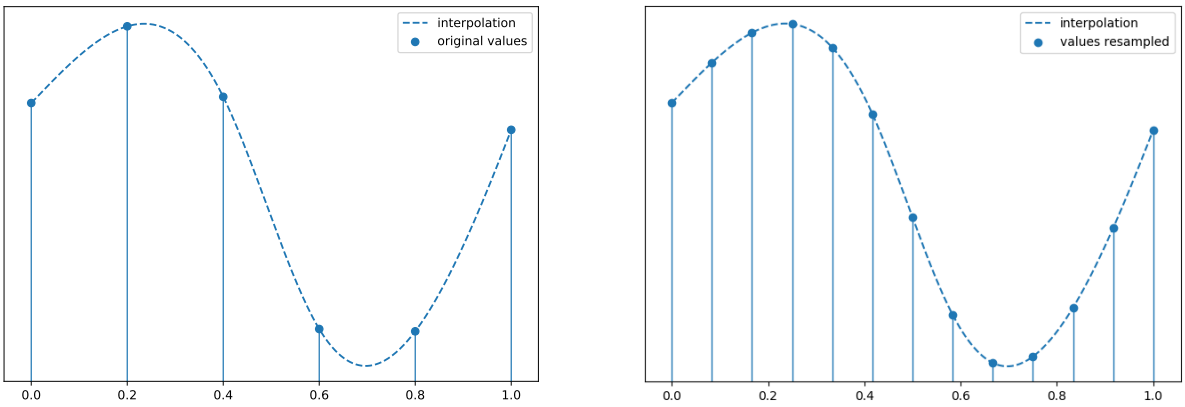
There are two main approaches to represent the data. The first, called discrete representation, stores a finite grid of pairs (t_j, y_j) with the recorded values, but unlike multivariate statistics, the continuous dependence between y_j and t_j will be taken into account.

The second one is a parametric approach, called basis representation, in which we will define a

system of functions $\{\Phi_n(t)\}_{n=1}^N$, such as a truncated Fourier basis or a set of polynomials. This system will form a finite subspace of the original functional space, where we will project the observations, obtaining a representation associated to the coefficients of the basis $x(t) \approx \sum_{n=1}^N c_n \Phi_n(t)$.

2.1.1. Interpolation

In the discrete representation, frequently, it will be necessary to resample or evaluate the data, or its derivatives, at points within the domain range, different that the original pairs $\{(t_j, y_j)\}$ at which our observations have been measured. An example of this it is shown in the figure 2.1. For this purpose, we will use interpolation.



(a) Original observation

(b) Observation resampled

Figure 2.1: Function resampled using interpolation

Although they are not the only methods used for interpolation, splines and smoothing splines are the most used, for this reason we will focus on them during this work.

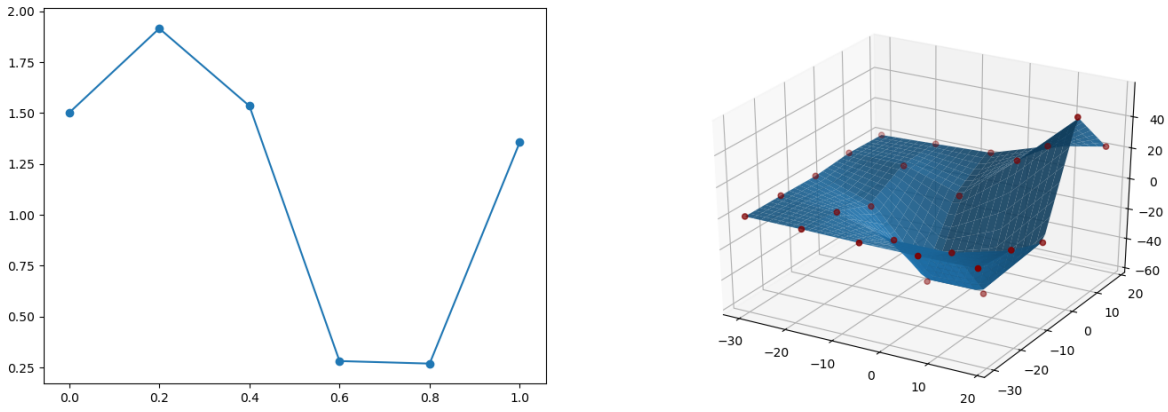
Linear interpolation

Given two points t_k and t_{k+1} for which the values of our function are known, denoted by y_k and y_{k+1}, we can use a line segment to join these values as a first approach. Using this type interpolation, we will obtain a piecewise linear function, whose values in the interval $[t_k, t_{k+1}]$ will be given by

$$x(t) = y_k + \frac{y_{k+1} - y_k}{t_{k+1} - t_k} (t - t_k) . \tag{2.1}$$

In the case of multivariate functions, multilinear interpolation generalizes this method to higher dimensions. The same principles will be applied, but the piecewise domain will be composed by regions calculated using triangulation, and the evaluation will be performed using a multilinear form, obtaining

in the case of surfaces piecewise functions formed by plane sections. In the figure 2.2 it is plotted the result of interpolating a temporal series and a surface.



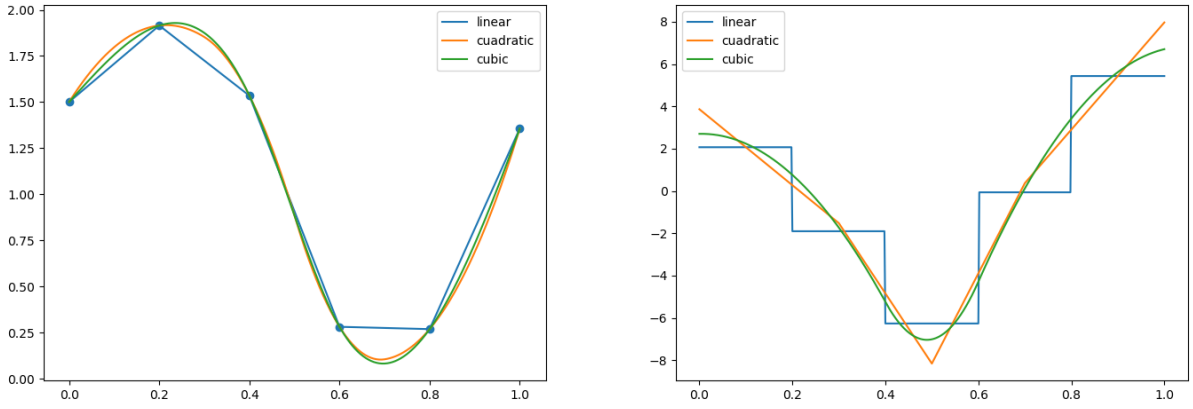
(a) Linear interpolation

(b) Bilinear interpolation

Figure 2.2: Example of linear interpolation

Spline interpolation

In spline interpolation [9], as in the linear case, we will define a piecewise function, but employing polynomials to join the different points known. The main advantage of using splines of order n, i.e., using polynomial of order n in the different intervals, is that we will not only obtain continuous functions, as in the linear case, but we will also obtain functions $\mathcal{C}^{n-1}[a,b]$. This fact will be crucial in FDA, because we will need to use derivatives in many phases of the analysis.



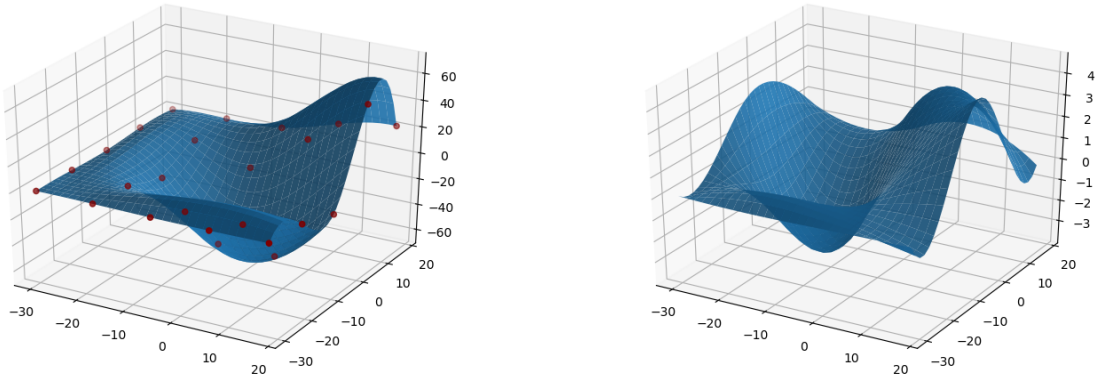
(a) Spline interpolation

(b) First derivatives

Figure 2.3: Example of spline interpolation

To achieve this, we will have to match the values of the derivatives of the adjacent splines in the interpolation knots. If we denote by $p_k(t) = \sum_{j=0}^{n} c_{jk}t^k$ to the spline defined in the region $[t_{k-1}, t_k]$,

during the calculation of the coefficients c_{jk}, we must impose the restriction $p_k^{(d)}(t_k) = p_{k+1}^{(d)}$ for $d = 1, \ldots, n-1$. For this purpose, we will define a linear system of equations which will be solved iteratively. Figure 2.3(a) shows the result of interpolate a temporal series using splines of different orders, and the first derivatives of these splines, which are splines of order $n-1$ due to the derivation of the polynomials.



(a) Surface interpolated

(b) Partial derivative ∂_x

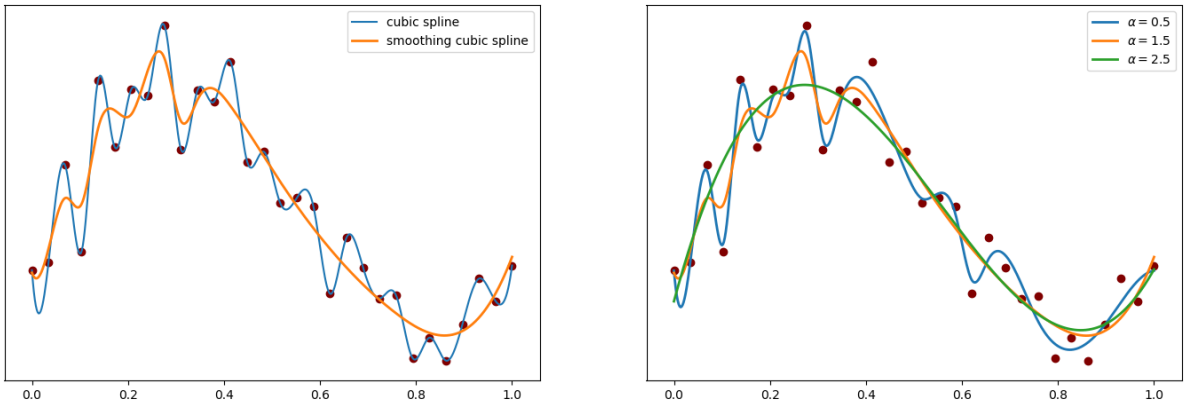
Figure 2.4: Interpolation of surface with bicubic splines

We can extend spline interpolation for multivariate functions, such as surfaces, where bivariate splines will be used. In this case, it is used triangulation to calculate the regions in which the polynomials will be defined, which will be of the form $p(x, y) = \sum_{0 \leq i+j \leq n} c_{i,j} x^i y^j$. In the figure 2.4(a) it is shown the result of the interpolation of a surface using bicubic splines and the partial derivative, respect to x, of the surface.

