POLITECNICO
MILANO 1863

Industrial and Information Engineering School

Master of Science in Mathematical Engineering

Permutation tests for the equality of covariance operators of functional data with applications to evolutionary biology

Author

Alessandra Cabassi
Matr. 817358

Supervisors

Prof. Piercesare Secchi
Dr. Davide Pigoli

Academic Year 2015/2016

Preface

This thesis is made as a completion of the Master in Mathematical Engineering at the Polytechnic University of Milan. The research described therein has been conducted under the supervision of Doctor Davide Pigoli and Professor Piercesare Secchi. Part of the work has been developed at the the Statistical Laboratory of the University of Cambridge, between October and December 2015, and supported by the "Thesis abroad" funding provided by the Polytechnic University of Milan.

Foremost, I would like to express my sincere gratitude to my supervisors for offering me great guidance and advice and for giving me the extraordinary opportunity of visiting the University of Cambridge. Furthermore, I would like to thank all the members of the research group on functional data analysis at the Statistical Laboratory. I would especially like to acknowledge Adam Kashlak for his helpful comments and the brilliant idea of writing an R package together. I am also extremely grateful to Doctor Daniela De Angelis and Professor Alastair Young for their support and enthusiasm.

I take this occasion to express my appreciation to my family and all the friends I met during my time in Milan and Cambridge, with special thanks to the Association of Mathematical Engineers and the Celtic Team. Finally, my deepest gratitude goes to Nahuel for his unfailing encouragement throughout the whole course; this thesis is dedicated to him.

Milan, 28 July 2016

Abstract

The two-sample permutation test for the equality of covariance operators of functional data proposed by Pigoli et al. (2014) is extended to the case of multiple data samples. To this end, the non-parametric combination methodology of Pesarin and Salmaso (2010) is used to incorporate all the pairwise comparisons between the data samples into a global test. Different combining functions and permutation strategies are proposed and analysed in detail. The resulting test allows to make inference on the equality of the covariance operators of multiple groups and, if there is evidence to reject the null hypothesis, to identify the pairs of groups having different covariances. Additionally, a review of the most advanced methods for multiplicity control is presented. It is shown that, for some combining functions, step-down adjusting procedures are available. Also, the empirical power of this new test is computed and compared with those of the already existing tests for different case studies. These show that the performances are the same as those of the best methods proposed so far, whilst making less stringent assumptions on the data-generating process. Finally, the proposed methodology is applied to the data collected during the experiment described in Swallow et al. (1998), that used selective breeding to study the genetics and evolution of locomotor behaviour in mice.

Key words: functional data analysis, covariance operators, permutation tests, non-parametric combination.

Sommario

Il presente lavoro di tesi è dedicato all'estensione del test di permutazione per l'uguaglianza degli operatori di covarianza di due campioni di dati funzionali proposto da Pigoli et al. (2014), al caso di confronti multipli. A tal fine, il metodo di combinazione non parametrica di Pesarin e Salmaso (2010) viene utilizzato per incorporare tutti i confronti a coppie tra i campioni in un test globale. Diverse funzioni di combinazione e strategie di permutazione sono proposte ed analizzate nel dettaglio. Il test risultante permette di fare inferenza sull'uguaglianza degli operatori covarianza di più gruppi e, se c'è evidenza per rifiutare l'ipotesi nulla, di identificare le coppie di gruppi aventi covarianze differenti. Viene inoltre presentata una revisione dei più avanzati metodi per il controllo della molteplicità. In particolare, per alcune funzioni di combinazione sono disponibili procedure step-down di aggiustamento dei p-values. La potenza empirica del test viene calcolata e confrontata con quelle dei metodi già esistenti in diversi casi test. Questi mostrano che le prestazioni sono pari a quelle dei migliori test proposti finora, nonostante le ipotesi sul processo che genera i dati siano meno stringenti. Infine, il metodo è applicato ai dati raccolti durante l'esperimento descritto in Swallow et al. (1998), nel quale l'allevamento selettivo di topi da laboratorio è stato usato per studiarne la genetica e l'evoluzione del comportamento motorio.

Parole chiave: analisi di dati funzionali, operatori covarianza, test di permutazione, combinazione non parametrica.

Contents

Chapter 1

Introduction

1.1 Motivations

Since the 1990s, an increasing number of applications has involved the measurement of finely sampled curves. When applying traditional statistical methods to this kind of data, two issues arise: not only the the number of variables of interest is much larger than the number of observations (so-called large p small n problem), but there are also strong correlations between the covariates. The field of functional data analysis emerged from the necessity of overcoming these obstacles and has been widely studied during the last decades (see e.g. Ramsay and Silverman, 2005, Ferraty and Vieu, 2006 and Horváth and Kokoszka, 2012). However, most of the literature on hypothesis testing deals with the tests on the mean function (Fan and Lin, 1998, Cardot et al., 2003, Cuevas et al., 2004 and Shen and Faraway, 2004). The problem of testing the equality of covariance operators, instead, has been studied only recently. Panaretos et al. (2010) derived a functional testing procedure under the assumption of Gaussianity in the two-sample case; Fremdt et al. (2013) extended it to the non-Gaussian case. In these works, the critical points of the testing procedures are obtained using asymptotic approximations of the test statistics distributions under the null hypothesis. Due to the complicated statistical functionals involved, the efficacy of these tests heavily relies on the accuracy of the approximations. The testing of equality of several covariance operators has been introduced by Boente et al. (2014): in order to improve asymptotic approximations, they proposed to apply a bootstrap procedure to calibrate the critical values of the considered test statistic. After that, Paparoditis and Sapatinas (2014) investigated the properties of an alternative and general bootstrap-based testing methodology, applicable to more than two populations, but its consistency has been proven only for a few test statistics. Also, a permutation test for the equality of covariance operators in the two-sample case has been proposed by Pigoli et al. (2014): it can be used with any test statistic and makes no assumptions on the data other than the exchangeability under the null hypothesis. More recently, Kashlak et al. (2016) applied concentration inequalities to the analysis of covariance operators. These allow to construct non-asymptotic confidence sets which can be used to make multiple-sample tests for the equality of covariance operators.

1.2 Contributions

This thesis has the aim of extending the two-sample permutation test for covariance operators proposed in Pigoli et al. (2014) to the multiple sample case. In particular, let us consider q samples of random curves. We assume that curves in sample i:

$$x_{i1},\dots,x_{in_i} \in L^2(I), \quad i = 1,\dots,q$$

are realisations of a random process with mean μ and covariance operator Σ_i. We would like to test the hypothesis

$$H_0 : \{\Sigma_1 = \Sigma_2 = \cdots = \Sigma_q\} \quad \text{against} \quad H_1 : \{\text{at least one equality is not true}\}.$$

Moreover, if the null hypothesis H_0 is rejected, we would like to identify all the pairs of samples that led to that conclusion. The starting point is the non-parametric combination methodology for permutation tests of Pesarin and Salmaso (2010), which enables to combine many different partial tests in an overall test. In our case, the idea is to combine all the pairwise comparisons between the q samples in order to obtain a global p-value. Using this method, the post-hoc comparisons are straightforward: the partial p-values of the paired comparisons are computed simultaneously. However, some care is required when jointly analysing them, because a multiplicity problem arises. To this end, we do a review of the most advanced methods for the control of the family-wise error rate and choose the most appropriate ones with respect to conservativity and computational cost. Also, the empirical power of this new test is computed and compared with those of the already existing tests for different case studies. These show that the performances are the same as those of the best methods proposed so far, whilst making less stringent assumptions on the data generating process. Lastly, the new test is applied to the data collected during an experiment of evolutionary biology, described in the next section.

All the functions implemented during the thesis to perform the permutation test for the equality of the covariance operators have been collected in the R package `fdcov`. This includes also the code related to the work of Kashlak et al. (2016), who developed an analogous test, based on non-asymptotic confidence sets.

1.3 Evolutionary biology dataset

We introduce here the dataset that motivated the extension of the two-sample test to the multiple sample case. Data were collected during the experiment presented in Swallow et al. (1998). The objective of the experiment was to use selective breeding to study the genetics and evolution of locomotor behaviour in mice. To this end, four replicate lines of laboratory house mice, *Mus domesticus*, have been compared to other four random-bred lines maintained as controls. The selection criterion was the total number of revolutions run on days 5 and 6 of a 6-day test. In the selected lines, the highest-running male and female from each family were chosen as breeders. In each generation, 10 pairs of mice were used to propagate each of eight lines. In the control lines, one male and one female from each family were chosen randomly with respect to wheel running. Each of the eight lines was represented by ten males and ten females. Data presented here refer to the thirteenth generation. Approximately half the individuals were sacrificed after 80 weeks; the rest of the individuals were allowed to live out their lifespan. One male from one of the selected lines died of unknown causes during the early stages of the experiment. The variables in the dataset are:

- `mouseid`: unique id number for each mouse;
- `sex`: females = 0; males = 1;
- `age`: age in days when wheel data were taken;
- `family`: id of full-sib family from which mouse was drawn;
- `line`: lines 1, 2, 4, 5 = control; 3, 6, 7, 8 = selected;
- `linetype`: control = 0; selected = 1;
- `week`: week of wheel measure;
- `whlrev`: number of revolutions run in a given week.

Total activity, measured as number of revolutions run in a given day, can be decomposed into the product of mean velocity and duration of activity. Thus, the evolution of increased total activity levels could be accomplished by an increase in mean velocity, an increase in the amount of time spent running, or a combination of both. In Figure 1.1 are represented the raw data. Each line connects the number of revolutions done by a specific mouse during the first 80 weeks of the experiment. Weeks from 81 on are not considered in the present study since only a few observations are available. Mice identified by ID numbers 90183 and 90224 will be taken as an example of the selected and control lines respectively throughout the whole work. The corresponding wheel-running functions have been highlighted in each figure. The first one is a male belonging to family number 29 and therefore line 1 (control), while the second one is a female belonging to family number 11, line 3 (selected). At several times during the experiment, data collection was skipped for one or two weeks. In this instances, the data collected after the skipped week(s) was divided by number of weeks, giving multiple weeks in a row with the same value. This is easily seen in Figure 1.1 at weeks 38, 39, 40, when the values are constant for each mouse, because the wheel revolutions recorded for week 40 were divided by 3 and assigned to weeks 38 and 39 as well as 40. The weeks in which this occurred are: $34, 35, 38, 39, 40, 50, 51, 72, 73$.

1.4 Thesis outline

The material is organised as follows:

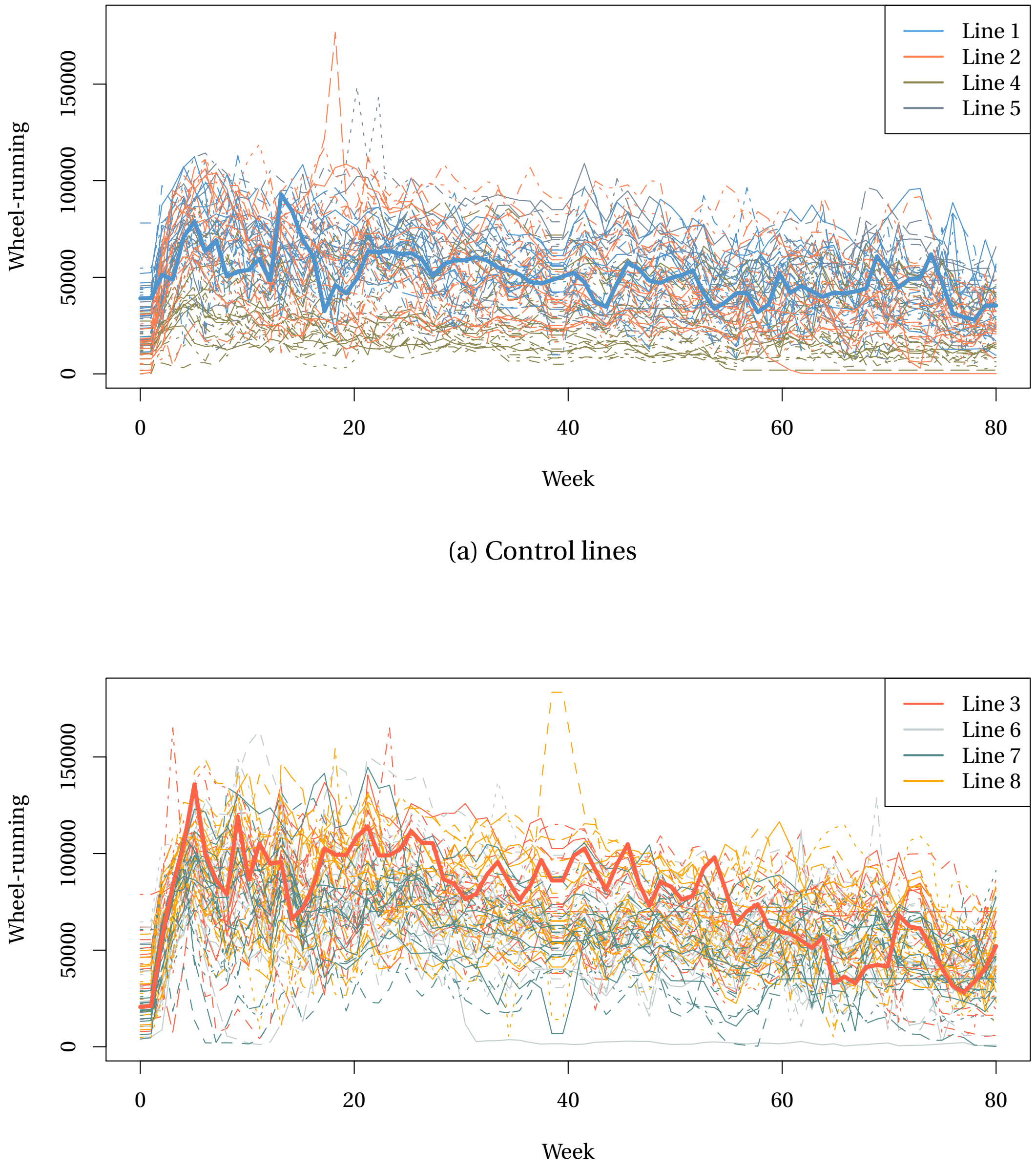
Chapter 2: Functional data analysis. The fundamental definitions of functional data analysis are presented before getting into details regarding covariance operators. Additionally, the techniques used for data preprocessing such as smoothing and alignment are described and applied to the evolutionary dataset.

Chapter 3: Permutation tests. This chapter deals with the theoretical foundations of permutation tests, the statistical tool that allows us to make inference on complicated mathematical objects without having to elicit their distribution. Then, the focus shifts on the algorithm for non-parametric combination of permutation tests, which is explained in detail.

Chapter 4: State of the art of equality tests for covariance operators. In this chapter, the existing methodologies for the test of the equality of covariance operators are explained.

Chapter 5: Permutation tests for covariance operators. This chapter describes the original methodological contributions of the thesis. Here it is explained how the non-parametric methodology can be used in order to extend the permutation test of Pigoli et al. (2014) to the multiple sample case. Also, different types of permutations are proposed and analysed for this specific application of this flexible method.

Chapter 6: Post-hoc comparisons. Once the global null hypothesis has been rejected, one may be interested in the pairwise comparisons between the groups. Using the non-parametric method, the p-values of each pairwise test are already known at this point. In this chapter we show how they can be adjusted in order to control the type I error of the whole procedure. In particular, we show that, using the Tippett combination function and the resampling-based multiple testing, it is possible to achieve a greater test power.



(a) Control lines

(b) Selected lines

Figure 1.1: Evolutionary biology dataset, raw data.

Chapter 7: Simulation studies. This chapter contains all the simulation studies that have been performed in order to assess the empirical power of the proposed methodology. When possible, this has been compared to those of the other existing tests explained in Chapter 4.

Chapter 8: Application to evolutionary biology. This chapter is dedicated to the analysis of the dataset presented in Section 1.3. First, it is proved that the developed methodology can be applied even if one observation is missing, then the non-parametric test for the equality of covariance operators is applied to the wheel-running activity functions.

Appendix A: Documentation of the R package. Here is reported the documentation of the R package "fdcov" that contains all the functions needed to perform the permutation tests for the equality of covariance operators and the corresponding post-hoc comparisons.

Appendix B: Code. This appendix contains the code implemented for the present work. Therefore, it includes all the functions needed in order to perform the proposed test, along with those related to the other considered tests.

Chapter 2

Functional data analysis

Functional Data Analysis (FDA) is a branch of statistics that analyses data providing information about curves, surfaces or other objects varying over a continuum. In its most general form, under an FDA framework each sample element is considered to be a function. The physical continuum over which these functions are defined is often time, but may also be spatial location, wavelength, etc. In many applications, data are recorded as samples from the generating curves. Thus, for each observation, we only have a finite number of values. Nonetheless, those values are used to approximate the generating curve, so that they can be analysed with tools specific to functional data. For this reason, FDA can be seen not only as the natural extension of Multivariate Data Analysis (MDA) that originated from the recent increase of dimensionality of data, but also as the abstraction of univariate analysis, in the sense that it considers the observations as points of a functional space, instead of the real line. This is also a special case of Object Oriented Data Analysis (OODA), the statistical analysis of data sets of complex objects Wang et al. (2007). According to Ramsay and Dalzell (1991), there are several practical reasons for considering data analysis from a functional perspective:

- *Functional observations present themselves in applied contexts increasingly more frequently;*
- *Some modelling problems are more natural to think through in functional terms even though only finite numbers of observations are available;*
- *The objectives of an analysis can be functional in nature, as would be the case if finite data are used to estimate an entire function, its derivatives or the values of other functionals;*
- *Taking considerations such as smoothness into account for multivariate data arising from functional processes can have important implications for their analyses.*

Since the aim of this work is to respond to issues related to the study of covariance for this particular kind of data, in this chapter we introduce a few basic concepts about the analysis of functional data. First we give the main definitions and then we focus on the properties of covariance operators and the different ways that have been proposed to measure differences between covariance operators. At the end of the chapter, we briefly describe the preprocessing steps that are needed to go from the raw discrete observations to functional objects that can be analysed in this framework.

2.1 Mathematical framework

In this section we give a brief review of the mathematical setting of functional data analysis. In the first two subsections the main definitions such as mean and covariance are given in the most general case of random variables in Hilbert spaces. After that, we focus on the particular case of functional variables belonging to the space of square integrable functions, which is the one considered in the remainder of this work. More details about the topics covered in this section can be found in Ramsay and Silverman (2005), Ferraty and Vieu (2006) and Horváth and Kokoszka (2012).