Smoothing spline interpolation

In several contexts, the functional data will present noise, due to the measurement procedure or to the intrinsic random behaviour of the process, this will motivate the use of smoothing splines.

This method is a variation of the spline interpolation, but we will not use directly the points $\{t_k\}_{k=1}^N$ as knots. Instead, we will use a smaller set of knots $\{\tilde{t}_k\}_{k=1}^M$, generally different from the originals, with their corresponding values $\tilde{y}_k = x(\tilde{t}_k)$ calculated using spline interpolation. The function interpolated using smoothing splines will be defined using spline interpolation on the set of knots $(\tilde{t}_k, \tilde{y}_k)$. Figure 2.5 shows the smoothing interpolation of a noisy observation.



(a) Cubic smoothing

(b) Several values of the smoothing factor

Figure 2.5: Smoothing interpolation

2.2. Registration

Once our data is in functional form, we may encounter an interesting phenomenon that creates problems in FDA. In many situations, functional observations have similar shapes, but they are in some way misaligned. This variability will interfere with further analysis, and therefore, a registration of our data will be necessary in order to eliminate and quantify its variation.

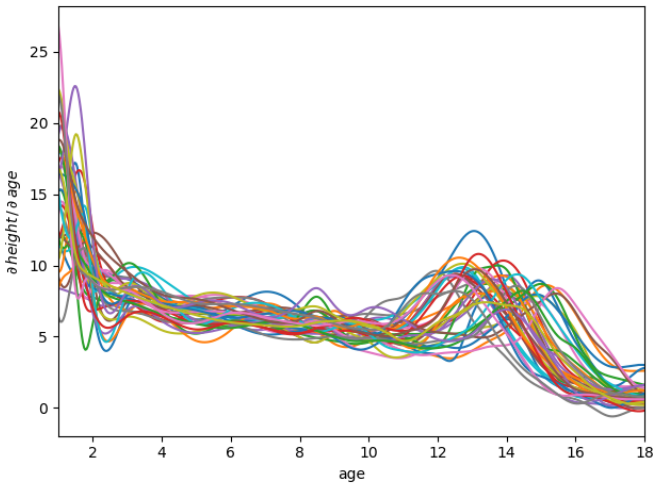
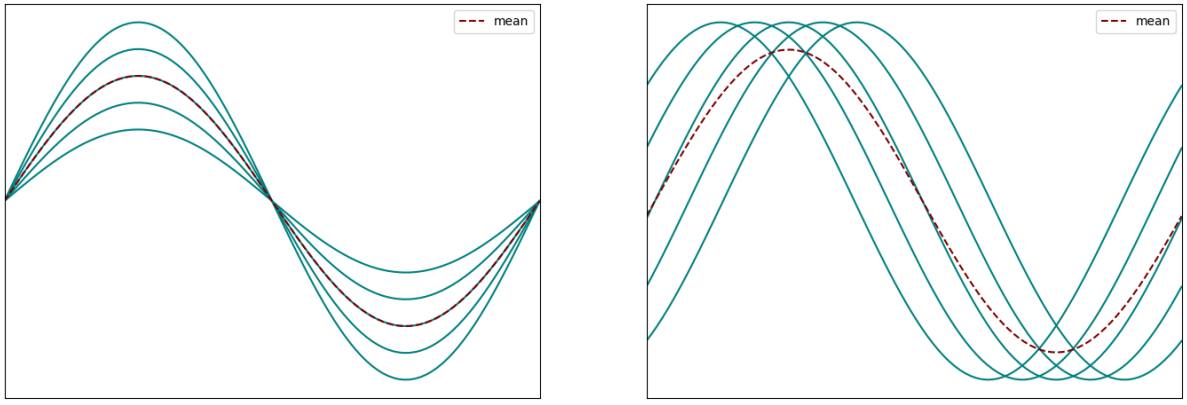


Figure 2.6: Male growth rate

The Berkeley Growth Study [10], performed in 1954, recorded the heights of 54 girls and 39 boys between the ages of 1 and 18 years. In the figure 2.6 it is shown the velocity growth curves of the boys. Every curve shows a main stage of growth corresponding with the puberty; however, these stages are not aligned due to hormonal and other physiological factors in the growth. The rigid metric of physical time may not be directly relevant to the internal dynamics of our problem, so it may be convenient to

make a transformation of the time scale to adapt it to the nature of our data. In the registration of the data, or alignment, this type of variability is analyzed, quantified and separated.

The variability of the data can be decomposed into two sources; the first one comes from the random curve-to-curve variation [11], this is called **amplitude variation**. In the figure 2.21(a) it is shown a set of curves whose variability proceeds exclusively from the amplitude. The second type proceeds from misalignments of the curves with respect to the domain; this is called **phase variation**. In the figure 2.21(b) it is shown a set of curves whose variation is completely due to the phase; this source of variability is the one that will be dealt with in the registration process.



(a) Dataset with amplitude variability

(b) Dataset with phase variability

Figure 2.7: Amplitude and phase variability

2.2.1. Shift registration

A first approach to solve this problem was presented in Ramsay and Silverman (2005) [7], by considering a simple shift in the domain to make the registration. Although it is a basic approach, it will be useful in many cases. Figure 2.8(a) shows a set of sinusoidal waves whose phases are not aligned, and a shift in the time scale will be adequate to make the alignment.

Let $\{x_i\}_{i=1}^n$ be a set of functional observations, which will be aligned using this transformation. We are actually interested in finding the values

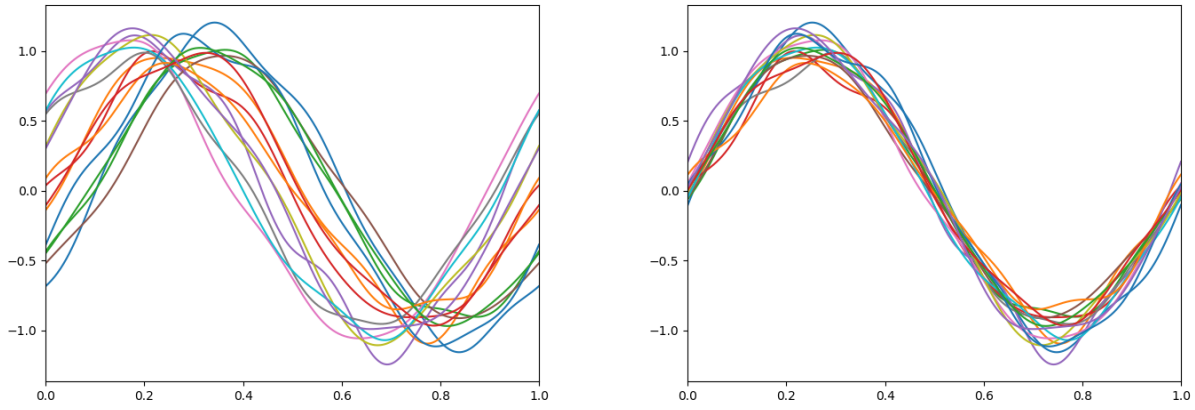
$$x_i^*(t) = x_i(t + \delta_i) \qquad i = 1, 2, \dots n \tag{2.2}$$

where the shift parameter δ_i is chosen in order to align the features of the curves. In the figure 2.8(b) it is shown the result of apply this type of transformation to the set of sinusoidal waves.

A possible solution for the calculation of shifts δ_i, is using a least square criterion, defined as

$$\text{REGSSE} = \sum_{i=1}^{n} \int_{\mathcal{T}} [x_i(t + \delta_i) - \hat{\mu}(t)]^2 \, dt \tag{2.3}$$

The alignment problem will be based on finding the values δ_i that minimize $\mathbb{L}^2$ distance to the cross-sectional mean. For this purpose, we will use the derivatives of the functions x_i, using a variation of the Newton-Rhapson method. A description of the algorithm used to calculate these values is given in the Appendix A.1.



(a) Unregistered curves $x(t)$

(b) Registered curves $x^*(t)$

Figure 2.8: Shift registration of a dataset

2.2.2. Warping functions

In most cases, it is necessary to consider a more general transformation than a simple translation; generally, this transformation will not be linear. In the case of univariate data, we will have functions $x_i : \mathcal{T} \to \mathbb{R}$, in such a way that we can understand the problem as the search for an appropriate parameterization of our data, according with the intrinsic structure of the dataset.

We will consider the functions $\gamma_i : \mathcal{T} \to \mathcal{T}$, referred to as warping functions in te related literature, that we will use to reparametrize the domain, through which we will be able to obtain the curves registered by means of composition of functions, i.e.,

$$x_i^*(t) = x_i(\gamma_i(t)) = x_i \circ \gamma_i. \tag{2.4}$$

So that the alignment does not alter the structure of out functional data, these functions γ_i must be boundary-preserving dipheomorphisms. In the case where the domain of the functions $\mathcal{T}$ is an interval $[a, b]$, the warpings will be strictly increasing functions that fix the bounds of the domain, i.e., $\gamma_i(a) = a$

and $\gamma_i(b) = b$, as could be seen in the figure 2.9.

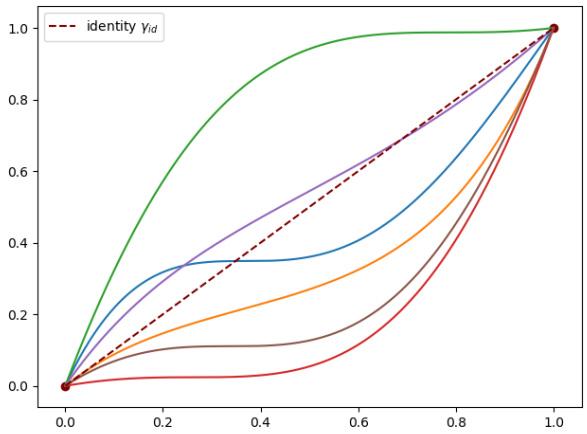


Figure 2.9: Set of warping functions defined in $\mathcal{T} = [0, 1]$

Without loss of generality, in the following sections, we will assume that $\mathcal{T} = [0, 1]$, because the general case $\mathcal{T} = [a, b]$ can be reduced to this with an affine transformation. Also, we will denote the set of these warping functions as Γ.

2.2.3. Landmark registration

A possible solution to register a dataset is to select a set of points for each of the observations and align these points using a warping function. This method is called landmark registration.