2.1.1 Random elements in Hilbert spaces

Suppose one observes a variable of interest X on an interval $I = [t_{\min}, t_{\max}]$ of successive instants. In many applications, thanks to the modern technologies, it is possible to obtain measurements of the phenomena of interest on a very fine grid. Therefore, even if each datum can be seen as an observation of the random family $\{X(t_j)\}_{j=1,\ldots,J}$, it is often more interesting to consider it as part of the continuous family $\{X(t), t \in I\}$ to take advantage of the intrinsic order between observations and of the smoothness of the underlying process. In this work we will consider the random family $\{X(t_j)\}_{j=1,\ldots,J}$ as a sampling of a function X defined from I, a compact subset of $\mathbb{R}$, to $\mathbb{R}$. Nevertheless, FDA can be used also when data correspond to surfaces, vectors of curves or even more complex infinite-dimensional objects. With this in mind, we give the following definitions, from Ferraty and Vieu (2006).

Definition 2.1. *A random variable X is called* functional variable *(f.v.) if it takes values in an infinite dimensional space (or functional space). An observation x of X is called a* functional datum.

Definition 2.2. *A* functional dataset *$x_1, \ldots, x_n$ is the observation of n functional variables $X_1, \ldots, X_n$ identically distributed as X.*

Now, let $(\Omega, \mathcal{M}, \mathbb{P})$ be a probability space and $\mathcal{H}$ a vector space, with a σ-algebra $\mathcal{N}$. In particular, suppose that $\mathcal{H}$ is an Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and the induced norm $\|\cdot\|_{\mathcal{H}}$ (from now on they shall be indicated as $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$). Then, a random functional variable is a measurable function X

$$X : \Omega \to \mathcal{H},$$

while a functional datum is the function

$$x = X(\omega) : I \to \mathbb{R}$$

for some fixed $\omega \in \Omega$, i.e. a realisation of X. We can now introduce the generalisation of the concepts of mean, variance and covariance to the case of functional data.

Definition 2.3. *Suppose that X is integrable with respect to the measure $\mathbb{P}$. Then the* mean *of X is defined as*

$$\mu = \mathbb{E}[X] = \int_{\Omega} X(\omega)\mathbb{P}(d\omega).$$

Median and mode can be defined analogously (Ferraty and Vieu, 2006). It is also possible to define a covariance operator as follows:

Definition 2.4. *Suppose that X is such that $\mathbb{E}[\|X\|^4] < \infty$. The* covariance operator Σ *of X, is the operator that associates to each $y \in \mathcal{H}$ a $\Sigma(y) \in \mathcal{H}$ such that*

$$\Sigma(y) = \mathbb{E}[\langle X - \mathbb{E}[X], y\rangle (X - \mathbb{E}[X])].$$

2.1.2 Estimation of mean and covariance

In applications, one observes a sample consisting of n curves $x_1, x_2, \ldots, x_n$. Each curve is a realization of a random function X. We assume that the functional variables $X_1, X_2, \ldots, X_n$ generating the observations are independent and identically distributed in $\mathcal{H}$ and have the same distribution as X, which is square integrable.

Definition 2.5. *Given a random sample $x_1, x_2, \ldots, x_n$ we define the following sample estimators:*

- Sample mean

$$m = \frac{1}{n}\sum_{i=1}^{n} x_i;$$

– Sample covariance operator

$$S(y) = \frac{1}{n-1}\sum_{i=1}^{n}\langle x_i, y\rangle x_i, \; t \in I.$$

2.1.3 Functions of a real variable

From now on, for simplicity, we will suppose that the considered data belong to the Hilbert space of square-integrable functions on $I \subseteq \mathbb{R}$, $L^2(I)$, equipped with the usual inner product

$$\langle x, y\rangle = \int_I xy dl, \quad \forall x, y \in L^2(I)$$

where l denotes the Lebesgue measure. In this case we can define the mean function of $X \in L^2(I)$ as

$$\mu(t) = \mathbb{E}[X(t)], \quad t \in I.$$

Regarding the variability measure, two definitions must be introduced:

Definition 2.6. *Suppose that* $\mathbb{E}[X^2(t)] < \infty$ *for all* $t \in I$. *Then we can define the* variance function *as*

$$\gamma(t) = \mathbb{E}[(X(t) - \mu(t))^2], \quad t \in I.$$

Moreover, we can define the autocovariance function *as*

$$\Gamma(s,t) = Cov(X(s), X(t)) = \mathbb{E}[(X(s) - \mu(s)) \cdot (X(t) - \mu(t))], \quad s, t \in I.$$

Notice that this definition is strictly linked to Definition 2.4. In fact, for all $y \in L^2(I)$, it holds

$$\begin{aligned} \Sigma(y)(t) &= \langle \Gamma(t,\cdot), y\rangle, \\ &= \int_I \sigma(s,t) y(s) ds, \quad \forall y \in L^2(I), t \in I, \end{aligned} \tag{2.1}$$

where $\sigma(t,s) = \mathbb{E}[(X(t) - \mu(t)) \cdot (X(s) - \mu(s))]$. Clearly, $\sigma(t,s) = \sigma(s,t)$ and

$$\begin{aligned} \int\int \sigma(t,s) y(t) y(s) dt ds &= \int\int \mathbb{E}[(X(t) - \mu(t)) \cdot (X(s) - \mu(s))] y(t) y(s) dt ds \\ &= \mathbb{E}\Big[\Big(\int (X(t) - \mu(t)) y(t) dt\Big)^2\Big] \geq 0. \end{aligned}$$

Definition 2.7. *Given a random sample* $x_1, x_2, \dots, x_n$ *we define the following sample estimators:*

– Sample mean function

$$m(t) = \frac{1}{n}\sum_{i=1}^{n} x_i(t), \quad t \in I;$$

– Sample variance function

$$c^2(t) = \frac{1}{n-1}\sum_{i=1}^{n} (x_i(t) - m(t))^2, \quad t \in I;$$

– Sample autocovariance function

$$C(s,t) = \frac{1}{n-1}\sum_{i=1}^{n} (x_i(s) - m(s))(x_i(t) - m(t)), \quad s, t \in I.$$

2.2 Covariance operators

Since the main focus of this work is on inference for the covariance of functional data, this section is dedicated to an in-depth study of covariance operators, starting from their main theoretical properties and then introducing the concept of distance between covariance operators. The main focus are the distances used in the present study. Other distances and metrics can be found in Pigoli et al. (2014).

2.2.1 Main properties

We introduce here the basics definitions we need in the following chapters. For further details about operator theory in function spaces, refer to Zhu (2007).

Definition 2.8. *Let* B_1 *be the closed ball of unitary radius in* $L^2(I)$, *consisting of all* $y \in L^2(I)$ *such that* $\|y\|_{L^2(I)} \leq 1$. *A bounded linear operator* $K : L^2(I) \to L^2(I)$ *is* compact *if* $K(B_1)$ *is compact in the norm of* $L^2(I)$.

An important property of a compact operator on $L^2(I)$ is the existence of a *canonical decomposition.* This decomposition implies that two orthonormal bases $\{v_n\}_{n\in\mathbb{N}}$, $\{z_n\}_{n\in\mathbb{N}}$ for $L^2(I)$ exist so that

$$Ky = \sum_{n\in\mathbb{N}} \rho_n \langle y, v_n \rangle u_n,$$

or, equivalently, $Kv_n = \rho_n u_n$ and $\{\rho_n\}_{n\in\mathbb{N}} \in \mathbb{R}$ is called the sequence of *singular values* for K. If the operator is self-adjoint, there exists a basis $\{v_n\}_{n\in\mathbb{N}}$ such that

$$Ky = \sum_{n\in\mathbb{N}} \phi_n \langle y, v_n\rangle v_n,$$

or, equivalently, $Kv_n = \phi_n v_n$. $\{\phi_n\}_{n\in\mathbb{N}} \in \mathbb{R}$ is called the sequence of *eigenvalues* of K.

Definition 2.9. *A compact operator K is said to be* trace-class *if*

$$\mathit{trace}(K) := \sum_{n\in\mathbb{N}} \langle Ke_n, e_n\rangle < +\infty$$

for an orthonormal basis $\{e_n\}_{n\in\mathbb{N}}$*. We indicate with* $\mathcal{S}(L^2(I))$ *the space of the trace-class operators on* $L^2(I)$*.*

Definition 2.10. *A compact operator K is said to be* Hilbert-Schmidt *if its Hilbert-Schmidt norm is bounded, i.e.,*

$$||K||^2_{HS} = \mathit{trace}(K'K) < +\infty.$$

This is a generalisation of the Frobenius norm for finite-dimensional matrices. Then the covariance operator defined by Equation (2.1) is a trace-class, self-adjoint, compact operator on $L^2(I)$ with nonnegative eigenvalues (Bosq, 2012, Section 1.5).

2.2.2 Distances between covariance operators

The distances between covariance operators considered in this work are some of those proposed in Pigoli et al. (2014). These are a generalisation to the functional setting of metrics that have been proved useful for the case of positive semi-definite matrices (Dryden et al., 2009).

Distance between kernels in $L^2(I \times I)$

Distances between covariance operators can be naturally defined using the distance between their kernels in $L^2(I \times I)$. Let Σ_1 and Σ_2 be two covariance operators and

$$\Sigma_i y(t) = \int_I \sigma_i(s,t) y(s) ds, \quad \forall y \in L^2(I).$$

We can define a distance between their kernels as

$$d_L(\Sigma_1, \Sigma_2) = \|\sigma_1 - \sigma_2\|_{L^2(I\times I)} = \sqrt{\int_I\int_I (\sigma_1(s,t) - \sigma_2(s,t))^2 dsdt}.$$

This corresponds to the distance induced by the Hilbert-Schmidt norm, since for Hilbert-Schmidt kernel operators one has

$$\|\Sigma_1 - \Sigma_2\|_{HS} = \|\sigma_1 - \sigma_2\|^2_{L^2(I\times I)}.$$

This distance is well-defined, since it inherits all the properties of the distance in the Hilbert space $L^2(I \times I)$. However, it does not exploit in any way the particular structure of covariance operators and therefore its suitability to highlight significant differences between covariance structures is questionable.

Spectral distance

Considering covariance operators as elements of $\mathcal{L}(L^2(I))$, the space of the linear bounded operators on $L^2(I)$, Pigoli et al. (2014) defined another distance between Σ_1 and Σ_2 as the operator norm of the difference between the two. Remembering that the norm of a self-adjoint bounded linear operator on $L^2(I)$ is

$$\|K\|_{\mathcal{L}(L^2(I))} = \sup_{y\in L^2(I)} \frac{|\langle Ky, y\rangle|}{\|y\|^2_{L^2(I)}}$$

and that for a covariance operator this coincides with the absolute value of the largest eigenvalue, we have

$$d_L(\Sigma_1, \Sigma_2) = \|\Sigma_1 - \Sigma_2\|_{\mathcal{L}(L^2(I))} = |\tilde{\phi}_1| \tag{2.2}$$

where $\tilde{\phi}_1$ is the largest eigenvalue of the operator $\Sigma_1 - \Sigma_2$. This distance generalises the matrix spectral norm often used in the finite-dimensional case (El Karoui, 2008). It takes into account the spectral structure of the covariance operators, but focuses only on the behaviour of the first mode of variation. This can describe effectively the distance between the operators only if $\tilde{\phi}_1$ explains the majority of the variation, which is often not the case in practical applications.

Square root operator distance

Since covariance operators are trace-class, it has also been possible to generalise the square root matrix distance of Dryden et al. (2009). Σ being a self-adjoint trace-class operator, there exists a Hilbert-Schmidt self-adjoint operator

$$(\Sigma)^{1/2} y = \sum_n \phi_n^{1/2} \langle y, v_n\rangle v_n,$$

where ϕ_n and v_n are the eigenvalues and eigenfunctions of Σ respectively. Then, Pigoli et al. (2014) define the square root distance between two covariance operators Σ_1 and Σ_2 as

$$d_R(\Sigma_1, \Sigma_2) = \|(\Sigma_1)^{1/2} - (\Sigma_2)^{1/2}\|_{HS}.$$

Inspiration for this kind of distance comes from the log-Euclidean distance for positive definite matrices. In that case, the logarithmic transformation allows to map the non Euclidean space in a linear space. Unfortunately, a logarithmic map for covariance operators is not available. Instead, the square root transformation has been shown to behave in a similar way in the finite dimensional setting (Dryden et al., 2009) and is well defined for trace-class operators. In theory, any power greater than $1/2$ would be a possible candidate distance. As explained in Pigoli et al. (2014), the reasons for this particular choice are twofold: first, for general trace-class operators, the square root operator is the smallest power that can be defined while still ensuring finite distances, meaning that it is the closest available to the log-Euclidean distance; in addition, this case can be interpreted as a distance which takes into account the full eigenstructure of the covariance operator (i.e. both eigenfunctions and eigenvalues).

Procrustes size-and-shapes distance

First of all, we recall the definition of unitary operator on $L^2(I)$ given by Zhu (2007).

Definition 2.11. *A bounded linear operator U on $L^2(I)$ is said to be* unitary *if*

$$\|Uy\|_{L^2(I)} = \|y\|_{L^2(I)}, \quad \forall y \in L^2(I).$$

Equivalently, we say that U is unitary if $UU' = 1$, where U' is the adjoint of U. Then, a Procrustes[1] size-and-shapes distance between covariance operators is defined by Pigoli et al. (2014) as

$$\begin{aligned} d_P(\Sigma_1, \Sigma_2)^2 &= \inf_{U \in \mathcal{O}(L^2(I))} \|D_1 - D_2 U\|_{HS}^2 \\ &= \inf_{U \in \mathcal{O}(L^2(I))} \operatorname{trace}((D_1 - D_2 U)'(D_1 - D_2 U)), \end{aligned}$$

where D_i are such that $\Sigma_i = D_i D_i'$ for $i = 1, 2$ and $\mathcal{O}(L^2(I))$ is the space of unitary operators on $L^2(I)$. The motivation for the definition of this distance is that the distance between the square root operators is only a particular choice in the broad family of distances based on the mapping of the two operators Σ_1 and Σ_2 from the space of covariance operators to the space of Hilbert-Schmidt operators. We may consider a generic transformation $S_i \mapsto D_i$, so that

[1] In Greek mythology, Procrustes was a rogue smith and bandit from Attica who physically attacked people by stretching them or cutting off their legs, so as to force them to fit the size of an iron bed. In general, when something is Procrustean, different lengths or sizes or properties are fitted to an arbitrary standard.

$S_i = D_i' D_i$ and define the distance as the Hilbert-Schmidt norm of $D_1 - D_2$. However, such transformations would be defined up to a unitary operator U:

$$(D_i U)(D_i U)' = D_i U U' D_i' = D_i D_i' = \Sigma_i.$$

The Procrustes approach avoids the arbitrariness of the transformation by choosing the unitary operator U which best matches the two operators D_1 and D_2. The following result shows an easier expression of this distance that will prove useful for its approximate practical computations.

Proposition 2.1. *Let ρ_k be the singular values of the compact operator $D_2' D_1$. Then*

$$d_P(\Sigma_1, \Sigma_2)^2 = \|D_1\|_{HS}^2 + \|D_2\|_{HS}^2 - 2 \sum_{n=1}^{+\infty} \rho_n.$$

The proof can be found in the supplementary material of Pigoli et al. (2014).

2.2.3 Finite dimensional approximation

In practical applications, we observe only a finite dimensional representation of the operators of interest. Therefore we want the square root distance and the Procrustes size-and-shape distance between two finite dimensional representations to be good approximations of the distance between the corresponding infinite dimensional covariance operators. In Pigoli et al. (2014) this is shown to be true in the case of Procrustes distance, with the square root distance being a special case where $D_i = (S_i)^{1/2}$ and U is the identity operator. The proof is based on the fact that if $\{e_n\}_{n\in\mathbb{N}}$ is a basis for $L^2(I)$, then $V_p = \text{span}\{e_1, \ldots, e_p\}$ is the subspace which contains the finite dimensional representations of functional data and

$$S_i^p y = \sum_{n=1}^{p} \langle y, e_n \rangle S_i e_n, \quad \forall y \in V_p$$

is the restriction of S_i on V_p. Now, if D_i^p is such that $D_i^p \to L_i$ for $p \to \infty$ with respect to the Hilbert-Schmidt norm, then the distance between the two restricted operators is

$$d_P(S_1^p, S_2^p)^2 = \|D_1^p\|_{HS}^2 + \|D_2^p\|_{HS}^2 - 2 \sum_{n=1}^{p} \langle \tilde{U}^p D_2^{p'} D_1^p e_n, e_n \rangle.$$

Since the subspaces V_p are nested, we can define a permutation $s:\mathbb{N}\to\mathbb{N}$ so that $\{v_{s(1)},\dots,v_{s(p)}\}$ is a basis for V_p, for every $p\in\mathbb{N}$. We obtain

$$\begin{aligned} d_P(S_1^p,S_2^p)^2 &= \|D_1^p\|_{HS}^2+\|D_2^p\|_{HS}^2-2\sum_{n=1}^{p}\langle \tilde{U}^p D_2^{p'} D_1^p v_{s(n)}, v_{s(n)}\rangle \\ &= \|D_1^p\|_{HS}^2+\|D_2^p\|_{HS}^2-2\sum_{n=1}^{p}\rho_{s(n)} \end{aligned}$$

where $\{\sigma_{s(k)}\}$ are singular values for $D_2'D_1$. This comes from the fact that the action of an operator $D_2^{p'}D_1^p$ should be equal to the action of the operator $D_2'D_1$ on every element belonging to the subspace V_p and $v_{s(k)}\in V_p$ for $k=1,\dots,p$. Finally, as $D_2'D_1$ is trace-class, the series of its singular values is absolutely convergent and therefore unconditionally convergent, i.e. convergent under any permutation. Thus,

$$\begin{aligned} \lim_{p\to\infty} d_P(S_1^p,S_2^p)^2 &= \|D_1\|_{HS}^2+\|D_2\|_{HS}^2-2\sum_{k=1}^{+\infty}\rho_{s(k)} \\ &= \|D_1\|_{HS}^2+\|D_2\|_{HS}^2-2\sum_{k=1}^{+\infty}\rho_{k} \\ &= d_P(S_1,S_2)^2. \end{aligned}$$

2.3 Data pre-processing

In applications, a functional dataset can originate from data collection in two ways:

1. Data are represented by their analytical expressions;
2. The data set is made of a collection of discrete data corresponding to measurements of the function for consecutive values of the argument.

In the first case, one can easily exploit data analytical and differential properties by explicit computation. The latter case is more frequent and requires to represent the observations as functions, before starting the analysis.

2.3.1 Smoothing

As in Section 2.1.1, let us suppose that $t_1,\dots,t_J$ are points in $I=[t_{\min},t_{\max}]$ satisfying $t_{\min}=t_1<t_2<\cdots<t_J=t_{\max}$ and $\{x(t_j)\}_{j=1,\dots,J}$ is an observation of the random family $\{X(t_j)\}_{j=1,\dots,J}$. We assume throughout this section that $J\geq 3$. In other words, we assume the existence of a

random function X giving rise to the observed data. We usually want the underlying function x to be smooth, so that adjacent data values are linked together and unlikely to be different to each other. If this smoothness property did not apply, there would be nothing much to be gained by treating the data as functional rather than just multivariate. By smooth, we mean that function x possesses one or more derivatives. Smoothness, in this sense, is a property of the latent function x. Raw data vectors, instead, usually contain observational errors. We express this as

$$y_j = x(t_j) + \epsilon_j$$

where the otherwise exogenous term ϵ_j contributes a roughness to the raw data, that can be due to noise or measurements errors. When representing the raw data as functions we want to filter out this exogenous term as efficiently as possible. Equivalently, in vector notation, we have

$$\mathbf{y} = x(\mathbf{t}) + \boldsymbol{\epsilon},$$

where $\mathbf{y}$, $x(\mathbf{t})$, $\mathbf{t}$ and $\boldsymbol{\epsilon}$ are column vectors of length n.

At the end of the last century there has been an upsurge of interest and activity in the area of non-parametric smoothing in statistics. Many methods have been proposed and studied so far: see e.g. Hastie and Tibshirani (1990), Green and Silverman (1993) and Ramsay and Silverman (2005). The latter will be used as a reference for the following exposition.

In general, a smoothing operator estimates the function value $\hat{x}(t_j)$ starting from the discrete observations

$$\hat{x}(t_j) = \sum_{i=1}^{p} F_j(t_i) y_i,$$

where $F_j(t_i)$ weights the ith discrete data value in order to estimate $x(t)$. In matrix terms, $\hat{x}(\mathbf{t}) = F\mathbf{y}$, where $\hat{x}(\mathbf{t})$ is a column vector containing the values of the estimate of function x at each sampling point t_j.

Usually, one wants to approximate $x(t)$ as a weighted sum of some set of known functions π_i belonging to a basis function system of $L^2(I)$. Then, the function x can be seen as the linear expansion of dimension k

$$x(t) = \sum_{i=1}^{k} c_i \pi_i = \mathbf{c}'\boldsymbol{\pi}.$$

Both k and the basis itself are parameters of the above representation. Ideally, the chosen basis function should have features matching those of the estimated functions, in order to achieve satisfactory approximations with a small number of basis functions. In the literature, many different bases systems have been used: polynomial, step-functions, exponential, wavelets, Fourier, splines, etc. For the present work we have chosen to use splines, because they are suitable for functional data without any strong cyclic variation.

Remembering that I is the compact subset of $\mathbb{R}$ over which the function x lies, let us split it into r subintervals separated by the so-called *breakpoints* (or *knots*) and denote them as $\iota_i, i = 1, \dots, r$. A spline of order u is composed by polynomials of degree $u-1$. Each one of them lies in one subinterval. Adjacent polynomials join up smoothly at breakpoints so that derivatives up to the order $u-2$ match at breakpoints. In particular, we use B-splines, i.e. splines having the compact-support property of being non-zero over at most u adjacent intervals. In this way, the computational cost of using splines reduces to the same as orthonormal bases (such as Fourier and wavelets, for example).

The simplest linear smoother is the one that determines the coefficients of the expansion by minimising the least squares criterion:

$$\texttt{SMSSE}(\mathbf{y}|\mathbf{c}) = (\mathbf{y} - \Pi\mathbf{c})'(\mathbf{y} - \Pi\mathbf{c})$$

However, this is based on the assumption that the ϵ_j are independent and identically distributed with zero-mean and variance σ^2 which, in most applications, is not the case. In order to deal with non-stationary errors, we have to modify the least squares criterion:

$$\texttt{SMSSE}(\mathbf{y}|\mathbf{c}) = (\mathbf{y} - \Pi\mathbf{c})'W(\mathbf{y} - \Pi\mathbf{c})$$

where W is a symmetric positive definite matrix allowing unequal weighting of the errors and Π is the matrix containing in row i the values of π_i evaluated at $t_1, \dots, t_{n_\cdot}$, i.e. $\pi(\mathbf{t})'$.

The method used in the present work combines the B-spline basis system and the roughness penalty objective. The curve x is estimated from observations y_j by finding the function minimising the Penalised Sum of Squared Errors (PENSSE)

$$\texttt{PENSSE}_\delta(x|\mathbf{y}) = [\mathbf{y} - x(\mathbf{t})]'W[\mathbf{y} - x(\mathbf{t})] + \delta\,\texttt{PEN}(x) \tag{2.3}$$

over the space of functions x for which some roughness quantification $\texttt{PEN}(x)$ is defined. A natural measure of the function roughness is the L^2 norm of its second derivative $\ddot{x}$ over the domain of interest I

$$\texttt{PEN}_2(x) = \|\ddot{x}\|^2 = \int_I [\ddot{x}(s)]^2 ds.$$

In equation (2.3), δ is a smoothing parameter that represents the rate of exchange between fit to the data and roughness of the function x. It is clear that, as $\delta \to \infty$, the curve x tends to the standard linear regression, while for $\delta \to 0$ it interpolates data at points $\mathbf{t}$. In other words, the spline smoothing method estimates a curve x from observations $\mathbf{y}$ by making explicit two conflicting goals in curve estimation. On the one hand, we wish to ensure that the estimated curve gives a good fit to the data, for example in terms of the residual sum of squares $\sum[y_j - x(t_j)]^2$. On the other hand, we do not wish the fit to be too good if this results in a curve x that is excessively locally variable. These competing aims are usually known as the trade off between bias and sampling variance, expressed by the following equation:

$$\mathrm{MSE}[\hat{x}(t)] = \mathrm{Bias}^2[\hat{x}(t)] + \mathrm{Var}[\hat{x}(t)].$$

where the Mean Squared Error (MSE) is originally defined as

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

This quantity is also referred to as the L^2 loss function and usually it cannot be used in this form, since there is no way of knowing $x(t)$ without using the data. Instead the bias in estimating $x(t)$ is defined as

$$\mathrm{Bias}[\hat{x}(t)] = x(t) - \mathbb{E}[\hat{x}(t)],$$

and the variance of estimate is

$$\mathrm{Var}[\hat{x}(t)] = \mathbb{E}[(\hat{x}(t) - \mathbb{E}[\hat{x}(t)])^2].$$

In spline smoothing, the MSE is one way of capturing the quality of estimate. A completely unbiased estimate of the function value $x(t_j)$ can be produced by a curve fitting y_j exactly, since this observed value is itself an unbiased estimate of $x(t_j)$. But any such curve must have high variance, manifested in the rapid local variation of the curve. MSE can often be dramatically reduced by sacrificing some bias in order to reduce sampling variance, and this is a key reason for imposing smoothness on the estimated curve. By requiring that the estimate vary only gently from one value to another, we are effectively “borrowing information” from neighbouring data values, thereby expressing our faith in the regularity of the underlying function x that we are trying to estimate. This pooling of information is what makes our estimated curve more stable, at the cost of some increase in bias. The roughness penalty makes explicit what we sacrifice in bias to achieve an improvement MSE or some other loss function. In De Boor (2002) it is proved that the curve x that minimises the objective functional $\mathtt{PENSSE}_\delta(x|\mathbf{y})$ is a cubic spline with breakpoints at the data points t_j. The cubic splines can be obtained by using order four B-spline basis function expansion. Placing breakpoints at $\mathbf{t}$ allows to adapt to the unequal spacing of sampling points and thus automatically takes advantage of high sampling density regions.

Lastly, in order to choose the penalisation parameter δ, Craven and Wahba (1978) developed the Generalised Cross-Validation measure (GCV). This method avoids the need to re-smooth data many times required by the classic cross-validation and reduces the tendency to overfitting. Given p pairs of a single curve (t_j, y_j), the optimal penalisation parameter $\hat{\delta}$ minimises the GCV function, i.e.

$$\hat{\delta} = \underset{\delta \in (0,+\infty)}{\arg\min}\, \mathtt{GCV}(\delta) = \underset{\delta \in (0,+\infty)}{\arg\min} \left(\frac{p}{p - \mathrm{df}(\delta)} \right) \left(\frac{\mathtt{SSE}}{p - \mathrm{df}(\delta)} \right)$$

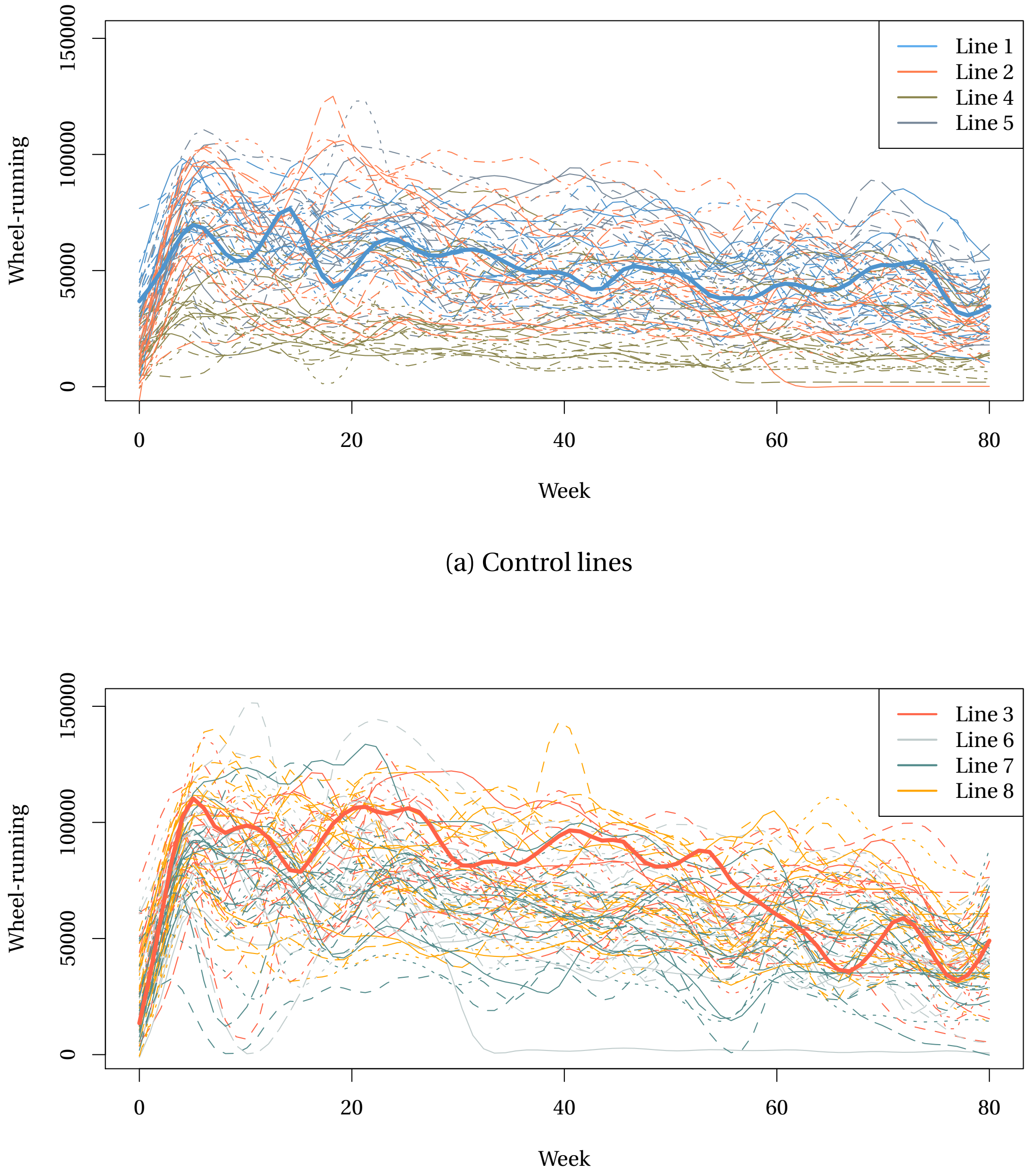
where df(δ) is the number of degrees of freedom of the spline.

This method is implemented in the function `spline.smooth()` of the R package `stats` (R Core Team, 2016). Applying it to the evolutionary dataset, we obtain the results presented in Figure 2.1. The breakpoints correspond to the times of data collection. We have assigned null weight to all the weeks when the number of revolutions has not been recorded correctly and equal to one for all the others.

2.3.2 Alignment

Once that the observations are in functional form, there is still another problem that has to be tackled before starting the actual data analysis. In fact, variation in functional observations involves both phase and amplitude and, while most approaches to FDA ignore this type of decomposition, it has become clear from a range of important real data analysis contexts that such ignorance can entail very substantial loss in statistical efficiency and interpretation. Heuristically, we can think of phase variability as the one that can be eliminated by suitably aligning the curves, while amplitude variability is the remaining variability among the curves once they have been aligned. This step of data pre-processing is called data *registration* or *alignment* and it usually involves transformations of the argument t rather than the values $x(t)$. Many methods exist for one dimensional curve registration and a summary of a detailed comparison of a number of major methods can be found in Marron et al. (2014).

Let $\mathcal{H}$ be some set of curves $x : I \subseteq \mathbb{R} \to \mathbb{R}$. Given two curves $x_1, x_2 \in \mathcal{H}$, the objective of the alignment procedure is to find a warping function $h(t) : \mathbb{R} \to \mathbb{R}$ that makes the registered curves $x_1 \circ h$ and x_2 as similar as possible. Similarity between functions is measured by some distance $d(\cdot,\cdot) : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$. The warping functions belong to a class h of functions such that for all $x \in \mathcal{H}$ and $h \in \mathcal{W}$ it holds $x \circ h \in \mathcal{H}$. In other words, we want to find $\bar{h} \in \mathcal{W}$ that minimises $d(x_1 \circ h, x_2)$. In this way it is possible separate amplitude and phase variability such that the



(a) Control lines

(b) Selected lines

Figure 2.1: Evolutionary biology data set after smoothing.

former is captured by the optimal warping function $\bar{h}$ and the former corresponds to the variability between the aligned functions.

The dataset analysed in this work has been aligned via the elastic analysis proposed in Tucker et al. (2013) and implemented in the R package `fdasrvf` (Tucker, 2016). This method has the advantage of allowing great flexibility in the choice of the warping functions and considering the same domain for all the functions.

In order to easily describe the method, we restrict the set of considered functions to those that are absolutely continuous on $I = [0,1]$. Now, let $\mathcal{I}$ denote the set of all such functions. In practice, since the observed data are discrete, this assumption is not restrictive. Also, the set of warping functions is defined as the set of all boundary-preserving diffeomorphisms over I:

$$\mathcal{W} = \{h : I \to I | h(0) = 0, h(1) = 1, h \text{ is a diffeomorphism}\}.$$

With the composition operation, the set $\mathcal{W}$ is a group with the identity element $h_I(t) = t$ for all $t \in I$. Then, the Square-Root Slope Function (SRSF) of x is defined as

$$g(t) = \text{sign}(\dot{x}(t))\sqrt{|\dot{x}(t)|}.$$

It can be shown that if the function x is absolutely continuous, then the resulting SRSF is square-integrable (see Robinson, 2012). For every $g \in L^2$ and a fixed $t \in I$, the function x can be obtained using the equation

$$x(t) = x(0) + \int_0^t g(s)|g(s)|ds,$$

since $g(s)|g(s)| = \dot{x}(s)$. Therefore, the mapping from $\mathcal{I}$ to $L^2 \times \mathbb{R}$ given by $x \mapsto (g, f(0))$ is a bijection (Robinson, 2012). If we warp a function x by h, the SRSF of $x \circ h$ is given by $\tilde{g}(t) = (g,h)(t) = g(h(t))\sqrt{\dot{h}(t)}$. Thanks to this, it can be shown that for any $x_1, x_2 \in \mathcal{I}$ and $h \in \mathcal{W}$,

$$\|g_1 - g_2\| = \|(x_1, h) - (x_2, h)\|,$$

where g_1, g_2 are SRSFs of x_1, x_2 respectively. This isometric property offered a new cost term for pairwise registration of functions:

$$\inf_{h \in \mathcal{W}} \|x_1 - (x_2, h)\|.$$

This equation suggests that one can align the SRSFs of any two functions and then map them back to $\mathcal{I}$ to obtain registered functions. The advantage of using this quantity is that it forms a proper distance on the quotient space $L^2/\mathcal{W}$. This allows to redefine amplitude as follows:

Definition 2.12. *For any two functions $x_1, x_2 \in \mathcal{W}$ and the corresponding SRSFs $g_1, g_2 \in L^2$, the* amplitude *or y-*distance *is*

$$d_y(x_1, x_2) = \inf_{h \in \mathcal{W}} \|x_1 - (x_2 \circ h)\sqrt{\dot{h}}\|.$$

It can be shown that, for any $h_1, h_2 \in \mathcal{W}$, it holds $d_y(x_1 \circ h_1, x_2 \circ h_2) = d_y(x_1, x_2)$. Define also the Karcher mean of the given function as a local minimum of the following cost function:

$$\mu_f = \operatorname*{argmin}_{x \in \mathcal{I}} \sum_{i=1}^{n} d_y(f, f_i)^2 \quad \text{or, equivalently,} \quad \mu_q = \operatorname*{argmin}_{g \in L^2} \sum_{i=1}^{n} \Big(\inf_{h_i \in \mathcal{W}} \|g - (g_i, h_i)\|^2 \Big).$$

Then, the phase-amplitude separation can be performed as stated in Algorithm 2.1.

Algorithm 2.1 (Phase-amplitude separation)**.**

1. *Compute the SRSFs $g_1, \dots, g_n$ of the given $x_1, \dots, x_n$ and select $\mu = q_i$ where*

$$i = \operatorname*{argmin}_{1 \le i \le n} \left\| g_i - \frac{1}{n} \sum_{j=1}^{n} g_j \right\|;$$

2. *For each q_i find the $\bar{h}_i$ such that*

$$\bar{h}_i = \operatorname*{argmin}_{h \in \mathcal{W}} \Big(\Big\| \mu - (g_i \circ h)\sqrt{\dot{h}} \Big\| \Big).$$

   The solution to this optimisation problem comes from the dynamic programming algorithm.