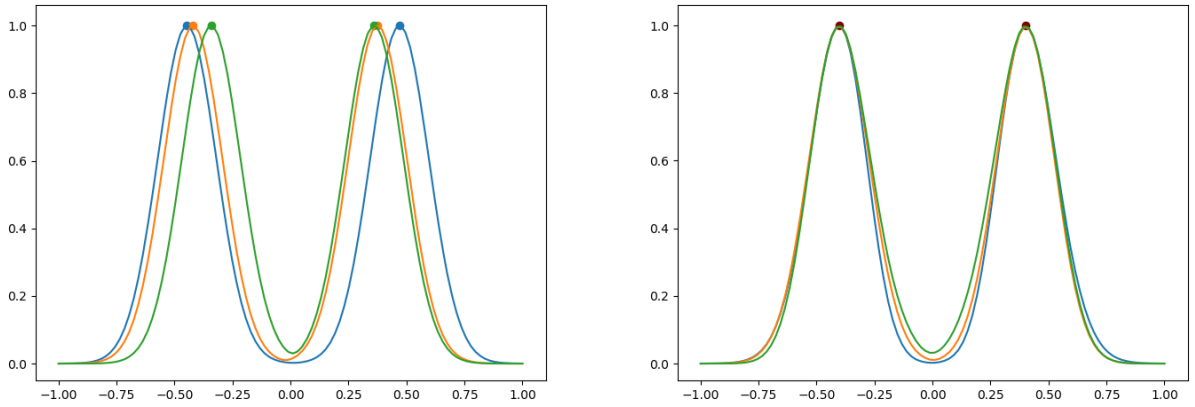
A landmark, or feature of a curve, is some characteristic that can be associated with a specific point of the domain, typically maximums, minimums or zero crossings points. For instance, the population shown in figure 2.10 has two distinctive features, formed by the maximum points of each of the samples.

The landmark registration process will require, for each observation x_i, the identification of the values $\{t_{ij}\}_{j=1}^{F}$ associated with each of the F features, which will be aligned to a common point $\{t_j^*\}_{j=1}^{F}$. Once we have the landmarks points, we will build the warping functions so that $\gamma_i(t_j^*) = t_{ij}$, thus

$$x_i^*(t_j^*) = x_i(\gamma_i(t_j^*)) = x_i(t_{ij}). \tag{2.5}$$

Not all sets of populations have differentiated characteristics of this type, moreover, by taking into account only a limited number of points the resulting alignments can become quite artificial, altering the internal structure of the samples.

This will motivate us to build methods which use a global criterion for alignment and not just a discrete number of points.



(a) Unregistered dataset $x_i(t)$

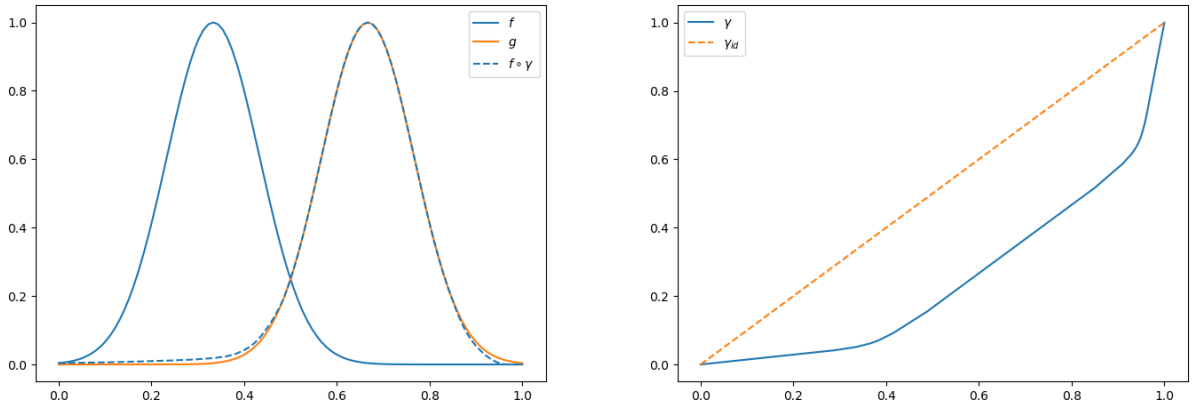
(b) Registered population $x_i^*(t)$

Figure 2.10: Shift registration of a dataset

2.2.4. Pairwise and groupwise alignment

We will define the pairwise alignment problem [8] to deal with the registration problem of two functions using a global criterion.

Let $f_1, f_2 \in \mathcal{F}$ be functional observations of a general space $\mathcal{F}$ and $E : \mathcal{F} \times \mathcal{F} \to \mathbb{R}^+$ an energy functional. The alignment problem may be understood as the search of a warping function γ^* which minimizes the energy between the two functions, i.e., $\gamma^* = \operatorname{argmin}_{\gamma \in \Gamma} E[f_1, f_2 \circ \gamma]$. When a warping γ^* fulfills this property, we will say that f_1 is registered to $f_2 \circ \gamma^*$. In the figure 2.11(a) it is shown an example where two functions have been registered using this approach.



(a) Alignment of f_1 to f_2

(b) Warping built by the DPA algorithm

Figure 2.11: Pairwise alignment

To estimate γ^*, we will explore different paths that the reparameterization may take in a discretized grid, trying to minimize the energy term. In the appendix **??** it is described in detail the algorithm used

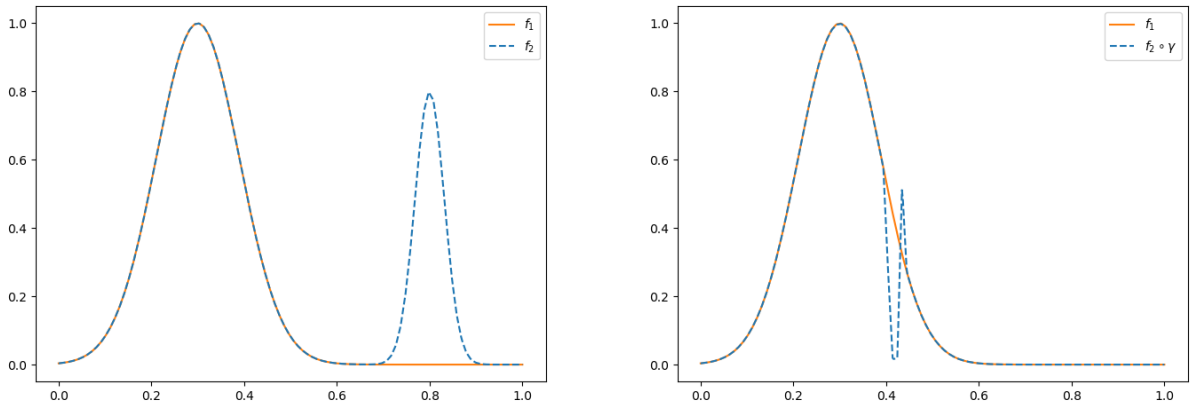
for this task, which is usually referred by several authors as dynamic programing algorithm, or simply DPA [12], because it makes use of this programming technique to search the optimal path.

Given a set of functions $\{f_i\}_{i=1}^n \subset \mathcal{F}$, the groupwise alignment problem will consist in the search of warping functions $\{\gamma_i^*\}_{i=1}^n \subset \Gamma$ to align each of the functions with the rest of them. To achieve this, we will build a target function μ, also called template, to which all the curves will be aligned. For instance, the cross-sectional mean of the functions may be used as target.

A possible choice for the energy term would be to take $E[f,g] = \|f-g\|^2_{\mathbb{L}^2}$, but this criterion is not used in practice because three problems will arise that will not make it adequate: the pinching effect, the lack of symmetry and the inverse inconsistency [13].

Pinching effect

It is a phenomenon we may encounter when we try to align two functions without a perfect match taking the $\mathbb{L}^2$ distance as energy. To minimize the term energy, the optimal solution will tend to squeeze an annoying region, until it disappears. Figure 2.12 shows the result of the registration of two functions with pinching effect. A part of f_2 is identical to f_1 over [0, 0.6] and completely different over the remaining domain. The optimal solution will tend to squeeze the second part of f_2, until it vanishes completely.



(a) f_1 and f_2

(b) Pinching effect

Figure 2.12: Pinching force effect

Lack of symmetry

Let $f_1, f_2 \in \mathcal{F}$ be two functions aligned. Generally, if we apply a reparameterization $\gamma \in \Gamma$ to both functions, their distance will change, thus

$$\|f_1 \circ \gamma - f_2 \circ \gamma\|^2 = \int_{\mathcal{T}} (f_1(\gamma(t)) - f_2(\gamma(t)))^2 dt = \int_{\mathcal{T}} |f_1(s) - f_2(s)|^2 \frac{1}{\dot{\gamma}(\gamma^{-1}(s))} ds \qquad (s = \gamma(t)), \quad (2.6)$$

in other words, the action of Γ over $\mathcal{F}$ under de $\mathbb{L}^2$ metric is not by isometries.

Therefore $E[f_1, f_2] \neq E[f_1 \circ \gamma, f_2 \circ \gamma]$ and consequently $f_1 \circ \gamma$ y $f_2 \circ \gamma$ may not be aligned. A direct consequence of this issue will be the inverse inconsistency of the problem, because of $E[f_1 \circ \gamma, f_2] \neq E[f_1, f_2 \circ \gamma^{-1}]$. It would be desirable that the transformation necessary to align f_1 to f_2 be the inverse transformation to align f_2 to f_1. The search of a more adequate energy for this problem will motivate the section 2.3 of this work.

2.2.5. Amplitude phase decomposition

It will be useful to quantify the amount of variability due to the phase and the amplitude, in order to understand the data and validate the registration procedure. For this purpose, in Kneip and Ramsay (2008) [14] it is developed an effective method for quantifying this variation once the data have been aligned.

Let $\{x_i\}_{i=1}^N$ be a set of functional observations, and $\{y_i\}_{i=1}^N = \{x_i \circ \gamma_i\}_{i=1}^N$ the corresponding registered observations. Also, we will denote as $\bar{x}$ and $\bar{y}$ the cross-sectional means. The Total Mean Square Error is defined as

$$\text{MSE}_{total} = \frac{1}{N}\sum_{i=1}^{N} \|x_i(t) - \overline{x}(t)\|^2 = \frac{1}{N}\sum_{i=1}^{N} \int [x_i(t) - \overline{x}(t)]^2 dt. \tag{2.7}$$

We can decompose the error in the part to each of the two sources of variability.

$$\text{MSE}_{amp} = C_R \frac{1}{N}\sum_{i=1}^{N} \int [y_i(t) - \overline{y}(t)]^2 \, dt \quad \text{MSE}_{phase} = \int \left[C_R \overline{y}^2(t) - \overline{x}^2(t)\right] dt \tag{2.8}$$

Where C_R is a constant related with the covariance between $\dot{\gamma}_i$ and y_i^2. May be proved[*] that $\text{MSE}_{total} = \text{MSE}_{amp} + \text{MSE}_{phase}$.

From this separation we will construct a square multiple correlation index R^2, which will indicate the proportion of the variation due to the phase explained by our registration process. This index will be defined as

$$R^2 = \frac{\text{MSE}_{phase}}{\text{MSE}_{total}}. \tag{2.9}$$

For instance, by quantifying the alignment of the dataset of the figure 2.8, we get a value of $R^2 = 0{,}99$, which indicates that the $99\,\%$ of the variability it is produced by the phase .