3. *Compute the aligned SRSFs using $\tilde{g}_i \mapsto (g_i \circ \bar{h}_i)\sqrt{\dot{\bar{h}}_i}$.*

4. *If the increment $\|n^{-1}\sum_{i=1}^{n} - \mu\|$ is "small", then continue. Else, update the mean using $\mu \mapsto n^{-1}\sum_{i=1}^{n} \tilde{g}_i$ and return to step 2.*

5. *The function μ represents a whole equivalence class of solutions and now we select the preferred element μ_q of that orbit:*

   (a) *Compute the mean h_μ of all $\{\bar{h}_i\}$ (an ad hoc algorithm can be found in Tucker et al., 2013). Then compute*

$$\mu_q = (\mu \circ h_\mu^{-1})\sqrt{\dot{h}_\mu^{-1}}.$$

   (b) *Update $\bar{h}_i \mapsto \bar{h}_i \circ h_\mu^{-1}$. Then compute the aligned SRSFs using $\tilde{g}_i \mapsto (g_i \circ \bar{h}_i)\sqrt{\dot{\bar{h}}_i}$.*

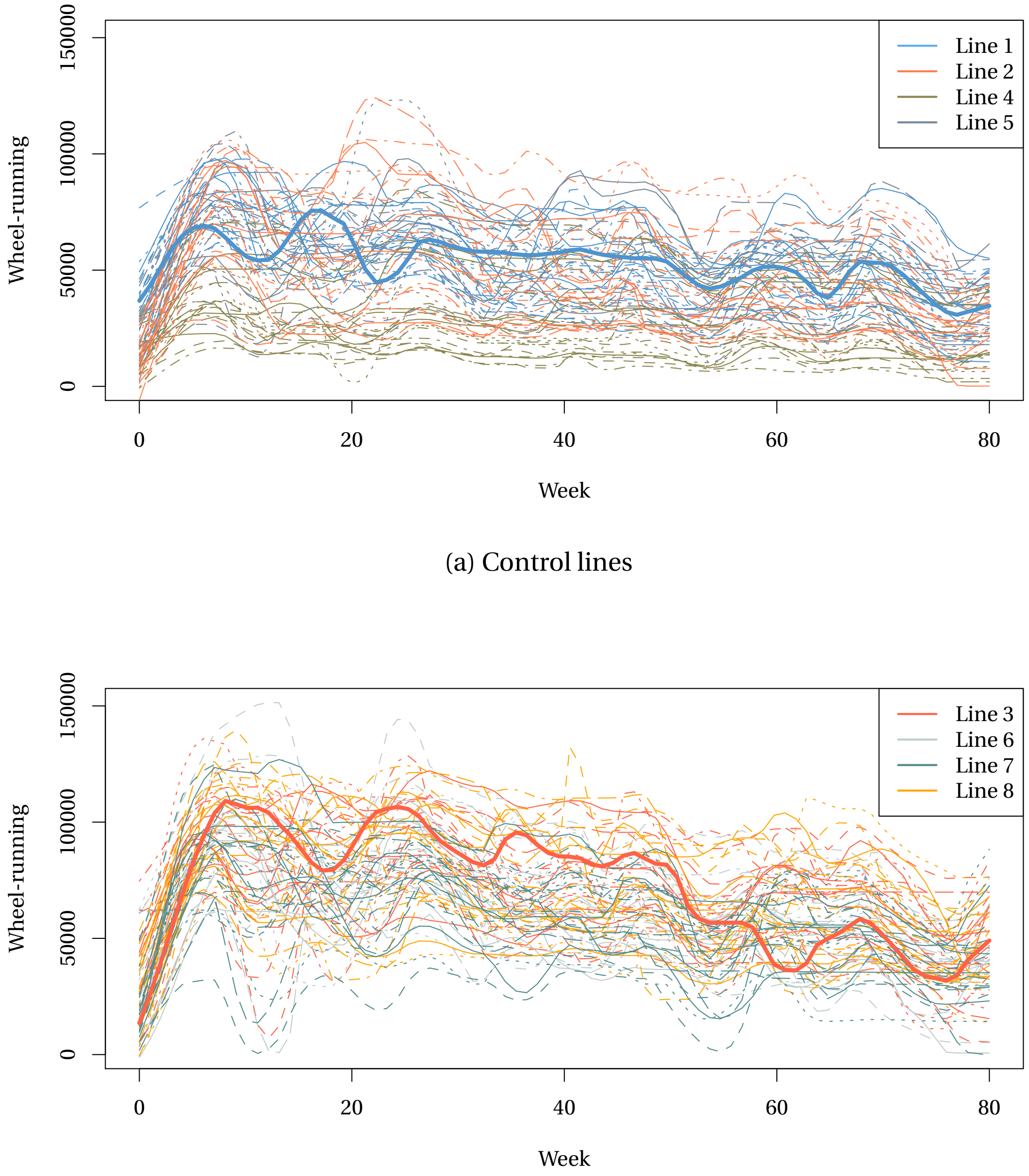
This procedure results in three items:

- μ_q, the preferred element of the Karcher mean class $\{(\mu_q, h) | h \in \mathcal{W}\}$;
- $\{\tilde{g}_i\}$, the set of aligned SRSFs;
- $\{\bar{h}_i\}$, the set of optimal warping functions.

From the aligned SRSFs, one can compute individual aligned functions using

$$\tilde{x}_i(t) = x_i(0) + \int_0^t \tilde{g}_i(s)|\tilde{g}_i(s)|ds.$$

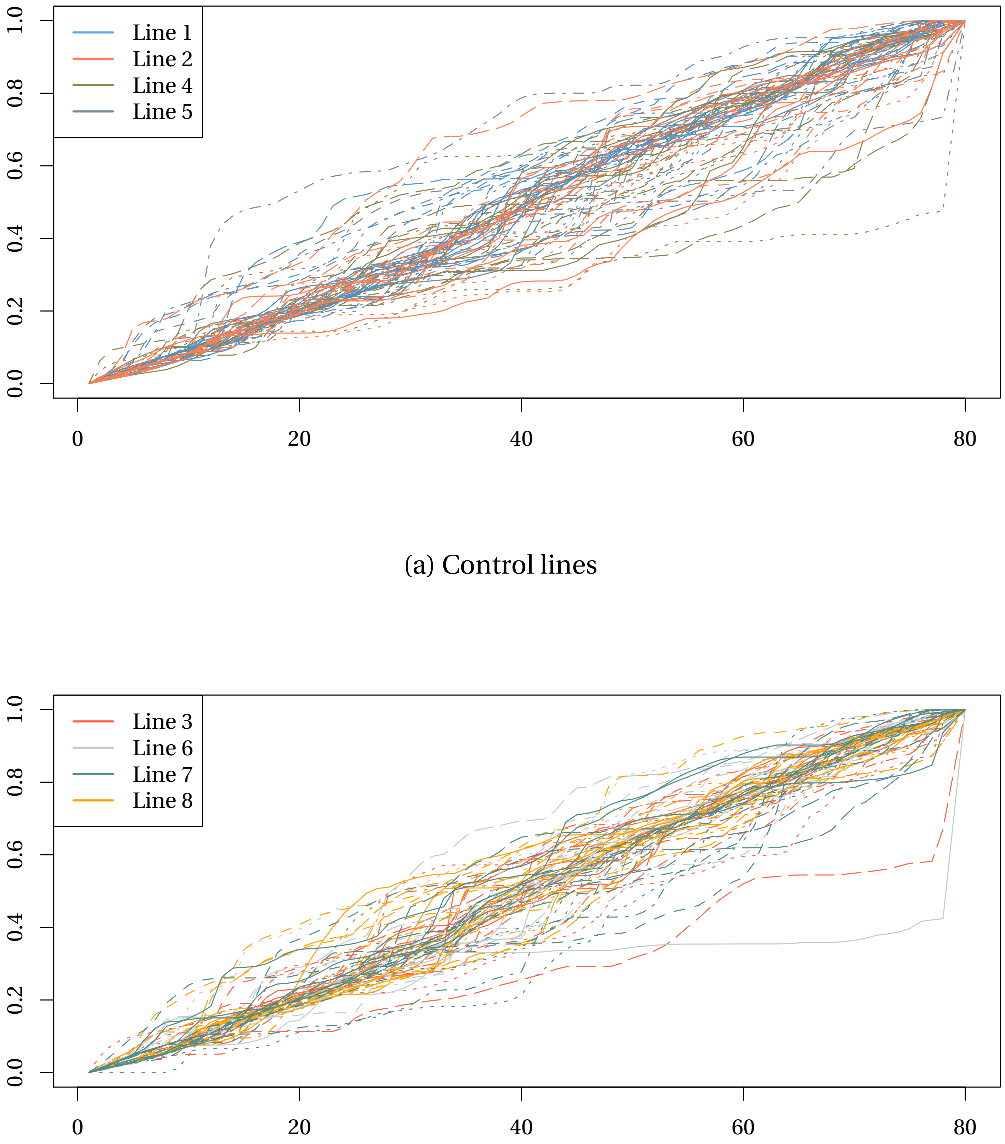
In Figure 2.2 are reported the aligned data and in Figure 2.3 the corresponding warping functions.



(a) Control lines

(b) Selected lines

Figure 2.2: Evolutionary biology data set after alignment.



(a) Control lines

(b) Selected lines

Figure 2.3: Evolutionary biology data set, warping functions.

Chapter 3

Permutation tests

It has already been mentioned that the main contribution of this thesis is the introduction of a new methodology that allows to test for the equality of covariance operators. Such method relies on the permutation approach. For this reason, this chapter is devoted to the exposition of this branch of Inferential Statistics and the presentation of the techniques used in the following.

The theory of permutation tests has evolved from the works of Ronald Fisher and Edwin J. G. Pitman in the 1930s: it was the first resampling technique ever used. After that, thanks to the recent computational advances, many other resampling techniques have been proposed, such as jackknife and delta methods (Quenouille, 1949). Another renowned resampling method is bootstrap, introduced by Efron (1979) and inspired by earlier works on the jackknife. The main benefit of using resampling methods is that larger classes of statistical problems may be analysed. Diaconis and Efron (1983) wrote:

> *The new [resampling] methods free the statistician to attack more complicated problems, exploiting a wider array of statistical tools.*

The basic idea of permutation tests is to obtain the distribution of the test statistic under the null hypothesis by calculating all possible values of the test statistic under rearrangements of the labels on the observed data points. In other words, the method by which treatments are allocated to subjects in an experimental design is mirrored in the analysis of that design: if labels are exchangeable under the null hypothesis, the resulting tests yield exact significance levels.

3.1 Theoretical foundations

In this section we introduce the main terminology, definitions and general theory of permutation tests for the case of one-dimensional problems, as presented in Pesarin and Salmaso (2010). Particular emphasis is given to the two-sample design, taken as a guide. The analyses for multi-aspect problems are obtained by the non-parametric combination methodology, which is presented in Section 3.2.

Let us assume that a one-dimensional non-degenerate variable X takes values on the sample space $\mathcal{X}$ and that associated with $(X,\mathcal{X})$ are parent distributions P belonging to a non-parametric[1] family $\mathcal{P}$. Each P gives the probability measure to events A belonging to a suitable σ-algebra $\mathcal{A}$ of events. The family $\mathcal{P}$ may consist of distributions of either quantitative (continuous, discrete or mixed) or categorical (nominal or ordered) variables. In the present work, we consider only the first case. The notation $(X,\mathcal{X},\mathcal{A},P),\ P\in\mathcal{P}$ summarises the statistical model associated with the problem at hand. In the following, $\mathcal{X}^n$ denotes the usual product space, while $\mathcal{A}^{(n)}$ and $P^{(n)}$ are the product σ-algebra and measure on $\mathcal{X}^n$ respectively.

Considering the two-sample design, let $\mathbf{X}_j=\{X_{ij}, i=1,\dots,n_j\}\in\mathcal{X}^{n_j}$ be the i.i.d. sample data of size n_j drawn from the model $(X,\mathcal{X},\mathcal{A},P_j)$, $P_j\in\mathcal{P},\ j=1,2$, respectively. The whole data set can be indicated as follows:

$$\mathbf{X}=(\mathbf{X}_1,\mathbf{X}_2)=\{X_i,\ i=1,\dots,n;\ n=n_1+n_2\}$$

where it is intended that the first n_1 data belong to the first sample and the rest to the second. In practice, with $\mathbf{u}^*=(u_1^*,\dots,u_n^*)$ denoting a permutation of unit labels $\mathbf{u}=(1,\dots,n)$,

$$\mathbf{X}^*=(\mathbf{X}_1^*,\mathbf{X}_2^*)=\{X_i^*=X_{u_i^*},\ i=1,\dots,n;\ n=n_1+n_2\}$$

is the related permutation of $\mathbf{X}$, where $\mathbf{X}_1^*$ and $\mathbf{X}_2^*$ are the two permuted samples. Suppose that we want to test for stochastic dominance. In particular, the alternative assumes that treatments produce effects Δ_1 and Δ_2 respectively and that $\Delta_1\overset{d}{>}\Delta_2$, where $\overset{d}{>}$ stands for stochastic dominance (i.e. the cumulative distribution functions are such that $F_1(x)\le F_2(x),\ \forall x\in\mathbb{R},\ F_1\not\equiv F_2$). Without loss of generality, we assume that effects in H_1 are such that $\Delta_1=\Delta>0$ and $\text{prob}(\Delta_2=0)=1$. This can be seen for example as an experiment that consists in assigning an "active treatment" only to to subjects in the first group and a "placebo" to the others. Thus, the hypotheses are

$$H_0:\{\Delta=0\}=\{P_1=P_2\}\quad\text{against}\quad H_1:\{\Delta>0\}.$$

[1] Of course $\mathcal{P}$ may belong to any parametric family. However, in such cases, there are often parametric counterparts that may perform optimally.

First of all, we have to define the conditional reference space $\mathcal{X}^n_{/\mathbf{X}}$ associated with $\mathbf{X}$ under the assumption that H_0 is true.

Definition 3.1. *Given a data set* $\mathbf{X}$ *of size n and a model* $(\mathbf{X}, \mathcal{X}^n, \mathcal{A}^{(n)}, P^{(n)})$, $P^{(n)} \in \mathcal{P}^{(n)}$, *the* conditional reference space *is the set of points of the sample space* $\mathcal{X}^n_{/\mathbf{X}}$ *which are equivalent to* $\mathbf{X}$ *in terms of information carried by the associated underlying likelihood. It is indicated by the symbol* $\mathcal{X}^n_{/\mathbf{X}}$.

Thus, $\mathcal{X}^n_{/\mathbf{X}}$ contains all points $\mathbf{X}^*$ such that the likelihood ratio $dP^{(n)}(\mathbf{X})/dP^{(n)}(\mathbf{X}^*)$ is independent of P: it corresponds to the *orbit* of equivalent points associated to $\mathbf{X}$. From now on it will be indicated as $\mathcal{X}_{/\mathbf{X}}$ for brevity. Since for finite sample sizes the number M of points in $\mathcal{X}_{/\mathbf{X}}$ is finite and, in particular,

$$M = \sum_{\mathcal{X}_{/\mathbf{X}}} \mathbb{1}\left[\mathbf{X}^* \in \mathcal{X}_{/\mathbf{X}}\right] < \infty,$$

the permutation conditional probability $\mathbb{P}$ of every $A \in \mathcal{A}^{(n)}$ is defined and calculated as

$$\mathbb{P}(\mathbf{X}^* \in A|\mathcal{X}_{/\mathbf{X}}) = \frac{\sum_{\mathcal{X}_{/\mathbf{X}}} \mathbb{1}\left[\mathbf{X}^* \in A\right]}{M}.$$

Therefore, the restriction of the collection of events $\mathcal{A}$ over the permutation sample space $\mathcal{X}_{/\mathbf{X}}$, that is

$$\mathcal{A}^{(n)} \cap \mathcal{X}_{/\mathbf{X}} = \mathcal{A}^{(n)}_{/\mathbf{X}},$$

consisting of conditional events given $\mathbf{X}$, defines the *permutation measurable space* $(\mathcal{X}_{/\mathbf{X}}, \mathcal{A}^{(n)}_{/\mathbf{X}})$ on which the permutation probability $\mathbb{P}(A|\mathcal{X}_{/\mathbf{X}})$ is defined. As a consequence, the following proposition holds:

Proposition 3.1. *In* $H_0 : \{X_1 \overset{d}{=} X_2\} = \{P_1 = P_2\}$, *provided that in* $\mathcal{X}_{/\mathbf{X}}$ *there are no multiple points, that is*

$$\sum_{\mathcal{X}_{/\mathbf{X}}} \mathbb{1}\left[\mathbf{X}^* = \mathbf{x}\right] = \begin{cases} 1 & \textit{if } \mathbf{x} \in \mathcal{X}_{/\mathbf{X}} \\ 0 & \textit{otherwise} \end{cases}$$

permutations $\mathbf{X}^*$ *are equally likely:*

$$\mathbb{P}(\mathbf{X}^* = \mathbf{x}|\mathcal{X}_{/\mathbf{X}}) = \begin{cases} \frac{1}{M} & \textit{if } \mathbf{x} \in \mathcal{X}_{/\mathbf{X}} \\ 0 & \textit{otherwise.} \end{cases}$$

In other words, the permutations $\mathbf{X}^*$ *of the dataset are uniformly distributed over* $\mathcal{X}_{/\mathbf{X}}$ *conditionally on* $\mathbf{X}$.

This property shows that the null permutation distribution $\mathbb{P}(\mathbf{X}^* = \mathbf{x} \,|\mathcal{X}_{/\mathbf{X}})$ only depends on the data set $\mathbf{X}$. In H_1, where it is assumed that

$$\exists A \in \mathcal{A}^{(n)} \quad \text{s.t.} \quad P_1(A) \neq P_2(A),\ P_1(A) > 0,\ P_2(A) > 0$$

a set of sufficient statistics is the pair $(\mathbf{X}_1, \mathbf{X}_2)$. Consequently, the data are exchangeable within but not between the samples, and so the data set permutations are not uniformly distributed over $\mathcal{X}_{/\mathbf{X}}$ conditionally. Hence, if we are able to find statistics sensitive to such a non-uniform distribution, then we are able to construct permutation tests. However, the problem of establishing the best test when P is unknown remains open (Lehmann and Romano, 2005). A test statistic is a non-degenerate measurable function T, mapping $\mathcal{X}^n$ into $\mathbb{R}$, which satisfies properties suitable for inference. Suppose that $T : \mathcal{X}^n \to \mathbb{R}$ is such an appropriate test statistic for which, without loss of generality, we assume that large values are evidence against H_0.

Definition 3.2. *The* permutation support $\mathcal{T}_\mathbf{X}$ *induced by the pair* $(T, \mathbf{X})$ *is the set*

$$\mathcal{T}_\mathbf{X} = \{T^* = T(\mathbf{X}^*) : \mathbf{X}^* \in \mathcal{X}_{/\mathbf{X}}\}$$

containing all possible values assumed by T *as* $\mathbf{X}^*$ *varies in* $\mathcal{X}_{/\mathbf{X}}$.

Of course, when more than one aspect is of interest for the analysis, a test can be associated with a vector of statistics $\mathbf{T} = (T_1, \ldots, T_k) : \mathcal{X}^n \to \mathbb{R}^k$ where $k \geq 1$ is the (finite or countable) number of aspects under consideration. This kind of statistics is used for example in the theory of multidimensional permutation tests, developed in Section 3.2. In the two-sample case considered above, a test statistic could be the difference between the sample means:

$$T^* = m_1^* - m_2^* = \sum_{i=1}^{n_1} \frac{X_{1i}^*}{n_1} - \sum_{i=1}^{n_2} \frac{X_{2i}^*}{n_2}$$

Suppose now that H_0 is true. According to Proposition 3.1, $\mathbf{X}^*$ is uniformly distributed over $\mathcal{X}_{/\mathbf{X}}$ and put the M members of $\mathcal{T}_\mathbf{X}$ in non-decreasing order $T_{(1)}^* \leq T_{(2)}^* \leq \cdots \leq T_{(M)}^*$. For each value of $\alpha \in (0, 1)$,

$$T_{(M_\alpha)}^* = T_\alpha(\mathbf{X}) = T_\alpha$$

where $M_\alpha = M - \lceil \alpha M \rceil$ defines the *permutation critical value* associated with the pair $(T, \mathbf{X})$ and corresponds to $\sum_{\mathcal{X}_{/\mathbf{X}}} \mathbb{1}[T(\mathbf{X}^*) < T_\alpha]$, the number of permutation values T^* that are strictly less than T_α. We define $T^0 = T(\mathbf{X})$ and we reject the null hypothesis associated with the pair $(T, \mathbf{X})$ if $T^0 \geq T_\alpha$. Instead, if $T^0 < T_\alpha$, we say that there is no evidence to reject H_0.The attainable α-values belong to the set $\Lambda_\mathbf{X} = \{L_\mathbf{X}(t) : dL_\mathbf{X}(t) > 0\}$ of step points of the *significance level function* $L_\mathbf{X} = \mathbb{P}(T^* \geq t | \mathcal{X}_{/\mathbf{X}})$. $\Lambda_\mathbf{X}$ is always a discrete set, the elements of which depend on the pair $(T, \mathbf{X})$. Therefore, for permutation tests not all values of type I error rates are possible in practice. Then, if we wish to test a desired type I error rate of $\bar{\alpha}$ and choose $\sup \alpha \leq \bar{\alpha}$, with $\alpha \in \Lambda_{\mathbf{X}}$, the permutation tests become conservative. Of course, if the desired α-value $\bar{\alpha}$ belongs to $\Lambda_{\mathbf{X}}$, then $\alpha = \bar{\alpha}$.

3.1.1 p-value

It is often more convenient to refer to the p-value associated to $(T, \mathbf{X})$, instead of determining the critical value T_α. This is defined as

$$\lambda(\mathbf{X}) = L_{\mathbf{X}}(T^0) = \mathbb{P}(T^* \geq T^0 | \mathcal{X}_{/\mathbf{X}}),$$

and can be computed exactly by complete enumeration of $\mathcal{T}_{\mathbf{X}}$ or estimated, to the desired degree of accuracy, by a conditional Monte Carlo algorithm based on a random sampling from $\mathcal{X}_{/\mathbf{X}}$. Note that the p-value is a non-increasing function of T^0 and has a one-to-one relationship with the attainable α-values of the test, in the sense that $\lambda(\mathbf{X}) > \alpha$ implies $T^0 < T^\alpha$ and vice versa. Hence, we reject H_0 if $\lambda(\mathbf{X}) \leq \alpha$. It is important to notice that the attainable α-values play the role of critical values, in the sense that α is the exact critical value for $\lambda(\mathbf{X})$. In this sense, the p-value $\lambda(\mathbf{X})$ can be used as a test statistic; this fact will be used in Section 3.2.2. Moreover, in H_0 we have that $\mathbb{P}(\lambda(\mathbf{X}) | \mathcal{X}_{/\mathbf{X}}) = \alpha$ for every $\alpha \in \Lambda_{\mathbf{X}}$.

A Monte Carlo algorithm for estimating the p-value

The Monte Carlo (MC) algorithm for evaluating the p-value λ of a test statistic T on a data set $\mathbf{X}$, includes the following steps (see e.g. Pesarin and Salmaso, 2010):

Algorithm 3.1 (Monte Carlo approximation of permutation tests)**.**

1. *Calculate, on the given data set* $\mathbf{X}$*, the observed value* T^0 *of the test statistic* T*;*
2. *Take a random permutation* $\mathbf{X}^*$ *of* $\mathbf{X}$ *and calculate* $T^* = T(\mathbf{X}^*)$*;*
3. *Independently repeat step 2. B times;*
4. *The set* $\{\mathbf{X}^*_b,\ b = 1, \dots, B\}$ *is a random sample from the permutation sample space* $\mathcal{X}_{/\mathbf{X}}$ *and so the corresponding values* $\{T^*_b,\ b = 1, \dots, B\}$ *simulate the null permutation distribution of* T*. Therefore, the p-value is estimated as*

$$\hat{\lambda} = \sum_{1 \leq b \leq B} \frac{\mathbb{1}[T^*_b \geq T^0]}{B}$$

that is the proportion of permutation values greater or equal than the observed one.

The symbol ˆ indicates that the so obtained p-value is a Monte Carlo approximation.

3.2 Non-parametric combination methodology

In this section we present a natural extension of permutation testing to a vast range of multi-aspect problems. In particular, we introduce a method for Non-Parametric Combination (NPC) of a finite number of dependent permutation tests, proposed in Pesarin and Salmaso (2010). Indeed, when many response variables are involved or many different aspects are of interest, it is convenient to process data using a finite set of $k > 1$ different partial tests, that may be useful for marginal inference and, considering them together, to extract information on an overall hypothesis.

3.2.1 Main assumptions

Here we introduce the notation and main assumptions regarding the data structure, the set of partial tests and the hypotheses being tested in NPC contexts. In this case, without loss of generality, we refer to a one-way Analysis of Variance (ANOVA) design. Consider a data set

$$\mathbf{X} = \{X_{ij},\ i = 1,\dots,q,\ j = 1,\dots,n_i\}.$$

that consists of $q \geq 2$ samples or groups of size $n_i \geq 2$, with $n = \sum_i n_i$. The groups are related to q levels of a treatment and the data X_{ij} are supposed to be independent and identically distributed with distributions $P_i \in \mathcal{P}$, $i = 1,\dots,q$. The null hypothesis refers to equality of distributions of responses on q groups:

$$H_0 : \{P_1 = \cdots = P_q\} = \{X_1 \overset{d}{=} \dots \overset{d}{=} X_q\}.$$

Let us suppose that H_0 may be properly and equivalently broken down into a finite set of sub-hypotheses H_{0i}, $i = 1,\dots,k$, each appropriate for a partial aspect of interest. In the case of a one-way ANOVA, this corresponds to

$$H_{0i} : \{P_h = P_j\} = \{X_h \overset{d}{=} X_j\}, \quad \text{for some } h, j = 1,\dots,q, \quad h \neq j.$$

Therefore, H_0 is true if all the H_{0i} are jointly true:

$$H_0 : \left\{ \bigcap_{i=1}^{k} H_{0i} \right\}$$

H_0 is called the *global* (or overall) *null hypothesis* and implies that data are exchangeable with respect to the groups. The alternative hypothesis states that at least one of the null sub-hypotheses H_{0i} is not true. Hence, the alternative may be represented by the union of k sub-alternatives,

$$H_1 : \left\{ \bigcup_{i=1}^{k} H_{1i} \right\}$$

$\mathbf{T} = \mathbf{T}(X)$ represents a k-dimensional vector of test statistics, in which the ith component $T_i = T_i(X)$, $i = 1, \dots, k$, represents the non-degenerate ith partial test which is assumed to be appropriate for testing sub-hypothesis H_{0i} against H_{1i}. We also have to specify the assumptions regarding the set of partial tests $\mathbf{T} = \{T_i,\ i = 1, \dots, k\}$ which are needed for NPC. In particular,

Assumption 3.1. *All permutations partial tests T_i are marginally unbiased and significant for large values, so that they are stochastically larger in H_1 than in H_0.*

This implies that

$$\mathbb{P}(T_i \geq T_{i\alpha} | \mathcal{X}_{/\mathbf{X}}, H_{1i}) \geq \alpha, \forall \alpha > 0, i = 1, \dots, k,$$

and

$$\begin{aligned} \mathbb{P}(T_i \leq z | \mathcal{X}_{/\mathbf{X}}, H_{0i}) &= \mathbb{P}(T_i \leq z | \mathcal{X}_{/\mathbf{X}}, H_{0i} \cap \check{H}_i) \\ &\geq \mathbb{P}(T_i \leq z | \mathcal{X}_{/\mathbf{X}}, H_{1i}) \\ &= \mathbb{P}(T_i \leq z | \mathcal{X}_{/\mathbf{X}}, H_{1i} \cap \check{H}_i), \quad \forall z \in \mathbb{R}, i = 1, \dots, k, \end{aligned}$$

where irrelevance with respect to the complementary set of hypotheses

$$\check{H}_i : \bigcup_{i \neq j} (H_{0j} \cup H_{1j})$$

means that it does not matter which among H_{0j} and H_{1j}, $j \neq i$, is true when testing for the ith sub-hypothesis.

Assumption 3.2. *All permutation partial tests T_i are marginally consistent, i.e. as the sample sizes tend to infinity*

$$\mathbb{P}(T_i \geq T_{i\alpha} | H_{1i}) \rightarrow 1,\ \forall \alpha \in (0, 1)$$

where $T_{i\alpha}$, which is assumed to be finite, is the critical value of T_i at level α.

3.2.2 Combining functions

In order to perform the test, we also have to define function $\Psi : \mathbb{R}^k \to \mathbb{R}$ to combine the partial test statistics or p-values. For the sake of simplicity and uniformity of analysis, but without loss of generality, we only refer to combining functions applied to p-values associated with partial tests. The application to the other cases is straightforward. All combining functions Ψ must satisfy at least the following reasonable properties.

Property 3.1. *Ψ must be non-decreasing in each argument: $\Psi(\ldots,\lambda_i,\ldots) \geq \Psi(\ldots,\lambda'_i,\ldots)$ if $\lambda_i \leq \lambda'_i$, $i \in \{1,\ldots,k\}$.*

Property 3.2. *Every combining function Ψ must attain its supremum value $\bar{\Psi}$, possibly not finite, even when only one argument attains zero: $\Psi(\ldots,\lambda_i,\ldots) \to \bar{\Psi}$ if $\lambda_i \to 0, i = 1,\ldots,k$.*

Property 3.3. *For all $\alpha > 0$, the critical value T^{α}_{Ψ} of every Ψ is assumed to be finite and strictly smaller than $\bar{\Psi}$.*

In the following, we will indicate by $\mathcal{C}$ the class of combining functions that satisfy Properties 3.1, 3.2 and 3.3. Also, it is generally desirable for Ψ to be symmetric i.e. invariant with respect to rearrangements of the input arguments: $\Psi(\lambda_{u_i},\ldots,\lambda_{u_k})$ where $(u_i,\ldots,u_k)$ is any permutation of $(1,\ldots,k)$. Below we list some convex combining functions with appealing power and consistency properties. We assume that, under the null hypothesis, the p-values are uniformly distributed on $[0,1]$.

Fisher omnibus function

$$T_F = -2\sum_i \log(\lambda_i)$$

This is a good default choice: it has greatest statistical power when most but not all the partial alternative hypotheses are true. If the partial test statistics are independent and continuous, then under the null hypothesis T_F follows a central χ^2 distribution with $2k$ degrees of freedom.

Lipták function

$$T_L = \sum_i \Phi^{-1}(1-\lambda_i)$$

It is used when one has confidence that under the alternative hypothesis every sub-alternative will be true. If the partial tests are independent and continuous, under the null hypothesis T_L is normally distributed with 0 mean and variance k (Lipták, 1958).

Tippett function

$$T_T = \max_{1 \le i \le k} (1 - \lambda_i)$$

It has optimal power when only one sub-alternative is expected to be true under the alternative hypothesis. As explained in Chapter 5, if interested in post-hoc comparisons, this combining function enables computation to be speeded up. Its null distribution, if the partial tests are independent and continuous, behaves according to the largest of k random values uniformly distributed on $(0, 1)$. For dependent partial tests it allows for bounds on the rejection probability according to the Bonferroni inequality.

Direct combination

$$T_D = \sum_i T_i$$

This function allows us to avoid the quite intensive calculations for the computation of the permutational distributions of the partial p-values (see Algorithm 3.3, step 3.). However, its use requires some additional assumptions: all partial test statistics must be homogeneous and their asymptotic support must be at least unbounded on the right. Under the null hypothesis, if the partial test statistics are independent and continuous, this is a sum of n independent and identically distributed random variables.

Figure 3.1 shows the critical regions for these combination functions when the considered number of partial tests is equal to 2, the level of the test is $\alpha = 0.1$ and the partial tests are independent and continuous.

3.2.3 Algorithm

This section deals with a two-phase algorithm used to obtain a conditional Monte Carlo estimate of the permutation distribution of combined tests. In this setting, simulations from the permutation sample space $\mathcal{X}_{/\mathbf{X}}$ by a Monte Carlo method are carried out in analogy with the Algorithm 5.1 discussed for univariate problems. The first phase of the algorithm is dedicated to the estimation of the k-variate distribution of the vector of test statistics $\mathbf{T}$:

Algorithm 3.2 (Non-parametric combination methodology, phase I)**.**

1. *Calculate the vector of observed values of tests:* $\mathbf{T}^0 = \mathbf{T}(\mathbf{X})$*;*
2. *Consider a random permutation* $\mathbf{X}^* \in \mathcal{X}_{/\mathbf{X}}$ *of* $\mathbf{X}$*, where* $\mathcal{X}_{/\mathbf{X}} = \{\bigcup_{\mathbf{u}^*} [X(u_i^*), i = 1, \ldots, \sum_i n_i]\}$ *in which* $\mathbf{u}^*$ *is a permutation of unit labels and compute* $\mathbf{T}^* = \mathbf{T}(\mathbf{X}^*)$*;*

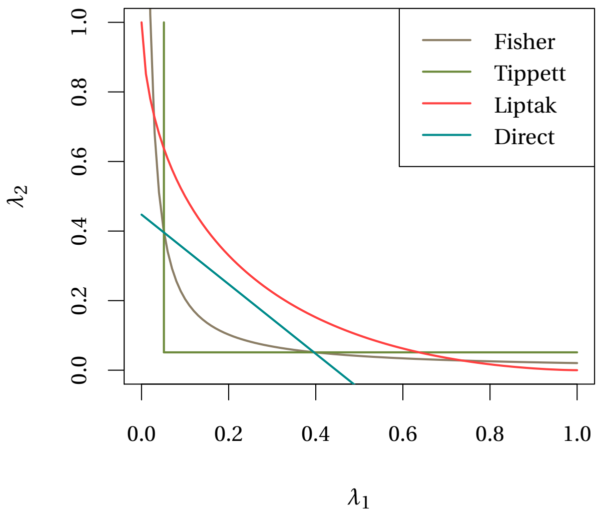


Figure 3.1: Critical regions of the combined tests for the case $k = 2$, $\alpha = 0.1$, independent and continuous partial tests and the Fisher, Tippett, Liptak and direct combination functions.

3. *Carry out B independent repetitions of the previous step: $\{\boldsymbol{T}^{(b)}\}_{b=1}^{B}$ is a random sampling from the permutation k-variate distribution of $\boldsymbol{T}$, $b = 1, \dots, B$;*

4. *A consistent estimate of the cumulative distribution function $F(\mathbf{t}|\mathcal{X}_{/\mathbf{X}})$ is*

$$\hat{F}(\boldsymbol{t}|\mathcal{X}_{/\mathbf{X}}) = \frac{\sum_b \mathbb{1}\left[\mathbf{T}^{(b)} \le \mathbf{t}\right]}{B};$$

5. *A consistent estimate of $\lambda_i = \mathbb{P}\{T_i^* \ge t|\mathcal{X}_{/\mathbf{X}}\}$ is*

$$\hat{\lambda}_i(t|\mathcal{X}_{/\mathbf{X}}) = \frac{\sum_b \mathbb{1}\left[T_i^{(b)} \ge t\right]}{B}.$$

Note that, with respect to traditional empirical distribution function estimators, 1/2 and 1 can be added respectively to the numerators and denominators of equations in steps 4 and 5. This is done in order to obtain estimated values of the cumulative distributions function and of the significance distribution functions compatible with the combining functions mentioned in Section 3.2.2. The following result is stated in Pesarin and Salmaso (2010):

Theorem 3.1. *As B tends to infinity, $\forall \mathbf{t} \in \mathbb{R}^k$, $\hat{F}(\mathbf{t}|\mathcal{X}_{/\mathbf{X}})$ and $\hat{L}_i(\mathbf{t}|\mathcal{X}_{/\mathbf{X}})$ almost surely converge to the permutation cumulative distribution function $F(\mathbf{t}|\mathcal{X}_{/\mathbf{X}})$ and the permutation significance level function $L(\mathbf{t}|\mathcal{X}_{/\mathbf{X}})$, respectively.*

The second phase of the algorithm consists in estimating the global p-value of the test:

Algorithm 3.3 (Non-parametric combination methodology, phase II)**.**

1. *The k observed p-values are estimated by $\hat{\lambda}_i = \hat{\lambda}_i(T_i^0|\mathcal{X}_{/\mathbf{X}})$;*
2. *The combined observed value of the test is $T_\Psi^0 = \Psi(\hat{\lambda}_1, \dots, \hat{\lambda}_k)$;*
3. *The* b*th combined value is*

$$T_\Psi^{(b)} = \Psi(\hat{\lambda}_1^{(b)}, \dots, \hat{\lambda}_k^{(b)}), \;\; \hat{\lambda}_i^{(b)} = \hat{\lambda}_i(T_i^{(b)}|\mathcal{X}_{/\mathbf{X}}) \;\; i = 1, \dots, k, \; b = 1, \dots, B;$$

4. *The p-value of the combined test T is estimated as*

$$\hat{\lambda}_\Psi = \frac{\sum_b \mathbb{1}[T_\Psi^{(b)} \geq T_\Psi^0]}{B};$$

5. *If $\hat{\lambda}_\Psi \leq \alpha$, H_0 is rejected.*

According to Theorem 3.1, as by assumption k is a fixed finite integer and Ψ is continuous, when B tends to infinity, the combined empirical distribution function of T_Ψ tends to the actual cumulative distribution function with probability one. The same holds for the approximate combined p-value.

Chapter 4

State of the art of equality tests for covariance operators

Let us recall the problem of interest: consider a multiple sample dataset $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_q)$, where

$$x_{i1}, \ldots, x_{in_i} \in L^2(I), \quad i = 1, \ldots, q,$$

are realisations of a random process with mean function μ and covariance operator Σ_i. We would like to test the hypothesis

$$H_0 : \{\Sigma_1 = \Sigma_2 = \cdots = \Sigma_q\} \quad \text{against} \quad H_1 : \{\text{at least one equality is not true}\}.$$

In this chapter we review the statistical tests that have been proposed so far, highlighting their strengths and weaknesses. These methods will be compared in the following with the proposed permutation approach (see Chapter 7). The works are presented in chronological order. First a generalisation of Levene's test is shown. Then we introduce the theory of bootstrap, that is fundamental to understand the other considered methods. After that, we briefly explain a more recent asymptotic approach by Boente et al. (2014) and the bootstrap-based methodology proposed in Paparoditis and Sapatinas (2014). The last part of the chapter is dedicated to the newest method: it has been proposed by Kashlak et al. (2016) and is based on non-asymptotic confidence sets.