2.3. Elastic methods in functional data analysis

As discussed in previous chapter, variability in functional data may come from the intrinsic structure of its domain. This problem does not appear exclusively in functional data analysis, other fields such as shape analysis or computer vision should deal with this kind of variability.

In the classical approach, phase variation is separated in the registration of the data, as a preprocessing step. However, in the elastic analysis approach the separation of the two sources of variability is incorporated as a fundamental part of the analysis. The term *elastic* comes from the fact that we will allow deforming the domain of the functions throughout the analysis.

In Srivastava et. al. (2011) [15] it is presented a novel framework to treat this approach using the Fisher-Rao metric. Although this framework covers a wide variety of topics, as classification, regression or functional component analysis [16], in this chapter we will focus on the part concerning the so-called elastic registration of functional data. For that, we will follow the explanation outlined on chapters 1, 3, 4 and 8 of the book Functional and Shape Data Analysis [8] and the implementation in the R-package *fdasrvf* [17].

2.3.1. Fisher-Rao metric

As introduced in section 2.2.4, the use of the metric $\mathbb{L}^2$ for the pairwise alignment of functions leads to a series of issues, which make its use unsuitable for this problem. This is why we will use differential geometry techniques to find a suitable metric. In particular, given two functions f_1, f_2 and $\gamma \in \Gamma$, we will look for a metric that remains invariant to warpings in the domain, i. e., $d(f_1, f_2) = d(f_1 \circ \gamma, f_2 \circ \gamma)$.

For this purpose we will use the Fisher-Rao metric. This riemannian metric was introduced in 1945 by C. R. Fisher [Ref to fisher paper] in a version on the space of probability distributions. In our case we will use a non-parametric version, slightly different from the original one, defined on the space of signed measures. This metric plays a very important role in information geometry, and it is the only metric that possesses this property of invariance [18].

A riemannian metric is an application that assigns an inner product at each point of variety on its tangent space that varies smoothly from point to point. This will give us local notions of angles, length of curves or volumes.

In our case our manifold will be formed by the set of absolutely continuous functions $\mathcal{F}$, that is, those functions whose derivative exists a.e. and belongs to $\mathbb{L}^2$. We will endow $\mathcal{F}$ with the Fisher-Rao metric.

Let $f \in \mathcal{F}$ and $v_1, v_2 \in T_f(\mathcal{F})$, the Fisher-Rao metric is defined as

$$\langle\langle v_1, v_2 \rangle\rangle_f = \frac{1}{4} \int_0^1 \dot{v}_1(t) \dot{v}_2(t) \frac{1}{|\dot{f}(t)|} dt, \tag{2.10}$$

where $\dot{v}$ denotes the derivative.

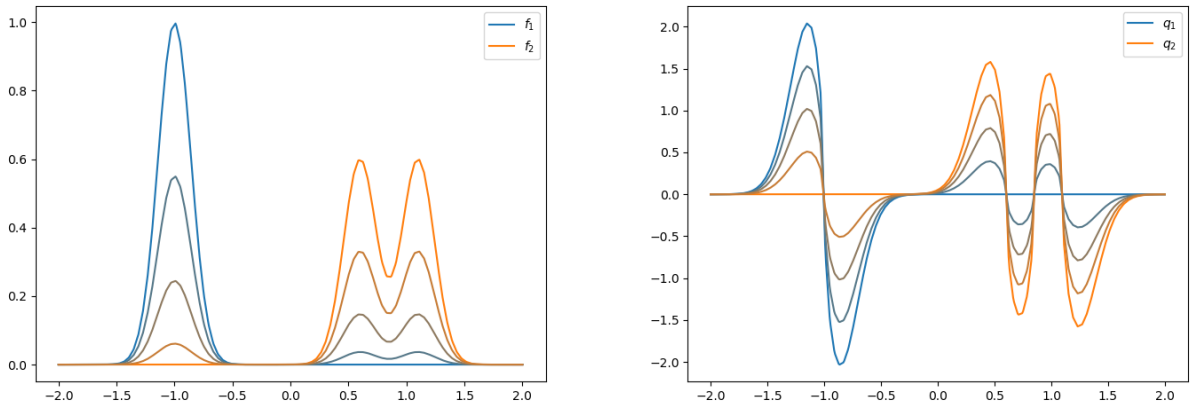
Using this local notion of inner product, we will be able to calculate the length of a differentiable path in our manifold $\alpha(\tau) \subset \mathcal{F}$ integrating through it, i.e.,

$$L[\alpha] = \int_0^1 \langle\langle \dot{\alpha}(\tau), \dot{\alpha}(\tau) \rangle\rangle_{\alpha(\tau)} d\tau. \tag{2.11}$$

This will allow us to define the distance between two points of the manifold $f_1, f_2 \in \mathcal{F}$ as the length of the geodesic path, or shortest path, which connects f_1 and f_2,

$$d_{FR}(f_1, f_2) = \inf_{\alpha:[0,1]\to\mathcal{F}, \alpha(0)=f_1, \alpha(1)=f_2} L[\alpha]\,. \tag{2.12}$$

Finding the geodesic path directly is computationally unapproachable, for this reason we will introduce the **square root slope function** (SRSF) transform, which will greatly simplify this computation.



(a) Some steps of the geodesic path between f_1 and f_2

(b) SRSF of the path

Figure 2.13: Geodesic path in $\mathcal{F}$ and the corresponding SRSF's

Given $f \in \mathcal{F}$, its SRSF is defined as $SRSF\{f\} = \operatorname{sgn}(\dot{f})\sqrt{\dot{f}}$. To simplify notation, in the subsequent sections, we will denote the SRSF of a function $f_i \in \mathcal{F}$ as q_i.

Under this transformation, the Fisher-Rao metric becomes the usual metric in $\mathbb{L}^2$, so that the distance between two functions will be calculated using the distance $\mathbb{L}^2$ of their corresponding SRSF's, i.e., $d_{FR}(f_1, f_2) = \|q_1 - q_2\|_{\mathbb{L}^2}$. A proof of this result is included in appendix A.2.

Taking advantage of this characterization we will take our functions to the SRSF's space to perform our analysis efficiently, and then transform the result to the original space, because the SRSF establishes a map up to constant between $\mathcal{F}$ and the space of SRSF's with the $\mathbb{L}^2$ metric. A consequence is that the computation of geodesics shown in 2.13 becomes in a straight line between SRSFs,

$$\alpha(\tau) = (1-\tau)q_1 + \tau q_2 \quad 0 \leq \tau \leq 1. \tag{2.13}$$

Given $\gamma \in \Gamma$, when it is calculated the SRSF of $f \circ \gamma$, due to the chain rule, we can observe that $SRSF\{f \circ \gamma\} = \text{sgn}(\dot{f} \circ \gamma)\sqrt{\dot{f} \circ \gamma}\sqrt{\dot{\gamma}} = (q \circ \gamma)\sqrt{\dot{\gamma}}$. To simplify the notation we will denote the SRSF of this composition as (q, γ). Using this fact, we can proof that the action of Γ on $\mathscr{F}$ is an action by isometries, i.e, given $f \in \mathscr{F}$ and $\gamma \in \Gamma$ the composition $f \circ \gamma$ preserves the metric. A proof of this result is given in the appendix A.2. Figure 2.14 shows a diagram with the action of the composition in the different spaces.

$$\begin{array}{ccc} f & \xrightarrow{\text{SRSF}} & q \\ \text{action on } \mathscr{F} \downarrow & & \downarrow \text{action on } \mathbb{L}^2 \\ f \circ \gamma & \xrightarrow{\text{SRSF}} & (q, \gamma) \end{array}$$

Figure 2.14: Action of Γ

2.3.2. Amplitude and phase spaces

We will formalize the amplitude and phase distance between functions of $\mathcal{F}$, which will come in a natural way due to the properties of the metric.

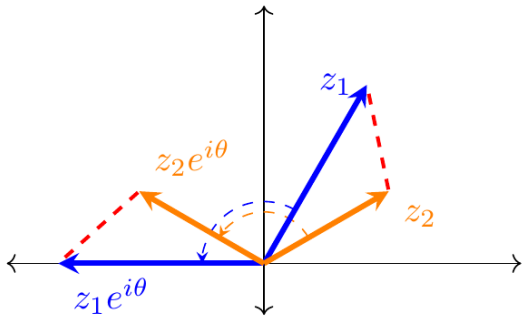


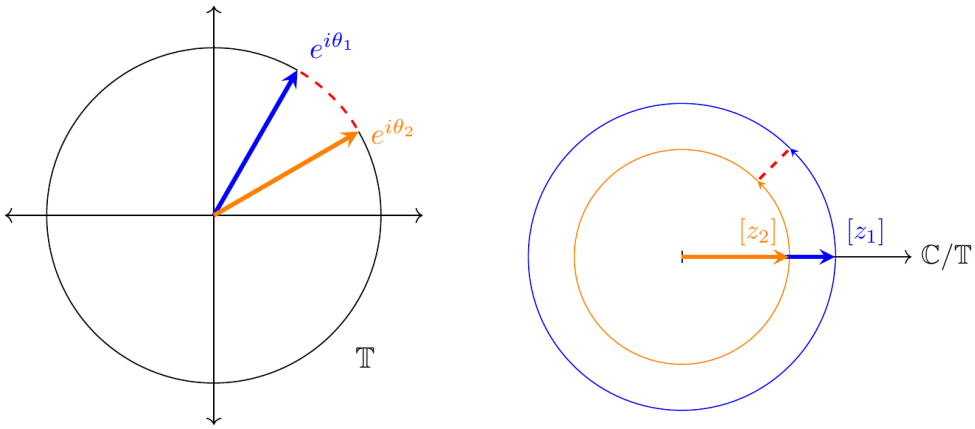
Figure 2.15: Rotation in complex plane

Firstly, before we proceed to define the phase and amplitude space, we will see an analogy with the rotations on the complex plane, which will be very useful to understand the role played by each of the magnitudes in our manifold.

Let z_1, z_2 be two points in $\mathbb{C}$, as it is illustrated in the figure 2.15, when it is applied a rotation on the plane, their distance remains invariant, i.e., it is an action by isometries, as the reparameterizations on our manifold.

The variability between two vectors in the plane can be completely specified by the angle between

them, which will play the role of the phase in our space, and the difference of their modules, which will be equivalent to the distance between the vector once aligned.