4.1 Generalisation of Levene's test

Levene's test is used to assess the equality of variances for a variable calculated for two or more groups (Levene, 1960). In the univariate case, let x_{ij} be a set of $j = 1,\dots,n_i$ observations in each of $i = 1,\dots,q$ groups. Levene's test statistic is then the ANOVA F-ratio comparing the groups, calculated on the absolute deviations $z_{ij} = |x_{ij} - \bar{x}_i|$ from the groups means $\bar{x}_i = n_i^{-1}\sum_{j=1}^{n_i} x_{ij}$. In other words, calling n the total number of observations $n = n_1 + \cdots + n_q$, $\bar{z}_i$ the mean of z_{ij} for group i and $\bar{z}$ the mean of all z_{ij}, the idea is to compare the test statistic

$$T = \frac{n-q}{q-1}\frac{\sum_{i=1}^{q}(\bar{z}_i - \bar{z})^2}{\sum_{i=1}^{q}\sum_{j=1}^{n_i}(z_{ij} - \bar{z}_i)^2}$$

against $F(\alpha, q-1, n-q)$ where F is a quantile of the F-test distribution with $q-1$ and $n-q$ degrees of freedom and α is the chosen level of significance. For the multivariate case, let $\mathbf{x}_{ij}$ be the vector which denotes the point for the jth observation in the ith group in the multivariate space of p variables. Let $d_e(\cdot,\cdot)$ denote the Euclidean distance between two points of $\mathbb{R}^p$. The centroid vector $\mathbf{c}_i$ for group i is defined as the point that minimises the sum of squared distances to points within that group, i.e. $\sum_{j=1}^{n_i} d_e^2(\mathbf{x}_{ij}, \mathbf{c}_i)$.

Anderson (2006) multivariate analogue to Levene's test is to perform ANOVA on the Euclidean distances from individual points within a group to their group centroid,

$$z_{ij} = d_e(\mathbf{x}_{ij}, \mathbf{c}_i).$$

A p-value for the F-statistic calculated on distances to centroids may be obtained either by using the traditional F-distribution (which assumes the errors in the z_{ij} are approximately normal), or by using a permutation procedure. For data with normally distributed errors, the permutation test and normal-theory test will give similar results (Manly, 2006). Note that the use of an appropriate permutation procedure avoids making any particular assumption regarding the distribution of the distances. Only the exchangeability of points in the multivariate space under the null hypothesis of equal dispersions after centring is assumed.

In the case of functional data analysis, we follow this approach performing an ANOVA analysis on the distances from individual points within each group to the group centroid, i.e. the sample functional mean:

$$z_{ij} = d(x_{ij}, \bar{x}_j), \quad i = 1,\dots,q, \quad j = 1,\dots,n_i,$$

as it has been done in Pigoli et al. (2014). The ANOVA analysis can be both parametric or non-parametric, exploiting all the existing techniques. In the present work we use for the univariate permutation version of the ANOVA in order to compare its empirical power with the method presented in Chapter 4. Any distance between covariance operators can be used, so we resort to those listed in Section 2.2.2.

The main advantage of using this test is that it is easy to implement and not computationally heavy. On the other hand, for the same reasons, it is not very accurate: it focuses only on the differences in global dispersion and it does not contain any reference to the shape of the covariance structure.

4.2 Asymptotic approach

In this section we explore the main steps of the asymptotic test to compare covariance operators of several functional data samples proposed in Boente et al. (2014). The methodology is based on the squared norm of the distances between the estimated covariance operators of each population. The asymptotic distribution of the test statistic under the null hypothesis is derived and, since it depends on parameters of the underlying distribution, also a bootstrap procedure is proposed.

Let us assume that we have independent observations $\mathbf{X}_i = (X_{i1}, \dots, X_{in_i})$ for each group $i = 1, \dots, q$ and define the tensor product operator $x \otimes y : L^2(I) \to L^2(I),\ x, y \in L^2(I)$ as

$$(x \otimes y)z = \langle y, z \rangle x, \quad \forall z \in L^2(I).$$

The starting point is the result given in Dauxois et al. (1982) about the asymptotic behaviour of the sample covariance operator S_i.

Proposition 4.1. *If* $\mathbb{E}\|X_{ij}\|^4 < \infty$, *then* $\sqrt{n_i}(S_i - \Sigma_i)$ *converges in distribution to a zero-mean Gaussian random element of* $\mathcal{F}$, *the Hilbert space of Hilbert-Schmidt operators, with covariance operator* Υ_i *given by*

$$\begin{aligned}\Upsilon_i = & \sum_{m,r,o,p} s_{im} s_{ir} s_{io} s_{ip} \mathbb{E}[f_{im} f_{ir} f_{io} f_{ip}] v_{im} \otimes v_{ir} \tilde{\otimes} v_{io} \otimes v_{ip} \\ & - \sum_{m,r} \phi_{im} \phi_{ir} v_{im} \otimes v_{im} \tilde{\otimes} v_{ir} \otimes v_{ir}\end{aligned} \tag{4.1}$$

where $\tilde{\otimes}$ *stands for the tensor product in* $\mathcal{F}$ *and* $\{v_{il}, l \geq 1\}$ *is an orthonormal basis of eigenfunctions of* Σ_i *with associated eigenvalues* $\{\phi_{il} : l \geq 1\}$ *such that* $\phi_{il} \geq \phi_{i,l+1}$. *The coefficients* s_{im} *are such that* $s_{im}^2 = \phi_{im}$ *while* f_{im} *are standardized coordinates of* $X_i - \mu_i$ *on the basis* $\{v_{il} : l \geq 1\}$.

Two-sample test

We first consider the problem of testing the hypothesis

$$H_0 : \{\Sigma_1 = \Sigma_2\} \quad \text{against} \quad H_1 : \{\Sigma_1 \neq \Sigma_2\}.$$

The considered test statistic is based on the Hilbert-Schmidt distance between the sample covariance operators, i.e. $T = n\|S_1 - S_2\|_{HS}$, where $n = n_1 + n_2$. The idea is to study the asymptotic behaviour of T when H_0 is true. The following result can be derived from Proposition 4.1:

Theorem 4.1. *Let $x_{i1}, \dots, x_{in_i}$ for $i = 1,2$ be independent observations from two independent samples in $L^2(I)$ with mean μ_i and covariance operator Σ_i. Assume that $n_i/n \to \tau_i$ with $\tau_i \in (0,1)$. Let $S_i, i = 1,2$ be the sample covariance operators of the ith population such that $\sqrt{n_i}(S_i - \Sigma_i) \xrightarrow{d} U_i$, with U_i a zero-mean Gaussian random element with covariance operator Υ_i. Denote by $\{\iota_l\}_{l\geq 1}$ the eigenvalues of the operator*

$$\Upsilon = \frac{1}{\tau_1}\Upsilon_1 + \frac{1}{\tau_2}\Upsilon_2$$

with $\sum_{l\geq 1} \iota_l < \infty$. Then,

$$n\|(S_1 - \Sigma_1) - (S_2 - \Sigma_2)\|_{HS}^2 \xrightarrow{d} \sum_{l\geq 1} \iota_l Z_l^2,$$

where Z_l are i.i.d. standard normal random variables. In particular, if $\Sigma_1 = \Sigma_2$ we have that $n\|S_1 - S_2\|_{HS}^2 \xrightarrow{d} \sum_{l\geq 1} \iota_l Z_l^2$.

The result presented in Boente et al. (2014) can actually be used for any set of independent estimators of the covariance operators $\tilde{S}_i$ such that $\sqrt{n_i}(\tilde{S}_i - \Sigma_i) \xrightarrow{d} U_i$. It is also shown that if q_n is a sequence of integers such that $q_n \to \infty$, the fact that $\sum_{l\geq 1} \iota_l < \infty$ implies that the sequence $\mathcal{U}_n = \sum_{l=1}^{q_n} \iota_l Z_i^2$ is Cauchy in L^2 and therefore the limit $\mathcal{U} = \sum_{l\geq 1} \iota_l Z_l^2$ is well defined. Theorem 4.1 implies that, under the null hypothesis, $H_0 : \Sigma_1 = \Sigma_2$, we have that

$$T = n\|S_1 - S_2\| \xrightarrow{d} \mathcal{U} = \sum_{l\geq 1} \iota_l Z_l^2.$$

Hence, an asymptotic test based on T, rejecting for large values of T, allows to test for H_0. It is also worth noticing that under $H_0 : \Sigma_1 = \Sigma_2$, for $i = 1,2$, Υ_i given in (4.1) reduces to

$$\Upsilon_i = \sum_{m,r,o,p} s_m s_r s_o s_p \mathbb{E}[f_{im} f_{ir} f_{io} f_{ip}] v_m \otimes v_r \tilde{\otimes} v_o \otimes v_p - \sum_{m,r} \phi_m \phi_r v_m \otimes v_m \tilde{\otimes} v_r \otimes v_r$$

where, for the sake of simplicity, the subscript 1 has been omitted and s_m denotes $\phi_m^{1/2}$ with

ϕ_m the mth largest eigenvalue and v_m the corresponding eigenfunction. In particular, if the two populations have the same underlying distribution except for the mean and covariance operator, as it happens when comparing Gaussian processes, the random functions f_{2m} have the same distribution as f_{1m} and so, $\Upsilon_1 = \Upsilon_2$. Boente et al. (2014) thus propose the following bootstrap calibration for the distribution of the test:

Algorithm 4.1 (Bootstrap procedure for the two-sample asymptotic test)**.**

Given a sample $x_{i,1},\dots,x_{i,n_i}$, let $\hat{\Upsilon}_i$ be consistent estimators of Υ_i $i = 1,2$.

1. *Define $\hat{\Upsilon} = \hat{\tau}_1^{-1}\hat{\Upsilon}_1 + \hat{\tau}_2^{-1}\hat{\Upsilon}_2$ with $\hat{\tau}_i = n_i/(n_1+n_2)$.*
2. *For $1 \leq l \leq q_n$ denote by $\hat{\iota}_l$ the positive eigenvalues of $\hat{\Upsilon}$*
3. *Generate $Z_1^*,\dots,Z_{q_n}^*$ i.i.d. such that $Z_i^* \sim \mathcal{N}(0,1)$ and let $U_n^* = \sum_{j=1}^{q_n} \hat{\iota}_j Z_j^{*2}$*
4. *Repeat the previous step B times, to get B values of U_n^**

The $(1-\alpha)$-quantile of the asymptotic distribution of T can be approximated by the $(1-\alpha)$-quantile of the empirical distribution of $\mathcal{U}_{nr}^*$ for $1 \leq r \leq B$. The p-value of the test can be evaluated by the number of $\mathcal{U}_{nr}^*$ larger or equal than the observed value of T divided by the number of repetitions B.

Multiple-sample test

This test is then generalised for the multiple sample case. That is, if Σ_i denotes the covariance operator of the ith population, we wish to test the hypothesis

$$H_0 : \{\Sigma_1 = \cdots = \Sigma_q\} \quad \text{against} \quad H_1 : \{\text{at least one equality is not true}\}.$$

Let $n = n_1 + \cdots + n_q$ and assume that $n_i/n \to \tau_i$, $0 \leq \tau_i \leq 1$, $\sum_{i=1}^q \tau_i = 1$. The generalisation of the test described above uses the test statistic

$$T = n \sum_{j=2}^{q} \|S_j - S_1\|_{HS}^2,$$

where S_i stands for the sample covariance operator of the ith population.

Theorem 4.2. *Let $x_{i1},\dots,x_{in_i}$, for $1 \leq i \leq q$, be independent observations from q independent distributions, with mean μ_i and covariance operator Σ_i such that $\mathbb{E}(\|X_{i1}\|^4) < \infty$. Let S_i be the sample covariance operator of the ith population. Assume that $n_i/n \to \tau_i$ with $\tau_i \in (0,1)$ where*

$n = \sum_{i=1}^{q} n_i$. Denote Υ_W the linear operator $\Upsilon_W : L^2(I)^{q-1} \to L^2(I)^{q-1}$ defined as

$$\Upsilon_W(u_1, \dots, u_{q-1}) = \left(\frac{1}{\tau_2} \Upsilon_2(u_1), \dots, \frac{1}{\tau_q} \Upsilon_q(u_{C-1}) \right)$$

where the Υ_i are given in (4.1)*. Let $\{\iota_l\}_{l \geq 1}$ stand for the sequence of eigenvalues of Υ_W ordered in decreasing order. Under H_0, we have:*

$$n \sum_{j=2}^{q} \|S_j - S_1\|_{HS}^2 \xrightarrow{d} \sum_{l \geq 1} \iota_l Z_l^2$$

where $Z_l \sim \mathcal{N}(0, 1)$ are independent.

This theorem is a natural extension of its analogous in the finite-dimensional case. Here too, a bootstrap procedure can be considered. In order to estimate ι_l, one can consider estimators of the operators Υ_i for $1 \leq i \leq q$ and thus estimate Υ_W. Therefore, if $\hat{\iota}_l$ are positive eigenvalues of $\hat{\Upsilon}_W$, a bootstrap procedure can be defined using steps 3. and 4. of Algorithm 4.

4.3 Empirical bootstrap for the equality of covariance operators

Paparoditis and Sapatinas (2014) presented a bootstrap-based methodology for testing hypotheses about equality of certain characteristics of the distributions between different populations in the context of functional data. It is very flexible: it can potentially be applied to different test statistics and for more than two populations. Among other things, the proposed procedure can be applied to the problems of comparing the covariance operators between several populations. The basic idea behind this test is to resample the observed functional data set in such a way that the obtained bootstrap functional pseudo-observations satisfy the null hypothesis of interest. This requirement leads to a particular resampling scheme that, when (for example) testing the equality of covariance operators, automatically generates pseudo-functional observations with identical covariance operators of the observed functional data. We show here how the methodology proposed in Paparoditis and Sapatinas (2014) can be used for the case of covariance operators.

Let $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_q)$ be the observed collection of random functions satisfying

$$x_{ij}(t) = \mu_i(t) + \varepsilon_{ij}(t), \quad i = 1, \dots, q, \quad j = 1, \dots, n_i, \quad t \in I.$$

Let Σ_i be the covariance operators of the ith population. It is assumed that the q populations are independent and, for each $i = 1, 2, \dots, q$ and $j = 1, \dots, n_i$, the ε_{ij} are independent and identically distributed random elements with $\mathbb{E}[\varepsilon_{ij}(t)] = 0$, $t \in T$ and $\mathbb{E}\|\varepsilon_{ij}\|^4 < \infty$. It is also assumed that there exist at least p distinct (positive) eigenvalues of the covariance operators $\Sigma_i,\ i = 1,\dots,q$.

The objective is to test the hypothesis:

$$H_0 : \{\Sigma_1 = \Sigma_2 = \cdots = \Sigma_q\} \quad \text{against} \quad H_1 : \{\text{at least one equality is not true}\}. \tag{4.2}$$

Let T be a given test statistic of interest for testing (4.2) which is based on X. Assume, without loss of generality, that T rejects the null hypothesis H_0 when $T > T_\alpha$ where, for $\alpha \in (0,1)$, T_α denotes the critical value of the test. The algorithm is composed by the following steps:

Algorithm 4.2 (Empirical bootstrap procedure)**.**

1. *Calculate the sample mean function of each population:*

$$m_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_{ij}, \quad i = 1,2,\dots,q.$$

2. *Calculate the residual function with respect to the mean for each observation:*

$$\varepsilon_{ij} = x_{ij} - m_i, \quad i = 1,2,\dots,q, \quad j = 1,2,\dots,n_i.$$

3. *Generate bootstrap functional pseudo-observations according to*

$$x^*_{ij} = m_i + \varepsilon^*_{ij}, \quad i = 1,2,\dots,q, \quad j = 1,2,\dots,n_i$$

where $\varepsilon^*_{i,j} = \varepsilon_{I,J}$ *where* (I,J) *is a pair of random variables. I takes values in* $\{1,2,\dots,q\}$ *with probability* $\mathbb{P}(I = i) = n_i/n$ *and, given $I = i$, the random variable J has the discrete uniform distribution in the set* $\{1,2,\dots,n_i\}$, *i.e.* $\mathbb{P}(J = j|I = i) = n_i^{-1}$ *for* $i = 1,2,\dots,q$, $j = 1,2,\dots,n_i$.

4. *Let T^* be the same test statistic as T calculated using the bootstrap functional pseudo-observations X^*.*

5. *For any given* $\alpha \in (0,1)$, *reject the null hypothesis H_0 if and only if $T > T^*_\alpha$ where T^*_α denotes the α-quantile of the distribution of T^* given the functional observations X.*

In the specific case of two-sample tests for covariance operators, three different test statistics have been considered by Paparoditis and Sapatinas (2014). First, the Hilbert-Schmidt norm of the difference between the sample covariance operators, which is also used in Boente et al. (2014)

$$T = n\|S_1 - S_2\|_{HS}.$$

Then, the approach that looks at the distance between the sample covariance operators S_1 and S_2 and the pooled sample covariance function S_{pool}, based on the entire set of functional observations, is studied. The idea is to look at the projections of S_1 and S_2 on certain directions. Such approach has been considered by Panaretos et al. (2010) and Fremdt et al. (2013) and requires some preliminary definitions. With the same notation used in Section 2.2.1, we denote by (ϕ_k, v_k), $k = 1, \dots, n$ the eigenvalues and eigenfunctions of the pooled sample covariance operator S_{pool} with $\phi_1 \geq \phi_2 \geq \dots$ and assume that the v_k form an orthonormal system. Select a natural number p and consider, for $i = 1, 2, \dots, p$ the projections

$$a_{k,j} = \langle x_{k,j} - \bar{x}_{k,n_k}, v_i \rangle = \int_T (x_{k,j} - m_k) v_i \, dt, \quad j = 1, 2, \dots, n_k, \; k = 1, 2$$

Now, for $1 \leq r, m \leq p$, consider the matrices A_{k,n_k} with elements

$$A_{k,n_k}(r,m) = \frac{1}{n_k} \sum_{j=1}^{n_k} a_{k,j}(r) a_{k,j}(m), \quad k = 1, 2$$

and define

$$\Delta(r,m) = A_{1,n_1}(r,m) - A_{2,n_2}(r,m).$$

$\Delta(r,m)$ is the projection of the difference $S_1 - S_2$ in the direction of $v_r v_m$. Based on this, Panaretos et al. (2010) defined the following test statistic for the Gaussian case:

$$T = \frac{n_1}{n_2} \sum_{1 \leq r,m \leq p} \frac{\Delta(r,m)}{2\phi_r \phi_m}.$$

As regarding the non-Gaussian framework, Fremdt et al. (2013) considered the matrix $\Delta = (\Delta(r,m))_{r,m=1,\dots,p}$ and defined $\xi = \text{vech}(\Delta)$, i.e. the vector containing the elements on and below the main diagonal of Δ. The proposed test statistic is

$$T = \frac{n_1 n_2}{n} \xi' L^{-1} \xi$$

where L is an estimator of the asymptotic covariance matrix of ξ. For these three test statistics, the asymptotic distributions are known, so Paparoditis and Sapatinas (2014) showed that Algorithm 4.2 correctly approximates them.