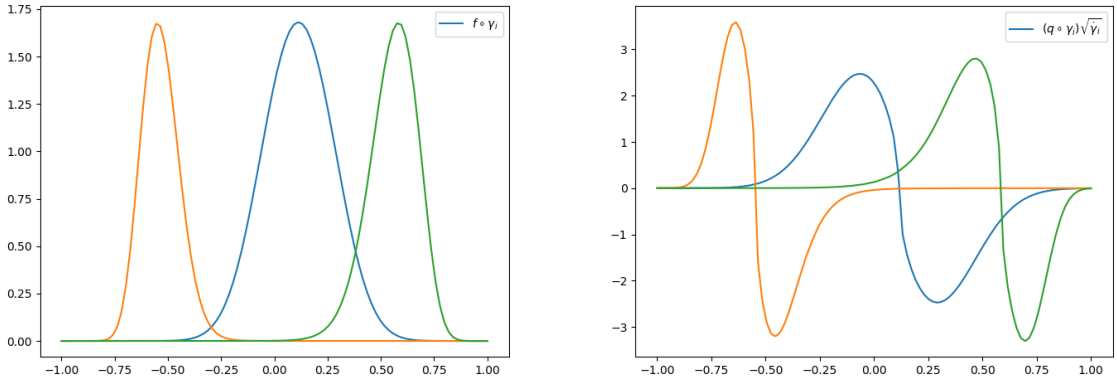
(a) Angle between vectors

(b) Distance in $\mathbb{C}/\mathbb{T}$

Figure 2.16: Phase and amplitude in the complex plane

This distance of the vectors aligned may be understood as a distance in the quotient space $\mathbb{C}/\mathbb{T}$, where $\mathbb{T}$ is the unit circle, which it is isomorph to the group of rotations in the plane $SO(2)$. In this quotient space we will define the equivalence classes $[z] = \{ze^{i\theta} : \theta \in [0, 2\pi]\}$.

In the case of our manifold, let $q \in \mathbb{L}^2$ be a SRSF. We will define its orbit under Γ as $[q] = \{(q, \gamma) : \gamma \in \Gamma\}$, in other words, it is the set of reparameterizations associated to q. In the figure 2.17(a) it is shown some reparameterizations associated to a function of $\mathscr{F}$ and their corresponding SRSF's in 2.17(b).



(a) $f \circ \gamma_i$

(b) (q, γ_i)

Figure 2.17: Functions in the same orbit

We will denote amplitude space $\mathscr{A} = \mathbb{L}^2/\Gamma$ to the set of these orbit. In this space the phase variation is incorporated within equivalence classes, while the amplitude variation appears across equivalence

classes, as in the analogy with the complex plane. We will endow the space with the elastic metric, defined as

$$d_a\left([q_1],[q_2]\right) = \inf_{\gamma_1,\gamma_2\in\Gamma}\left(\|(q_1,\gamma_1)-(q_2,\gamma_2)\|\right), \tag{2.14}$$

To quantify the other source of variability, we will define the phase space, which will denoted as $\Gamma = \{\gamma : [0,1] \to [0,1] : \gamma \text{ is a boundary-preserving diffeomorphism}\}$, for which the natural distance will be given by the Fisher-Rao metric.

Let $\gamma \in \Gamma$ be a warping function, under the Fisher-Rao metric, their norm will be 1, thus

$$\|\gamma\|_\Gamma^2 = \|SRSF(\gamma)\|_{\mathbb{L}^2}^2 = \|\dot{\gamma}\|_{\mathbb{L}^2}^2 = \int_0^1 \sqrt{\dot{\gamma}(t)}\sqrt{\dot{\gamma}(t)}dt = \int_0^1 \dot{\gamma}(t)dt = \gamma(1)-\gamma(0) = 1. \tag{2.15}$$

The SRSF transform will be an isometry between Γ and the unit sphere in $\mathbb{L}^2$, also known as Hilbert Sphere $\mathbb{S}_\infty = \{\psi \in \mathbb{L}^2 : \|\psi\|_{\mathbb{L}^2} = 1\}$. To calculate distances in Γ we will apply this transformation which will allow us to use the structure of $\mathbb{S}_\infty$.

Let γ_1, γ_2 be in Γ and $\psi_1 = \sqrt{\dot{\gamma}_1}, \psi_2 = \sqrt{\dot{\gamma}_1}$ be their SRSF, their distance in Γ will be given by

$$d_{phase}(\gamma_1,\gamma_2) = d_\psi(\psi_1,\psi_2) = \cos^{-1}\left(\int_0^1 \psi_1(t)\psi_2(t)dt\right). \tag{2.16}$$

This result is analogous to the rotations in the plane, where the cosine of the angle between two unit vectors will be given by the inner product. If we apply a rotation to two vectors, their angle remains invariant, in our case the phase distance will be invariant to common reparameterizations due to the properties of the Fisher-Rao metric.

Let f_1, f_2 be functions in $\mathcal{F}$, their relative phase is defined as

$$(\gamma_1^*,\gamma_2^*) = \operatorname*{argmin}_{\gamma_1,\gamma_2\in\Gamma}\|(q_1,\gamma_1)-(q_2,\gamma_2)\| \in \Gamma\times\Gamma. \tag{2.17}$$

Which will allow us to calculate the phase variability between them, as a generalization of the notion of an angle. The phase distance between these functions will be the distance of their relative phase $d_{phase}(\gamma_1^*,\gamma_2^*)$. Alternatively, we can use de properties of the metric to define the relative phase depending on a single warping, $\gamma_{12}^* = \operatorname*{argmin}_{\gamma\in\Gamma}\left(\|(q_1,\gamma)-q_2\|\right)$, which is quivalent to set γ_2^* to de identity in 2.17.

2.3.3. Pairwise alignment

The Fisher-Rao metric will allow us to formulate a suitable criterion for the pairwise alignment problem, as presented in section 2.2.4, that avoids all the problems associated with the metric $\mathbb{L}^2$.

Given $f_1, f_2 \in \mathscr{F}$, to register f_1 to f_2 the Fisher-Rao distance will be minimized as energy term, i.e., the warping used in the alignment will be

$$\gamma^* = \operatorname{argmin}_{\gamma\in\Gamma} d_{FR}(f_1 \circ \gamma, f_2) = \operatorname{argmin}_{\gamma\in\Gamma} \|(q_1, \gamma) - q_2\|_{\mathbb{L}^2}. \tag{2.18}$$

The energy used to align the functions in figure 2.11(a) was actually this. We will have a property of symmetry, since $E[f_1, f_2] = E[f_1 \circ \gamma, f_2 \circ \gamma]$, the warping used to register $f_1 \circ \gamma$ to $f_2 \circ \gamma$ will be the same as in the previous case.

Because of this property, our problem will have inverse consistency, since

$$E[f_1 \circ \gamma, f_2] = E[f_1 \circ \gamma \circ \gamma^{-1}, f_2 \circ \gamma^{-1}] = E[f_1, f_2 \circ \gamma^{-1}], \tag{2.19}$$

so that the warping needed to register f_2 to f_1 is ${\gamma^*}^{-1}$.

In addition, the so-called pinching effect will not appear, due to the term $\sqrt{\dot{\gamma}}$ that is present in the criterion to minimize.

2.3.4. Karcher means

In descriptive statistics, given a set of random points $\{x_i\}_{i=1}^N \subset \mathbb{R}^n$, the sample mean $\bar{x} = \frac{1}{N}\sum_{i=1}^N x_i$ is used to estimate the central tendency of the data. This mean minimizes the sum of square distances.

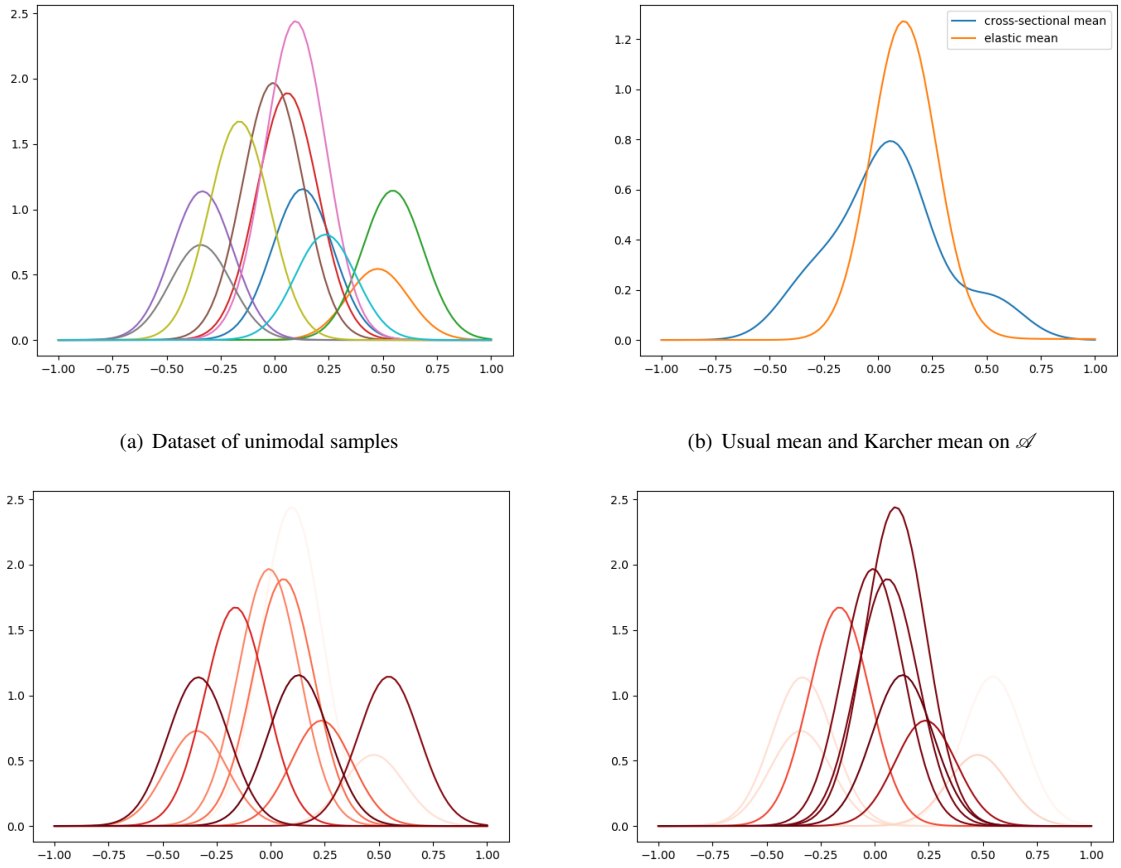
In the functional case, the mean function will have this property thus if $\{f_i\}_{i=1}^N \subset \mathbb{L}^2$ is a set of functional observations, their mean $\bar{f}(t) = \frac{1}{N}\sum_{i=1}^N f_i(t)$ will minimize $\sum_{i=1}^N \|f_i - \bar{f}\|_{\mathbb{L}^2}$.

We are interested in extend this idea to general metrics spaces. Let (X, d) be a metric space and $\{x_i\}_{i=1}^N$ random points in X. The Fréchet variance of a point $p \in X$ as $\Psi(p) = \sum_{i=1}^N d^2(p, x_i)$.

The Fréchet mean of these random points will be defined as the element $m \in X$ which globally minimizes the Fréchet Variance. If this global minimum does not exist, we will call Karcher means to the points that locally minimizes $\Psi(p)$, i. e.,

$$m = \arg\min_{p\in M} \sum_{i=1}^N d^2(p, x_i) \tag{2.20}$$

These Karcher means will be able to better capture the geometry of the problem than usual mean, as it is shown in the figure 2.18(b), where it is used a Karcher mean on $\mathscr{A}$. To calculate this mean it is used the DPA algorithm explained in the appendix **??**. For instance, the dataset shown in 2.18(a) contains gaussian-like samples, however the usual mean of figure 2.18(b) is not able to reflect this shape, unlike the Karcher mean in $\mathscr{A}$.