4.4 Inference on covariance operators via concentration inequalities

More recently, Kashlak et al. (2016) proposed another approach to the multiple testing of covariance operators, based on non-asymptotic confidence sets. In Section 4.4.1 we recall the definition given by Kashlak et al. (2016) of such confidence regions and in Section 4.4.2 we show how they are used to define a global test for the equality of covariance operators.

4.4.1 Confidence sets for covariance operators

Let $x_1, \dots, x_n \in L^2(I)$ be independent and identically distributed observations with mean zero and covariance operator Σ. The initial goal is to construct a confidence set for Σ with respect to some metric $d(\cdot,\cdot)$ of the form

$$\left\{\Sigma : d(S,\Sigma) \le r(n,S,\alpha)\right\},$$

which has coverage $1-\alpha$ for any desired $\alpha \in (0,1)$ and a radius r depending only on the data and α. The metrics used by Kashlak et al. (2016) are the p-Schatten norms.

Definition 4.1. *Given two separable Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, a bounded linear operator $\Sigma : \mathcal{H}_1 \to \mathcal{H}_2$ and $p \in [1,\infty)$, then the p-*Schatten norm *is*

$$\|\Sigma\|_p^p = trace\left[(\Sigma'\Sigma)^{p/2}\right].$$

For $p = \infty$, the Schatten norm is the operator norm:

$$\|\Sigma\|_\infty = \sup_{x \in \mathcal{H}_1} \frac{\|\Sigma x\|_{\mathcal{H}_2}}{\|x\|_{\mathcal{H}_1}}.$$

*In the case that Σ is compact, self-adjoint and trace-class, then, given the associated eigenvalues $\{\tilde{\phi}_n\}_{n=1}^{\infty}$, the p-*Schatten norm *coincides with the l^p norm of the eigenvalues*

$$\|\Sigma\|_p^p = \begin{cases} \|\tilde{\phi}\|_{l^p}^p = \sum_{n=1}^{\infty} |\tilde{\phi}_n^p|, & if\, p \in [1,\infty), \\ \max_{n\in\mathbb{N}} |\tilde{\phi}_n|, & if\, p = \infty. \end{cases}$$

Clearly, for $p = \infty$ this is equivalent to the spectral distance defined in Equation (2.2). The construction of the confidence set is based on Talagrand's concentration inequality (Talagrand, 1996) with explicit constants and the Rademacher symmetrisation technique (Giné and Nickl, 2015). The latter requires the use of the namesake Rademacher random variables.

Definition 4.2. *A random variable $\chi \in \mathbb{R}$ has a Rademacher distribution if*

$$\mathbb{P}(\chi = 1) = \mathbb{P}(\chi = -1) = \frac{1}{2}.$$

Now, for some desired p-Schatten norm $\|\cdot\|_p$, with $p \in [1, \infty)$, define the weak variance ς^2 as

$$\varsigma^2 = \frac{1}{n} \sum_{i=1}^{n} \sup_{\|\Xi\|_{\frac{p}{p-1}} \le 1} \mathbb{E}\Big[\langle x_i - \mathbb{E}[x_i^{\otimes 2}], \Xi\rangle^2\Big]$$

where $x^{\otimes 2} = x \otimes x$ and the supremum is being taken over a countably dense subset of the unit ball of the Hilbert space of bounded linear operators mapping $L^2(I)$ to $L^2(I)$. The $(1-\alpha)$-confidence set defined by Kashlak et al. (2016) is

$$\left\{\Sigma : \|S - \Sigma\|_p \le \|R_n\|_p + \varsigma\Big[-\frac{2}{n}\log(2\alpha)\Big]^{1/2} - \frac{\varsigma \log(2\alpha)}{3n}\right\}, \tag{4.3}$$

where R_n is the Rademacher average

$$R_n = \frac{1}{n}\sum_{i=1}^{n} \chi_i\big[(x_i - m)^{\otimes 2} - S\big].$$

4.4.2 Multiple-sample comparison

Now, let the q samples be $x_{i1}, \dots, x_{in_i}$, $i = 1, \dots, q$ where for each sample $i = 1, \dots, q$ and all elements $j = 1, \dots, n_i$, x_{ij} has covariance Σ_i. In order to design a test for the following two hypotheses

$$H_0 : \{\Sigma_1 = \cdots = \Sigma_q\} \quad \text{against} \quad H_1 : \{\text{at least one equality is not true}\},$$

the p-Schatten norms with the concentration inequality based confidence sets of Equation (4.3) are used. Let S_i be the empirical estimate of the covariance operator for the ith sample and let S be the estimate of the covariance operator for the total dataset. Also, let the total data size be $n = n_1 + \cdots + n_q$ and ς_i be the weak variance for sample i, then the pooled variance is defined as $\varsigma^2_{\text{pool}} = n^{-1}\sum_{i=1}^{q} n_i \varsigma_i^2$. In practice, $\varsigma^2_{\text{pool}}$ is estimated from the data in order to have confidence regions only depend on the data. More details about the practical computation of the weak variance can be found in the original work. Taking inspiration from the standard analysis of variance (Casella and Berger, 2002, Chapter 11), the rejection region is

$$\left\{x : \sum_{i=1}^{q} \|S_i - S\|_p > \sum_{i=1}^{q} \left\| \sum_{j=1}^{n_i} \chi_{ij}(x_{ij}^{\otimes 2} - S)\right\|_p + \left(\sum_{i=1}^{q} \frac{\varsigma_{\text{pool}}}{n_i}\right)^{1/2} (-2\log 2\alpha)^{1/2} + \left(\sum_{i=1}^{q} \frac{\varsigma_{\text{pool}}}{n_i}\right) \log\frac{2\alpha}{3}\right\}.$$

The size of the test induced by this rejection region is significantly less than the target size α, due to the use of multiple concentration inequalities. Hence, tuning inequalities is required to yield a useful test. Kashlak et al. (2016) determined experimentally that the coefficients of $1-q^{-1/2}$ for the Rademacher term and $(q+2)/(q+3)$ for the deviation term improve the size of the confidence region.

Chapter 5

Permutation tests for covariance operators

In this chapter we show how the distances introduced in Section 2.2.2 can be used to make inference on the covariance operators of multiple data samples. First of all, we briefly describe the permutation test proposed by Pigoli et al. (2014) for the two-sample case. Then, we present the methodological development of the thesis: the extension of this method to the multiple-sample case, by means of the NPC methodology of Pesarin and Salmaso (2010) explained in Chatper 3. Lastly, we discuss the different possible choices for the permutation strategy. Three different approaches are considered and their impact on the performance of the test is illustrated with a few synthetic examples.

5.1 The two-sample case

A permutation test for assessing the equality of the covariance operators of two functional data samples has been formulated in Pigoli et al. (2014). The main idea is to use the distance between two sample covariance operators to carry out inference on the equality of the true covariance operators. In order to test the hypotheses

$$H_0 : \Sigma_1 = \Sigma_2 \quad \text{against} \quad H_1 : \Sigma_1 \neq \Sigma_2,$$

it has been proposed to use a distance between covariance operators like, for example, those introduced in Section 2.2.2 as a test statistic. In this way, the two hypotheses can be reformulated as:

$$H_0 : d(\Sigma_1, \Sigma_2) = 0 \quad \text{against} \quad H_1 : d(\Sigma_1, \Sigma_2) \neq 0,$$

where $d(\cdot,\cdot)$ is a generic distance between covariance operators. For this formulation of the permutation test, equality of mean functions is essential to ensure the exchangeability of the observations under the null hypothesis. However, if the two groups have different (and unknown) means, an approximated permutation test can be performed, having first centred the curves using their sample means. This is a common strategy for testing scaling parameters, such as variance, for univariate real random variables (Good, 2005).

In analogy with the two-sample univariate permutation test explained in Section 3.1, the MC algorithm for computing a p-value for the test is the following:

Algorithm 5.1 (Two-sample permutation test for the equality of covariance operators)**.**

1. *Compute the observed test statistic, that corresponds to the distance between the two sample covariance operators according to the chosen metric:*

   $$T^0 = d(S_1, S_2);$$

2. *Apply* B *random permutations to the labels of the sample curves* $\mathbf{u}^*_{(b)}$, $b = 1, \dots, B$;

3. *For each of them, let* $S_i^{(b)}$ *be the sample covariance of the permuted sample* $\mathbf{X}_i^*$, $i = 1, 2$, *compute* $d(S_1^{(b)}, S_2^{(b)})$, $b = 1, \dots, B$;

4. *The* p*-value of the test is the proportion of permutations that give a greater distance between the sample covariance operators than the one observed:*

   $$\hat{\lambda} = \frac{\sum_{b=1}^{B} \mathbb{1}\left[d\left(S_1^{(b)}, S_2^{(b)}\right) \geq d\left(S_1, S_2\right)\right]}{B}.$$

5.2 The multiple-sample case

In this section we show how it is possible to extend the approach proposed in Pigoli et al. (2014) to the case of multiple samples of functional data. Let us suppose we have q groups of functional data

$$x_{i1}, \dots, x_{in_i} \in L^2(I), \quad i = 1, \dots, q.$$

We assume they are independent and identically distributed samples from a random process with distribution P_i, mean μ and covariance operator Σ_i. We would like to test if the covariance operators are all equal. The global null hypothesis can be viewed as an intersection of partial null hypotheses

$$H_0 : \{\Sigma_1 = \Sigma_2 = \cdots = \Sigma_q\} = \bigcap_{i \neq j} H_0^{ij}, \quad \text{where} \quad H_0^{ij} : \{\Sigma_i = \Sigma_j\}$$

and the global alternative hypothesis can be viewed as the union of the corresponding alternative hypotheses

$$H_1 : \{\text{at least one equality is not true}\} = \bigcup_{i \neq j} H_1^{ij}, \quad \text{where} \quad H_1^{ij} : \{\Sigma_i \neq \Sigma_j\}.$$

The idea is to combine the $k = q(q-1)/2$ two-sample tests for each pair of groups in a global test. This kind of approaches is called Union-Intersection (UI). This term encompasses any testing framework in which it is assumed that the hypotheses H_0 and H_1 can be equivalently broken down as

$$H_0 : \left\{ \bigcap_{i=1}^{k} H_{0i} \right\} \quad \text{and} \quad H_1 : \left\{ \bigcup_{i=1}^{k} H_{1i} \right\}$$

and that the global test is obtained by combining a suitable list of partial tests T_i, each one specific for testing H_{0i} against H_{1i}, $i = 1, \ldots, k$. The UI principle for multivariate testing has been studied for the first time by Roy (1953). When they are known, the provided UI solutions usually coincide with those obtained by likelihood techniques. But, outside that setting, no other solutions in closed form have been found so far. In this respect, Sen (2007) stated:

> *The crux of the problem is however to find the distribution theory for the maximum of these possibly correlated statistics. Unfortunately, this distribution depends on the unknown P, even under the null hypothesis. An easy way to eliminate this impasse is to take recourse to the permutation distribution theory.*

This proposal is not easy to achieve if one wishes to treat the underlying dependence by using suitable estimators of all coefficients, the number and type of which are usually unknown. Indeed, we will see in Section 5.3 how dependencies among the partial hypotheses can be much more complex than linear. However, this proposal has a general solution when it is possible to deal with such a dependence in a non-parametric way. The goal is obtained within the conditional testing principle by conditioning on the whole data set $\mathbf{X}$. If under H_0 the data set $\mathbf{X}$ is a sufficient statistic for the underlying distribution P, this principle provides exact permutation solutions even in multivariate settings and constrained alternatives. The related methods are based on the NPC presented in Section 3.2.

So we want to apply NPC to the considered problem. That is, we would like to use the algorithms reported in Section 3.2.3 setting each element of the vector of partial test statistics $\mathbf{T} = \{T_1, \dots, T_k\}$ to the distance between the permutation sample covariance operators $d(S_i^*, S_j^*)$ for some $1 \leq i, j \leq q$. It is immediate to verify that the so-defined partial tests $H_0 : d(\Sigma_i, \Sigma_j) = 0$ against $H_0 : d(\Sigma_i, \Sigma_j) \neq 0$ marginally satisfy Assumptions 3.1 and 3.2 for any of the distances presented in Section 2.2.2. Therefore, Algorithms 3.2 and 3.3 can be applied to any functional dataset **X** using $\mathbf{T} = \{d(S_i^*, S_j^*),\ i, j = 1, \dots, k, i \neq j\}$ and any combining function Ψ satisfying Properties 3.1, 3.2 and 3.3, i.e. $\Psi \in \mathcal{C}$. The choice of the combining function will be analysed in more detail in Chapters 6 and 7, while the possible permutation strategies are investigated in the next section. We obtain the following algorithm:

Algorithm 5.2 (Multiple-sample permutation test for the equality of covariance operators)**.** *Let $x_{ij},\ j = 1, \dots, q,\ i = 1, \dots, n_j$ be the considered dataset.*

1. *For all $j = 1, \dots, q, i = 1, \dots, n_j$, let*

   $$\tilde{x}_{ij} = x_{ij} - m_j,$$

   where m_j is the sample mean of group j;

2. *Let $\mathbf{T}^0$ be the k-dimensional vector containing the pairwise distances between the covariance operators*

   $$d(S_j, S_l), \forall j = 1, \dots, q,\ l = 1, \dots, q,\ j \neq l;$$

3. *Consider a random permutation $\mathbf{u}^*$ of the data labels and compute the k-dimensional vector $\mathbf{T}^*$ containing*

   $$d(S_j^*, S_l^*), \forall j = 1, \dots, q,\ l = 1, \dots, q,\ j \neq l;$$

4. *Carry out* B *independent repetitions of the previous step, obtaining $\{\mathbf{T}^{(b)}\}_{b=1}^{B}$, a random sampling from the permutational distribution of $\mathbf{T}$;*

5. *Let*

   $$\hat{\lambda}_i(d|\mathcal{X}_{/\mathbf{X}}) = \frac{\sum_b \mathbb{1}\left[d(S_i^{(b)}, S_j^{(b)}) \geq d\right]}{B}$$

   be consistent estimates of $\lambda_{jl} = \mathbb{P}(d|\mathcal{X}_{/\mathbf{X}})$, $d \in \mathbb{R}, d \geq 0$;

6. *Compute the estimated partial* p*-values of the test as*

   $$\hat{\lambda}_{jl} = \hat{\lambda}_{jl}(d(S_i, S_l)|\mathcal{X}_{/\mathbf{X}});$$

7. *Combine the $\hat{\lambda}_i$ to obtain the observed global test statistic*

$$T^0_{\Psi} = \Psi(\hat{\lambda}_{1,2}, \hat{\lambda}_{1,3}, \dots, \hat{\lambda}_{k,k-1});$$

8. *Compute the* b*th combined value as*

$$T^{(b)}_{\Psi} = \Psi(\hat{\lambda}^{(b)}_{1,2}, \hat{\lambda}^{(b)}_{1,3}, \dots, \hat{\lambda}^{(b)}_{k,k-1}), b = 1, \dots, B,$$

where $\hat{\lambda}^{(b)}_{jl} = \hat{\lambda}_{jl}(d(S^{(b)}_j, S^{(b)}_l)|\mathcal{X}_{/\mathbf{X}}), j = 1, \dots, q,\ l = 1, \dots, q,\ j \neq l;$

9. *The* p*-value of the combined test is estimated as*

$$\hat{\lambda}_{\Psi} = \frac{\Sigma_b \mathbb{1}[T^{(b)}_{\Psi} \geq T^0_{\Psi}]}{B};$$

10. *If* $\hat{\lambda}_{\Psi} \leq \alpha$, H_0 *is rejected.*

If we make the following additional assumptions

1. when n goes to infinity, then so also do the sample sizes of all groups, that is $n \to \infty$ implies $\min_i n_i \to \infty$,
2. the number B of MC iterations goes to infinity,
3. k and α are fixed,

then it is possible to prove that the test we obtain is strongly consistent and unbiased test for the overall null hypothesis H_0 against the alternative H_1. To be more precise, the following theorems hold:

Theorem 5.1. *If partial permutation tests T_i, $i = 1, \dots, k$, are marginally unbiased and at least one is strongly consistent for respectively H_{0i} against H_{1i} , then $T_{\Psi} = \Psi(\lambda_1, \dots, \lambda_k)$, for every $\Psi \in \mathcal{C}$, is a strongly consistent combined test for $H_0 : \{\cap_i H_{0i}\}$ against $H_1 : \{\cup_i H_{1i}\}$.*

Theorem 5.2. *If, given a data set X and any $\alpha > 0$, partial permutation tests T_i, $i = 1, \dots, k$ are all marginally unbiased for respectively $H_0 i$ against $H_{1i}, i = 1, \dots, k$, so that their associated p-values λ_i, $i = 1, \dots, k$, are positively dependent, then $T_{\Psi} = \Psi(\lambda_1, \dots, \lambda_k)$, for every $\Psi \in \mathcal{C}$, is an unbiased combined test for $H_0 : \{\cap_i H_{0i}\}$ against $H_1 : \{\cup_i H_{1i}\}$.*

The proofs of these statements can be found in Pesarin and Salmaso (2010).

It is important to note that, even if the test statistic is based on the sample covariance operator, we are supposing, under the null hypothesis, that the distributions related to the groups of data are the same, i.e.

$$H_0 : \{P_1 = \cdots = P_q\} \quad \text{against} \quad H_1 : \{\text{at least one equality is not true}\}.$$

Therefore, if the null hypothesis is rejected, it is more appropriate to say that there is evidence to affirm that at least two of the distributions generating the observed data samples are different.

Algorithm 5.2 has been implemented in the function `ksample.perm` of the R package “fdcov”. For further details, refer to Appendix A.

5.3 Permutation strategies

Step 3. of Algorithm 5.2 requires to choose a permutation of the original dataset $\mathbf{X}$. It is important to note that any subset of the permutations available can be used, provided that its cardinality N is known, so that one can assign probability $1/N$ to each of them. In Solari et al. (2009), three different ways of permuting data are proposed.

Pooled permutations

This is the simplest idea: to perform permutations involving the whole dataset. This can be done because, under H_0, the observations of all groups are exchangeable. However, this strategy does not allow to test also the partial hypotheses, since each comparison involves not only the observations belonging to the pair of considered groups, but also those of the other groups. Therefore, the resulting global p-value is correct, yet the partial p-values would not be accurate when doing post-hoc comparisons.