(a) Dataset of unimodal samples

(b) Usual mean and Karcher mean on $\mathscr{A}$

(c) Ranked by elastic distance

(d) Ranked by phase distance

Figure 2.18: Karcher mean of dataset

As an example of the behavior of these means with different metrics, we may measure the centrality of an observation in a dataset using its distance to the Karcher mean. In the figure 2.18 it is used this idea to rank a dataset of unimodal samples. Reddish colors indicate higher centrality of a sample, i.e., a smaller distance to the mean, and lighter colors indicates outlier samples. In the figure 2.18(c) it is used the amplitude distance, and their corresponding Karcher mean. We can observe that the location of the mode in a sample does not affect its centrality, in contrast to the phase distance, as it is shown in the figure 2.18(d).

2.3.5. Elastic registration

In this section we will define a procedure, which will be called elastic registration, to perform a groupwise alignment of a dataset under this framework. A target function will be created, called elastic mean, to which all samples will be aligned later, as discussed in section 2.2.4.

Let $\{f_i\} \subset \mathscr{F}$ be a dataset of functions to register and q_i their corresponding SRSFs. Firstly we will compute their Karcher mean in $\mathscr{A}$, defined as

$$[\mu_q] = \underset{[q]\in\mathscr{A}}{\operatorname{arginf}} \sum_{i=1}^{n} d_a\left([q],[q_i]\right)^2. \tag{2.21}$$

The bracket notation $[\mu_q]$ is used to emphasize the fact it is an orbit in a quotient space. We will need a criterion to select a particular element of this orbit, for that reason we will define the center of the orbit as the element $\tilde{\mu}_q \in [\mu_q]$ such that the relatives phases γ_i^* between q_i and $[\mu_q]$ have as Karcher mean the identity.

We will select an arbitrary element μ_q of $[\mu_q]$ to which we will calculate the corresponding warping functions to align the set of SRSFs, i.e., $\gamma_i = \operatorname{arginf}_{\gamma\in\Gamma} \|\tilde{q} - (q_i,\gamma)\|$.

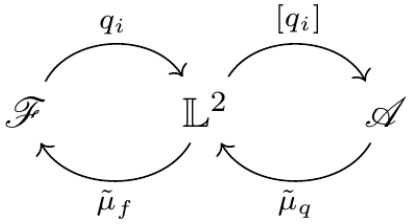


Figure 2.19: Scheme of the elastic registration procedure.

Then, we will compute the Karcher mean of $\{\gamma_i\}$, denoted as $\bar{\gamma}$, in the phase space, using the algorithm detailed in the appendix B.3. The center of the orbit will be $\tilde{\mu}_q = (\mu_q, \bar{\gamma}^{-1})$.

The mean of the relative phases with respect to $\tilde{\mu}_q$, will be the identity, thus

$$\operatorname{arginf}_{\gamma\in\Gamma} d_{FR}(\gamma_i \circ \bar{\gamma}^{-1}, \gamma) = \operatorname{arginf}_{\gamma\in\Gamma} d_{FR}(\gamma_i \circ \bar{\gamma}^{-1} \circ \bar{\gamma}, \gamma \circ \bar{\gamma}) = \operatorname{arginf}_{\gamma\in\Gamma} d_{FR}(\gamma_i, \gamma \circ \bar{\gamma}) = \gamma_{id}. \tag{2.22}$$

Finally, we will construct the template to which the samples will be aligned, called elastic mean, $\tilde{\mu}_f = \frac{1}{N}\sum_{i=1}^{N} f_i(0) + \int_0^t \tilde{\mu}_q(s)|\tilde{\mu}_q(s)|ds$, which is the pullback of $\tilde{\mu}_q$ to the original space $\mathscr{F}$. In the figure 2.20 may be observed the result of the registration of the berkeley growth curves of the figure 2.6.

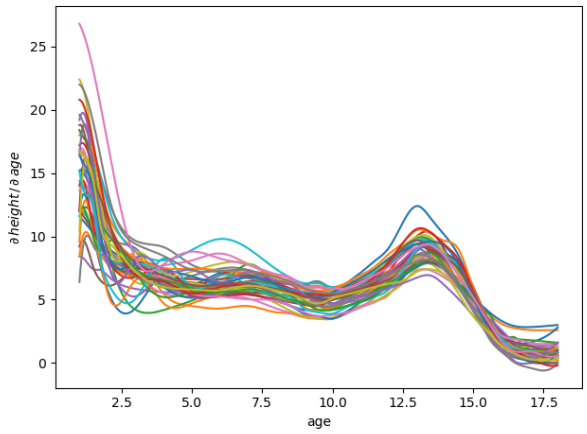
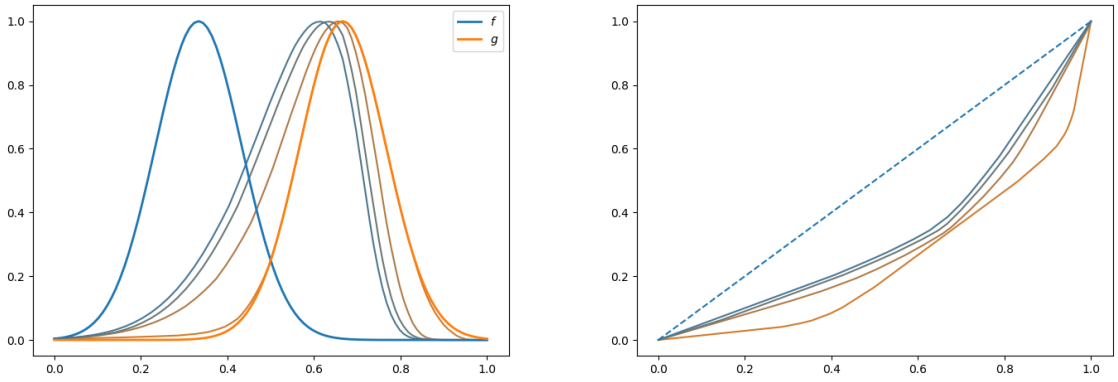


Figure 2.20: Elastic registration of the Berkeley velocity curves

An important property of the elastic mean $\tilde{\mu}_f$ is that given a dataset $\{c_i f(\gamma_i(t)) + e_i\}_{i=1}^n$, where $\{\gamma_i\}_{i=1}^n$ have as Karcher mean in Γ the identity and c_i and e_i are positive constants with mean 1 and 0 respectively, then $\tilde{\mu}_f$ is a consistent estimator of f [refff to paper].

2.3.6. Restricting elasticity



(a) Registration of f_1 to f_2 with a penalty term

(b) Warping used in the alignment

Figure 2.21: Penalized elastic registration with different values of λ

Sometimes it is necessary to control the amount of elasticity during the registration process, for this purpose it is possible to add a penalty term $\mathcal{R}(\lambda)$ to the elastic distance 2.14. Let $q_1, q_2 \in \mathbb{L}^2$ be two SRSFs and $\lambda > 0$, we will define the penalized elastic distance as

$$d_\lambda(q_1, q_2) \equiv \inf_{\gamma \in \Gamma} \left(\|q_1 - \left(q_2 \circ \gamma \sqrt{\dot{\gamma}}\right)\|^2 + \lambda \|\sqrt{\dot{\gamma}} - 1\|^2 \right)^{(1/2)}. \tag{2.23}$$

2.4. Nearest neighbors

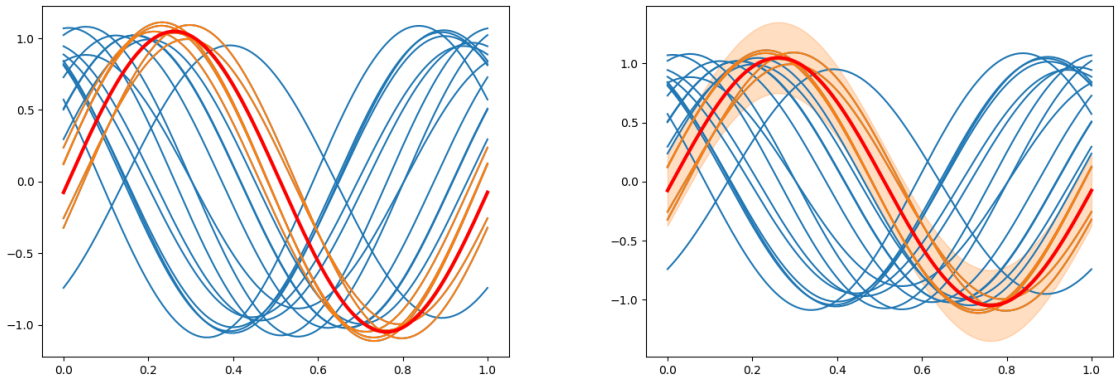
The nearest neighbors estimators (NN-estimators) are a family of methods widely used in statistics and machine learning, in problems of classification or regression, among others. These estimators are based on the idea of neighborhood, using the notion of distance, so that it is made a local estimation of the density of a dataset.

Although in their classic version they are used with sets of vectors, their ideas will work in the same way in general metric spaces [19], as the functional ones we will work with.

2.4.1. Nearest neighbor search

Let $(\mathcal{F}, d)$ be a metric space and $(\mathcal{X}_i, Y_i)_{i \leq i \leq n}$ a training set with their respective labels or responses. To estimate a datum x, either for classification or prediction of its response, firstly, it will be necessary to find the elements of the training set closest to this datum, which will form its neighborhood, denoted as $k(x)$.

There are two variants of these methods. In the first one, consisting of k-nearest neighbors estimators, it is taken as neighborhood the k closest elements to x, i. e., if the training pairs are re-indexed as $(\mathcal{X}_{(i)}, Y_{(i)})\, 1 \leq i \leq n$ so that the $\mathcal{X}_{(i)}$'s are re-arranged in increasing distance from x, $d(x, \mathcal{X}_{(1)}) \leq d(x, \mathcal{X}_{(2)}) \leq \cdots \leq d(x, \mathcal{X}_{(n)})$, then $k(x) = \{\mathcal{X}_{(i)} : 1 \leq i \leq i \leq k\}$.



(a) K-nearest neighbors

(b) Radius-nearest neighbors

Figure 2.22: neighborhoods using distance $\mathbb{L}^\infty$

In the second variant, less used in practice, consisting of the radius neighbors estimators, the neighborhood contains the samples $\mathcal{X}_i$ in the ball of radius r centered in x, i.e. $k(x) = \{\mathcal{X}_i : d(\mathcal{X}_i, x) \leq r\}$. For instance, if we use the distance $\mathbb{L}^\infty$, we may visualize $k(x)$ as the set of all functions within a band of radius r around x.

In the figure 2.22 there are shown the neighborhoods with these two different approaches.

In practice, for the construction of the neighborhoods, the simplest solution is to perform a linear search, calculating the distances between x and all the elements of the training set.

The naïve approach of the linear search may be improve using data structures based on spatial indexes, such as ball trees [20]. However, the best nearest-neighbor data structure for a given application will depend on the dimensionality, size, and underlying structure of the data.

2.4.2. Classification

In the classification problem, the samples of the training set $\mathcal{X}_i$ are associated to the labels Y_i corresponding to their class. Given a datum x, we will predict a new label using the majority class among its neighbors, so that the predicted class will be

$$\hat{Y} = \underset{j}{\operatorname{argmax}} \sum_{\mathcal{X}_i \in k(x)} \mathbb{1}_{\{Y_i = j\}}. \tag{2.24}$$

It is possible to make a weighting vote, so that the closest neighbors will have a greater weight, for example, using $w_i = 1/d(\mathcal{X}_i, x)$, so that the resulting label will be $\hat{Y} = \underset{j}{\operatorname{argmax}} \sum_{\mathcal{X}_i \in k(x)} w_i \mathbb{1}_{\{Y_i = j\}}$.

2.4.3. Regression

In the regression problem, each of the training samples $\mathcal{X}_i$ have a response Y_i associated with them. This response can be either scalar or functional, although the way to proceed will be similar.

In both cases, it will be necessary to select the responses associated with the elements of the neighborhood $k(x)$, which will be used to predict the response of the datum x.

In the scalar response case, a weighted average of the neighbors' responses Y_i will be used, so that the prediction will be calculated as

$$\hat{Y} = \sum_{(X_i, Y_i): X_i \in k(x)} w_i Y_i \,, \tag{2.25}$$

where $\sum w_i = 1$. which may be chosen based on distance or uniformly.

In the case where the responses $\mathcal{Y}_i$ are also functional data, the predicted response will be constructed in a similar way as in the previous case, using a weighted average of functions, or a centroid, such as the Karcher means presented in section 2.3.5.

3

Design and development

So far, we have focused on the mathematical framework of the functionalities incorporated into the package, however, most of the work was done during the design, development and integration parts.

Although the package is currently at an early stage of development, it is already beginning to be used by researchers in different countries, for this reason, and to allow the expansion of the project, it has not only been necessary the coordination of the team involved, but also special emphasis has been placed on carrying out the development in a public and transparent way and making a responsible integration of the new functionalities.

Figure 3.1: scikit-fda logo

3.1. Analysis

As mentioned above, the scikit-fda project started in 2017, at which time an analysis of the requirements of the package was made [5]. These requirements are still in effect today, some of them are:

- The software developed has to be a Python package.
- It has to be an open-source project.
- The software must follow Python standards defined in PEP 8 and PEP 257.
- Documentation has to be intended for a very general audience.
- The project has to include and extensive test bench of unit test and continuous integration mechanism.

In addition to the original ones, three new requirements have been educed:

- The software should be cross-platform and the mechanism of continuous integration should run the test bench in the main operating systems, that is, Linux, MacOs and Windows, as well as in the different Python versions supported.
- API should be similar, as far as possible, to the numpy [21], scipy [22] and scikit-learn [23] ones, allowing whenever possible the use of their functionalities with the objects of the software developed.
- The documentation should contain examples showing different functionalities.

3.2. Design

Originally the package was structured around two classes, FDataGrid and FDataBasis, designed for each of the main data representations of the data and consisted of four modules with basic statistics, operations and methods to perform kernel smoothing.

Due to the expansion of the project, the package has been completely restructured, with a more hierarchical structure. Figure 3.2 shows a diagram with a high-level description of this structure. The following subsections summarize, in general terms, the functionalities incorporated and design changes made during this work. A more detailed description may be found in Annex C, or in the documentation available online.

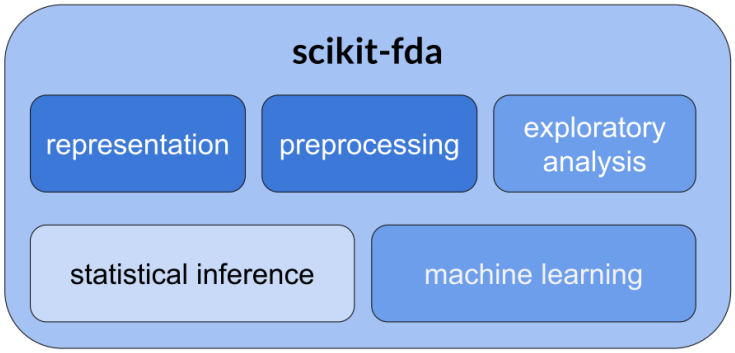


Figure 3.2: Map of scikit-fda [24]

3.2.1. Representation module

One of the most important changes carried out has been the creation of the FData class, from which the existing FDataGrid and FDataBasis inherit, designed to unify the API of the different representations. This class implements general methods for evaluation, extrapolation, and plotting and defines abstract ones common to all representations.

In addition, two submodules have been added, *interpolation* and *extrapolation*, containing classes

used during the evaluation of functional data, implementing the concepts explained in section 2.1.

3.2.2. Preprocessing module

In the preprocessing module a specific sub-module was created to deal with the registration problem, with methods to deal with the data with the techniques studied throughout sections 2.2 and 2.3, allowing data to be registered by means of shifts, landmarks or the elastic approach.

3.2.3. Machine learning module

The machine earning module is divided into three submodules: classification, regression and clustering. Classes have been added for the classification and regression of functional data with estimators based on nearest neighbors explained in section 2.4.

3.2.4. Miscellaneous module

This module contains different scattered topics used during the analysis of functional data. Specifically, it has been added a metrics submodule, which contains functional metrics, among which there are the family of $\mathbb{L}^p$ metrics for multivariate functions, the Fisher-Rao distance or elastic metrics based on the framework of section 2.3.2.

3.2.5. Dataset module

In the dataset module have been included synthetic generators, used during the development of tests or examples. May be found generators of sinusoidal processes, multimodal samples or random warpings.

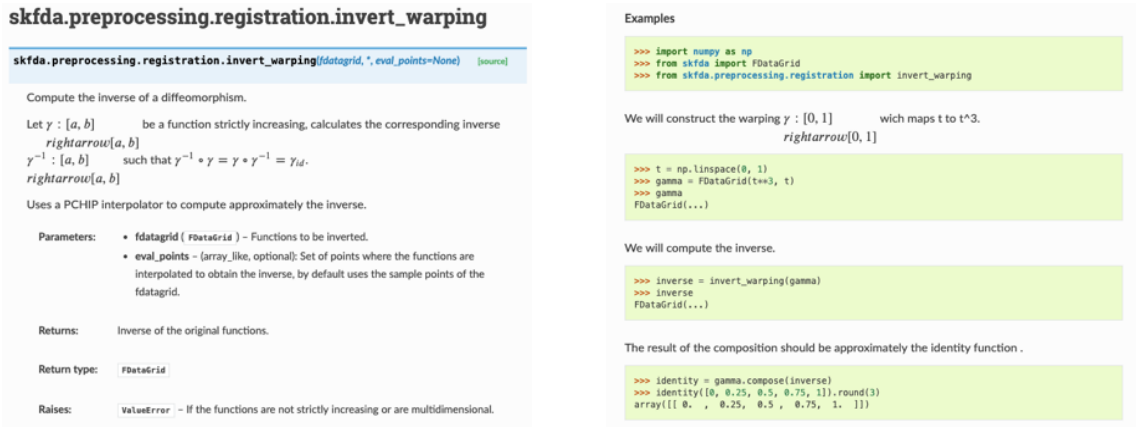
3.3. Coding, documenting and testing

Due to the fact that the project team is composed of several developers, who must review and integrate their work together, it has been necessary to establish a set of common coding standards.

In addition, because of the software-free character of the project, the code is public, and throughout its life will be maintained by multiple developers. For this reason, a great effort had to be dedicated to ensuring the readability of the code, along with its documentation, and compliance with strictly established standards. Among these standards are included two Python Enhancements Proposals (PEPs),

which are the directives that set the different conventions in Python: the PEP 8 - Style Guide for Python Code, and the PEP 257 -Docstring Conventions.

PEP 257 documents the semantics and conventions associated with Python docstrings, which allow include documentation within the code. These docstring are found next to the code as headers of modules, classes or functions. It has been used a google-like docstring style, based in the reStructuredText format. This style allows the automatic generation of documentation in pdf and html, using the Sphinx tool. These pages generated automatically are maintained by the continuous integration system and may be consulted online on https://fda.readthedocs.io/.



(a) Documentation of function

(b) Doctest included within the documentation

Figure 3.3: Scikit-fda online documentation

In addition, among the advantages of this kind of documentation is the possibility of including examples embedded in it, called doctests, which appears as dynamic short examples within de documentation, as it is shown in the figure 3.3(b).

These examples are in turn tests. When running the bench tests, using the tool pytest, the code is parsed looking for doctests, executing the code found in them and checking that the output matches with the output of the documentation.

However, the fundamental part of the testing is made up of unit tests, which are executed together with the doctests to check the integrity of the software.

A very relevant part of the package documentation is made up of examples, which are Python notebooks written as tutorials. Due to their extension, these examples can be found in Annex C or among the online documentation. Below it is shown the beginning of one of these examples.

HERE WILL BE AN EXAMPLE.

3.4. Development, version control and continuous integration

During the development of the package has been used an agile methodology, with similar precepts than the eXtreme programming [25], which is based on simplicity in development, continuous feedback and communication between team members.

For this communication in conjunction with version control git has been using, along with the functionalities available in Github. This platform offers a series of tools for the review and control of contributions. There are multiple workflows designed to carry out version control with git, of which we employ Gitflow.

Gitflow is a widely used workflow, which uses git branches to organize the work. It structures the code around two main branches: master and develop. The first is a stable branch, where any built-in commit must be ready to go into production and generates a new version. The second one, develop, is the code that will make up the next planned version of the project. To add new code to these branches, it is necessary to create other auxiliary ones that will be merged to develop by means of a pull request, after a review process and pass the bench tests. Figure 3.4 illustrates the mechanics of this workflow.

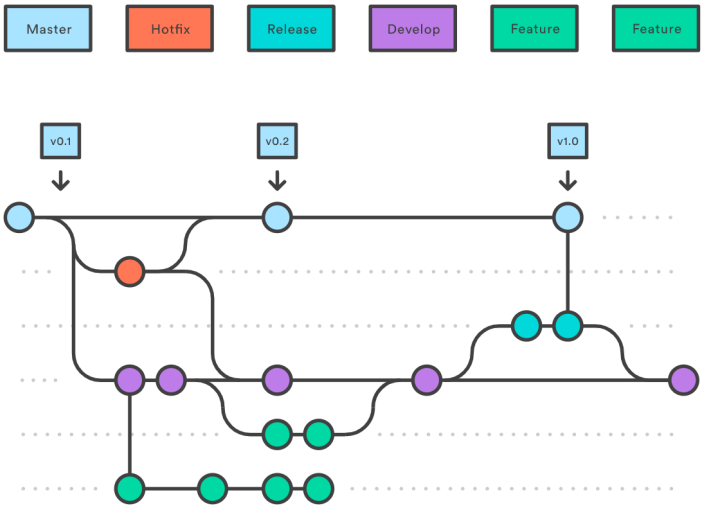


Figure 3.4: Example of git flow branches [1]

One of the main advantages of using github, over other version control systems, is the possibility of using a continuous integration mechanism. Travis CI has been used for this purpose, which compiles the library and runs the bench tests each time a commit is made, as in figure 3.5, where it may be seen a red cross or a green tick associated to each commit depending on the bench tests. In addition, it can also automate a wide variety of tasks, such as generating the documentation, compiling or integrating style reviewers.

[1] Picture from https://es.atlassian.com/git, licensed under a Creative Commons Attribution 2.5 Australia License

Figure 3.5: Example of travis checks

4

Conclusions and future work

The goal of this project is the creation of a Python reference package in FDA, along with the creation of a community to support and keep the project updated. So it will require the effort and dedication of a team of developers to achieve this goal.

During this year, the project has evolved drastically, redesigning and expanding its functionalities. Among the aspects that have allowed this development to be successfully achieved are communication between team members, with weekly meetings, and the use of an CI system which has made it much easier to integrate working together. In addition, the experience of the team throughout the last year in which the project was initiated has been fundamental.

In spite of all the advances made during this year there is still a lot of work to be done. Among these functionalities that it would be interesting to cover the basis representations of functional data, including support for surfaces, so that covariances can be represented in this form, and to support more types of basis, such as wavelets [**?**] or constrained basis [7].

It would also be interesting to extend registration techniques, incorporating functionalitiesfrom the *fdasrvf* [17] package, which include tools for clustering and registration, or for elastic alignment of spatial curves and surfaces.

Leaving aside these technical aspects, one of the most important tasks to carry out is to give visibility to the project, to discover the library to interested developers and researchers, presenting the package in congresses and publishing articles to promote its use.

Bibliography

[1] J. O. Ramsay, "When the data are functions," *PSYCHOMETRIKA*, vol. 47, no. 4, pp. 379–396, 1982.

[2] J. O. Ramsay, H. Wickham, S. Graves, and G. Hooker, "Cran R - fda package."

[3] J. O. Ramsay, G. Hooker, and S. Graves, *Functional Data Analysis with R and MATLAB*. Springer, 1 ed., 2009.

[4] M. Febrero-bande, "Statistical Computing in Functional Data Analysis :," *Journal of Statistical Softaware*, vol. 51, no. 4, pp. 1–28, 2012.

[5] M. Carbajo, *FDA-PY: Development of a python package for functional data analysis*. PhD thesis, Universidad Autónoma de Madrid, 2018.

[6] "III International Workshop on Advances in Functional Data Analysis," 2019.

[7] J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*. 2 ed., 2005.

[8] A. Srivastava and E. Klassen, *Functional and Shape Data Analysis*, vol. 49. Springer, 2016.

[9] C. R. de Boor, "B(asic)-Spline Basics," vol. 3, no. September, 1981.

[10] H. E. Jone and N. Bayley, "The Berkeley Growth Study," *Child Development*, vol. 12, no. Jun, pp. 167–173, 1941.

[11] P. Kokoszka and M. Reimherr, *Introduction to Functional Data Analysis*. Taylor & Francis Group, 2017.

[12] D. Bertsekas, "Dynamic Programming and Optimal Control," 1995.

[13] J. S. Marron, J. O. Ramsay, L. M. Sangalli, and A. Srivastava, "Functional Data Analysis of Amplitude and Phase Variation," *Statistical Science*, vol. 30, no. 4, pp. 468–484, 2015.

[14] A. Kneip and J. O. Ramsay, "Combining registration and fitting for functional models," *Journal of the American Statistical Association*, vol. 103, no. 483, pp. 1155–1165, 2008.

[15] A. Srivastava, W. Wu, S. Kurtek, E. Klassen, and J. S. Marron, "Registration of Functional Data Using Fisher-Rao Metric," no. March, 2011.

[16] J. D. Tucker, *Functional Component Analysis and Regression Using Elastic Methods*. PhD thesis, Florida State University, 2014.

[17] J. D. Tucker, "fdasrvf: Elastic Functional Data Analysis," 2017.

[18] N. Cencov, "Statistical Decision Rules and Optimal Inferences," *Translations ofMathematical Monographs*, vol. 53, 1982.

[19] A. Baíllo, A. Cuevas, and R. Fraiman, "Classification methods for functional data," in *The Oxford Handbook of Functional Data Analysis* (F. Ferraty and Y. Romain, eds.), ch. 10, pp. 259–297, Oxford University Press, 2010.

[20] N. Kumar, L. Zhang, and S. Nayar, "What is a good nearest neighbors algorithm for finding similar patches in images?," *Lecture Notes in Computer Science (including subseries Lecture Notes in*

Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 5303 LNCS, no. PART 2, pp. 364–378, 2008.

[21] T. Oliphant, "{NumPy}: A guide to {NumPy}." USA: Trelgol Publishing.

[22] E. Jones, T. Oliphant, and P. Peterson, "{SciPy}: Open source scientific tools for {Python}."

[23] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," pp. 1–15, 2013.

[24] C. Ramos, "scikit-fda," in *III International Workshop on Advances in Functional Data Analysis*, 2019.

[25] J. Blankenship, M. Bussa, and S. Millet, "eXtreme Programming," in *Pro Agile .NET Development with Scrum*, ch. 3, 2011.

Appendices

A

Algorithms and proofs

A.1. Shift registration by the Newton-Raphson algorithm

For the calculation of the values δ_i necessary for the registration of a set of functional samples $\{x_i\}_{i=1}^n$, according to ref, we will use a variation of the Newton-Rhapson's root-finding algorithm, applied to the derivative of REGSSE (2.3). This procedure is explained in more detail in Ramsay and Silverman (2005) [7].

For this computation it will be necessary to evaluate derivatives of x_i, so it will be crucial a previous smoothing step of the samples. Derivatives of REGGSE are given by:

$$\frac{\partial}{\partial \delta_i}\text{REGSSE} = 2\int_{\mathcal{T}} \left[x_i(t+\delta_i) - \hat{\mu}(t)\right] Dx_i(t)dt, \tag{A.1}$$

$$\frac{\partial^2}{\partial \delta_i^2}\text{REGSSE} = 2\int_{\mathcal{T}} \left[x_i(t+\delta_i) - \hat{\mu}(t)\right] D^2x_i(t)dt + 2\int_{\mathcal{T}} \left[Dx_i(t)^2\right] dt. \tag{A.2}$$

In practice the first term of $\frac{\partial^2}{\partial \delta_i^2}$**REGSSE** (A.2) it is deleted, because when the misalignment of the samples is large it can affect the convergence of the algorithm, and vanishes for the values that minimize the criterion. Therefore the following approximation is used:

$$\frac{\partial^2}{\partial \delta_i^2}\text{REGSSE} \approx 2\int_{\mathcal{T}} \left[Dx_i(t)^2\right] dt. \tag{A.3}$$

The initilization of $\delta_i^{(0)}$ in A.1 may be set to minimize some feature, or simply set $\delta_i^{(0)}$ to 0.

Could be used as stop criterion a maximun value of iterations along with a tolerance $|\delta_i^{(\nu)} - \delta_i^{(\nu-1)}| < \epsilon$. Generally the convergence is fast, obtaining good alignments with one or two iteration with a reasonable estimation of the initial values $\delta_i^{(0)}$. The step size α in A.1 may be set to 1.

Input : Set of functional observations $\{x_i(t)\}_{i=1}^n$
Output: Shifts $\{\delta_i\}_{i=1}^n$ used to register the data

Initialize $\delta^{(0)}$;
for *step* $\nu = 1, 2, \ldots$ *until stop criterion* **do**
    Update cross-sectional mean
    $\hat{\mu}(t) \leftarrow \frac{1}{n}\sum_{i=1}^n x_i(t + \delta_i^{(\nu-1)})$
    foreach $\delta_i^{(\nu-1)}$ **do**
    Update values of δ_i
    $\delta_i^{(\nu)} \leftarrow \delta_i^{(\nu-1)} - \alpha \frac{\partial}{\partial \delta_i} REGSSE / \frac{\partial^2}{\partial \delta_i^2} REGSSE$;
end

Algorithm A.1: Shift registration by Rhapson-Newton algorithm

A.2. Proofs of some mathematical results

In this section two fundamental results are proved, referenced in the section 2.3. These proofs have been extracted from [8] and [15]

Lemma 1: *Under the SRVF representation, the Fisher-Rao Riemannian metric becomes the standard metric.*

We can decompose the SRSF mapping into two steps: $f(t) \to \dot{f}(t) \to q(t) = sign(\dot{f})\sqrt{\dot{f}} = Q(\dot{f})$.

For any $v \in T_f(\mathcal{F})$, the differential of this mapping is $v(t) \to \dot{v}(t) \to w(t) = Q_{*,f(t)}(\dot{v}(t))$.

To evaluate the expression for w, we need the expression for Q_*. In case $x > 0$, we have $Q(x) = \sqrt{x}$ and its directional derivative in the direction of $y \in \mathbb{R}$ is $y/(2\sqrt{x})$. In case $x < 0$, we have $Q(x) = -\sqrt{-x}$ and its directional derivative is $y/(2\sqrt{-x})$. Combining the two, the directional derivative of Q is $Q_{*,x(y)} = y/(2\sqrt{|x|})$.

Let $v_1, v_2 \in T_f(\mathcal{F})$ be two vectors in the tangent space, and their mappings as $w_i(t) = \dot{v}_i(t)/(2\sqrt{|\dot{f}(t)|})$

Taking the $\mathbb{L}^2$ inner-product between the resulting tangent vectors, we get:

$$\langle w_1(t), w_2(t)\rangle = \int_0^1 w_1(t)w_2(t)dt = \frac{1}{4}\int_0^1 \dot{v}_1(t)\dot{v}_2(t)\frac{1}{|f(t)|}dt$$

The RHS is compared with Eqn.2.10 to complete the proof.

Lemma 2: *For any two SRVFs* $q1, q2 \in \mathbb{L}^2$ *and* $\gamma \in \Gamma$*, we have that* $\|(q_1, \gamma) - (q_2, \gamma)\| = \|q_1 - q_2\|$.

Let $\gamma \in \Gamma$ be and arbitrary warping and $q1, q2 \in \mathbb{L}^2$,

$$\|(q_1, \gamma) - (q_2, \gamma)\|^2 = \int_0^1 \left(q_1(\gamma(t))\sqrt{\dot{\gamma}(t)} - q_2(\gamma(t))\sqrt{\dot{\gamma}(t)}\right)^2 dt =$$

$$\int_0^1 \left(q_1(\gamma(t)) - q_2(\gamma(t))\right)^2 \dot{\gamma}(t)dt = \|q_1 - q_2\|^2 .\square$$

B

Example notebooks

C

Programmer's guide


UAM
UNIVERSIDAD AUTONOMA
DE MADRID