Paired permutations

The second proposal could be to apply paired permutations: while comparing the ith and jth groups, the inference is made on the paired vector $x_{ij} = (x_i, x_j)$ independently. The result would be opposite than the one obtained with pooled permutations. The partial tests are done exactly as in the two-sample case, but the global test is not reliable since this method does not take properly into account the dependencies between the marginal tests.

Synchronised permutations

In conclusion, we want paired permutations to be done not independently but jointly. At the same time, we would like to keep the partial comparisons separate, so as to be able to do post-hoc comparison with no additional computational effort. Then, if the design is balanced, i.e. $n_1 = \cdots = n_q = \bar{n}$, a further possibility is to apply synchronised permutations. The basic concept of synchronised permutations is exchanging the same number v of units between each pair of blocks. Applying synchronised permutations allows both maintaining the dependencies among partial tests and involving the observations of each comparison at the same time. First of all, we build the pseudo-data matrix

$$\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_1 & \dots & \mathbf{x}_{q-1} \\ \mathbf{x}_2 & \mathbf{x}_3 & \dots & \mathbf{x}_q \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^1 & \dots & x_1^{q-1} \\ x_2^1 & x_2^1 & \dots & x_2^{q-1} \\ \vdots & \vdots & & \vdots \\ x_{\bar{n}}^1 & x_{\bar{n}}^1 & \dots & x_{\bar{n}}^{q-1} \\ & & & \\ x_1^2 & x_1^3 & \dots & x_1^q \\ x_2^2 & x_2^3 & \dots & x_2^q \\ \vdots & \vdots & & \vdots \\ x_{\bar{n}}^2 & x_{\bar{n}}^3 & \dots & x_{\bar{n}}^q \end{bmatrix} \tag{5.1}$$

where each column is composed by two different samples of curves. The Constrained Synchronized Permutations (CSPs) consist in exchanging units in the same original position within each block. This can be done by permuting the rows of the pseudo-data matrix. Since there are

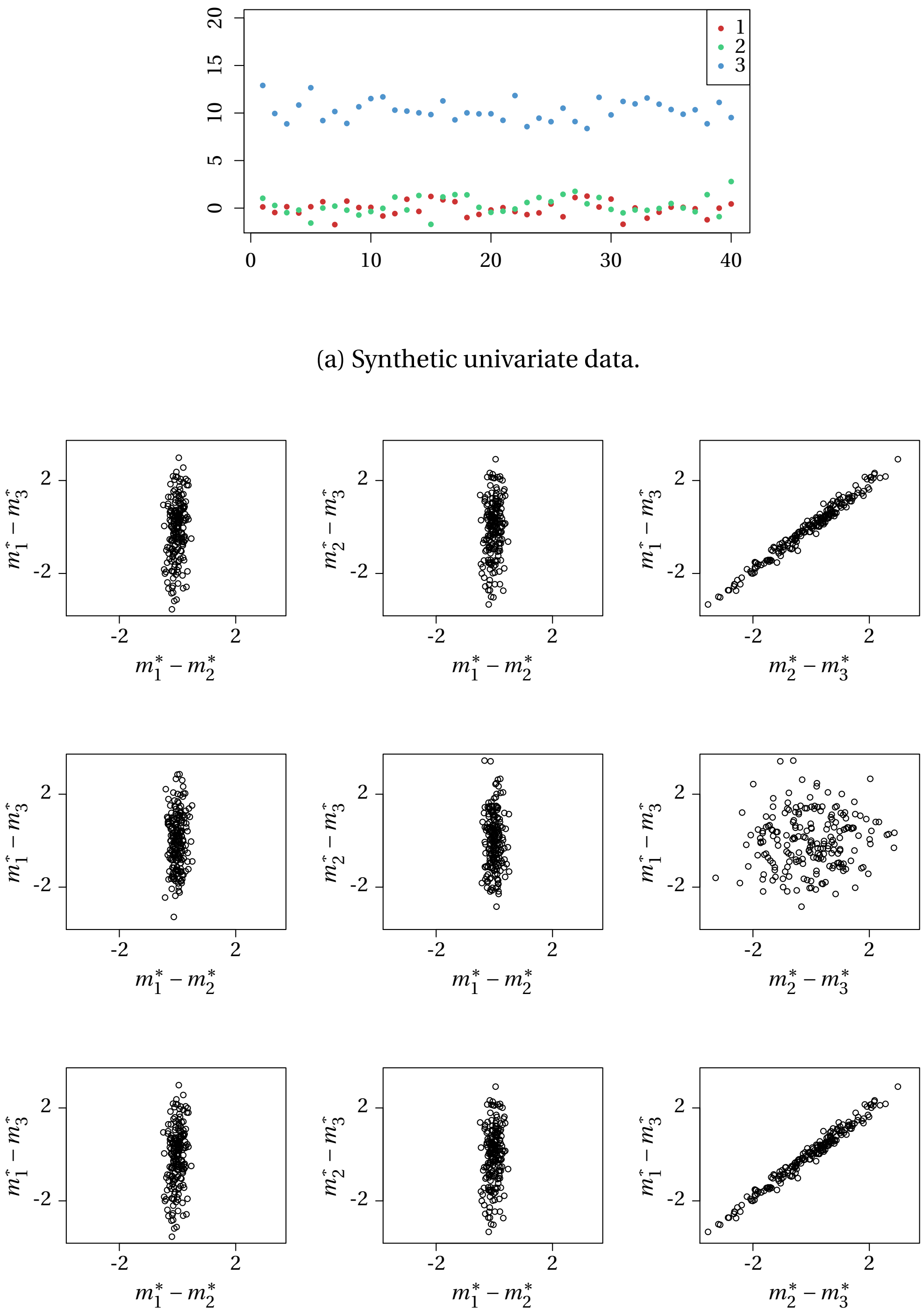
$$N_{\text{CSP}} = \binom{2\bar{n}}{\bar{n}}$$

possible ways to exchange units in the first pair of blocks, N_{CSP} is the cardinality of the CSPs. If the distance $d(\cdot,\cdot)$ is symmetric, then the number of distinct values of T^* is $N_{\text{CSP}}/2$. As pointed out in Section 3.1, being a permutation test, the attainable significance levels are multiples of $1/N$, where N is the cardinality of distinct values of the test statistic. Hence, if the number of observations in each sample n is very small, CSPs may give a minimum achieved significance level that is higher than the first type error rates commonly used. For this reason, Unconstrained Synchronized Permutations (USPs) have been introduced. USPs do not require the exchange units to be in the same original position within the blocks. The only requirement is that the number of exchanges is the same. Since the number of distinct values of T rapidly increases with n, in general it is recommended to use this strategy when few replicates are available (Solari et al., 2009). However, in FDA applications the dimension of the sample is usually large enough to not incur in this problem.

We give here an heuristic explanation of the differences between these methods and compare the performances of the three types of permutations by applying them to synthetic data. First we show the simple example of making inference on the mean of three groups of univariate data. Then we simulate three groups of functional data and repeat the experiment. Synthetic data for the first case are generated by sampling 40 observations from a Gaussian distribution with mean equal to zero for the first two groups and equal to 10 for the third (Figure 5.1a). In the case of functional data, instead, first we define the two sample covariance operators Σ_1 and Σ_2 as in Pigoli et al. (2014), that is we set Σ_1 and Σ_2 equal to the sample covariance operators for the male and female subjects, respectively, of the Berkeley growth study dataset Ramsay and Silverman (2005), rescaled to lie in $[0,1]$. Then we simulate data with with the same mean function $\sin(x)$. and covariance operators Σ_1 for the first two and

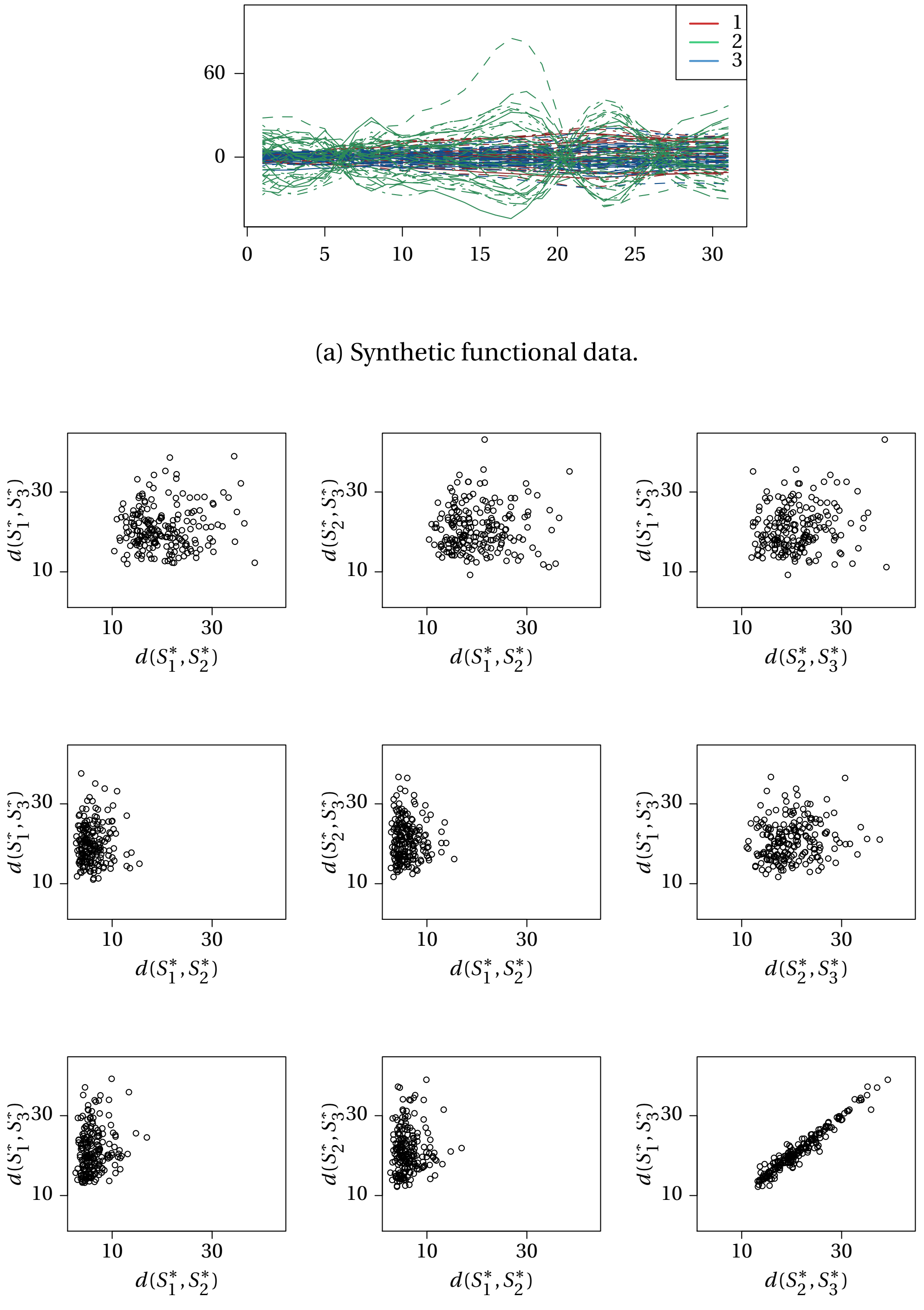
$$\Sigma_{10} = [(\Sigma_1)^{1/2} + 10\{(\Sigma_2)^{1/2}\tilde{R} - (\Sigma_1)^{1/2}\}][(\Sigma_1)^{1/2} + 10\{(\Sigma_2)^{1/2}\tilde{R} - (\Sigma_1)^{1/2}\}]'$$

for the third, where $\tilde{R}$ is the operator minimizing the Procrustes distance between Σ_1 and Σ_2. The synthetic dataset obtained is shown in Figure 5.2a. Figures 5.1 and 5.2 show the joint distributions of the vectors of partial statistics $\mathbf{T} = (T_1, \ldots, T_k)$ when applying pooled, paired, and synchronised permutations, respectively, with the number of replicates B equal to 200. Note how the inner dependencies among partial tests are maintained by pooled and synchronised permutations, while they are cancelled by the independent permutations of the pairwise strategy. Also, these examples illustrate how synchronised permutations perform better than pooled permutations when the dependencies are way more complex than linear, as in the case of test statistic related to covariance operators.



(a) Synthetic univariate data.

Figure 5.1: Joint distribution of $\mathbf{T} = (T_{12}^*, T_{13}^*, T_{23}^*)$ when permuting the whole data vector (pooled permutations, first row), when permuting x_{jk} independently (paired permutations, second row) and when using synchronised permutations (third row). The test statistic used is the difference between sample means, i.e. $T_{ij}^* = m_i^* - m_j^*$.



(a) Synthetic functional data.

Figure 5.2: Functional data. Joint distribution of $\mathbf{T} = (T_{12}^*, T_{13}^*, T_{23}^*)$ when permuting the whole data vector (pooled permutations, first row), when permuting x_{jk} independently (paired permutations, second row) and when using synchronised permutations (third row). The test statistic used is the square root distance between sample covariance operators, i.e. $T_{ij}^* = d_{\text{SQ}}(S_i^*, S_j^*)$.

Chapter 6

Post-hoc comparisons

After performing the global test, if the null hypothesis H_0 is rejected in favour of the alternative H_1, it is often of interest to find out which of the data samples led to this conclusion. In other words, if the null hypothesis

$$H_0 : \bigcap_{i \neq j} H_0^{ij}, \quad \text{with} \quad H_0^{ij} : \{\Sigma_i = \Sigma_j\}$$

is rejected in favour of

$$H_1 : \bigcup_{i \neq j} H_1^{ij}, \quad \text{with} \quad H_1^{ij} : \{\Sigma_i \neq \Sigma_j\},$$

i.e. at least one of the H_1^{ij} is true. In this chapter we show how to simultaneously assess which of the partial alternative hypotheses are true. It has been pointed out in Chapter 5 that one of the advantages of the NPC is that partial p-values are computed at the same time of the global one. Therefore, the post-hoc comparisons can be done with a small computational effort. Thus, we investigate here the methods that allow to control the Family-Wise Error Rate (FWE). In particular, in Section 6.1 we introduce the basic definitions and notations. In Section 6.2 are illustrated some single-step methods that can be used to adjust the partial p-value in order to account for the multiplicity. In Section 6.3, it is explained how single-step methods can be made less conservative by introducing an order of the partial hypotheses. The so-obtained procedures are referred to as step-down methods. However, we will see that they can only be applied for one of the combining functions introduced in Section 3.2.2. Therefore, in Section 6.4 we present the closed testing procedure, which is a little more conservative but can be utilised with any combining function.

6.1 Family-wise error rate and adjusted p-values

One solution for solving the multiplicity problem is to make the individual tests more conservative, using a procedure by which each of the k hypotheses $H_{0i}, i = 1, \dots, k$ is determined to be rejected or accepted at a particular level α. Suppose one wants to test a family of k null hypotheses $H_{01}, \dots, H_{0k}$ each against $H_{11}, \dots, H_{1k}$ respectively. Summing the test results gives the following table and related random variables:

	H_0 true	H_1 true	Total
H_1 declared significant	V	W	r
H_1 declared non-significant	U	Y	k-r
Total	k_0	$k - k_0$	k

Definition 6.1. *The* Family-Wise Error Rate *is the probability of making at least one type I error in the family, i.e.*

$$FWE = \mathbb{P}(V \geq 1) = 1 - \mathbb{P}(V = 0).$$

More precisely, according to Westfall and Young (1993), a simultaneous procedure is said to control the Family-Wise Error Rate in the weak sense (FWEC) if

$$\text{FWEC} = \mathbb{P}(\text{Reject at least one } H_{0i} | \text{All } H_{0i} \text{ are true}) \leq \alpha,$$

that is, it controls the FWE under the complete null hypothesis. It also controls the Family-Wise Error Rate in the strong sense (FWEP)

$$\text{FWEP} = \mathbb{P}(\text{Reject at least one } H_{0i},\ i = j_1, \dots, j_t | H_{0j_1}, \dots, H_{0j_t} \text{are true}) \leq \alpha,$$

regardless of which subset of hypotheses happens to be true.

Thus, once we have computed the partial p-values by means of Algorithm 5.2, we would like to compute an adjusted p-value $\tilde{\lambda}_i, i = 1, \dots, k$ for each test so that the decision to reject H_{0i} at FWE $= \alpha$ is obtained merely by noting whether $\tilde{\lambda}_i \leq \alpha$. Similarly to the definition of an ordinary (unadjusted) p-value, the mathematical definition of an adjusted p-value, $\tilde{\lambda}_i$, is

$$\tilde{\lambda}_i = \inf\{\alpha | H_{0i} \text{ is rejected at FWE} = \alpha\}$$

In other words, $\tilde{\lambda}_i$ is the smallest significance level for which one still rejects H_i, given a particular simultaneous test procedure.

6.2 Single-step methods

Single-step methods are Simultaneous Test Procedures (STPs) that perform equivalent multiplicity adjustments for all tests, regardless of the ordering of the observed p-values $\lambda_1, \dots, \lambda_k$ and without considering any predetermined sequence of hypotheses.

Bonferroni-type methods

The simplest single-step method is the Bonferroni method (Bonferroni, 1936), which rejects hypothesis H_{0i} when the p-value λ_i is less than α/k, where α is the chosen FWE level and k is the number of tests, i.e.

$$\tilde{\lambda}_i = \min(k\lambda_i, 1).$$

Using this adjustement, the FWE is always controlled in the weak sense:

$$\begin{aligned}\mathbb{P}(\text{Reject at least one } H_{0i}|H_0) &= \mathbb{P}(\min_{1\le i\le k} \Lambda_i \le \alpha/k|H_0) \\ &\le \sum_{i=1}^{k} \mathbb{P}(\Lambda_i \le \alpha/k|H_0).\end{aligned}$$

where Λ_i is the random variable associated to the observed p-value λ_i and the inequality is given by the well known Bonferroni inequality. Assuming that all p-value distributions are uniform on $[0,1]$ under their respective null hypotheses H_{0i}, the upper bound becomes $k(\alpha/k) = \alpha$. Since the actual probability of rejecting at least one null hypothesis is less than the nominal FWE level α, the Bonferroni method is conservative.

Very closely related to the Bonferroni method is the Šidák method (Šidák, 1967), which rejects hypothesis H_{0i} when the p-value λ_i is less than $1-(1-\alpha)^{1/k}$, i.e.

$$\tilde{\lambda}_i = 1-(1-\lambda_i)^k.$$

While the Šidák adjustments are usually conservative, they are less conservative than those obtained with the Bonferroni method.

Resampling method

Bonferroni-type methods fail to incorporate dependence and distributional characteristics of the λ_i. Westfall and Young (1993) pointed out for the first time that if we knew the joint distributions of the p-values, we could compute the single-step adjusted p-values as

$$\tilde{\lambda}_i = \mathbb{P}(\min_{1\le j\le k} \Lambda_j \le \lambda_i | H_0),$$

i.e. the proportion of replicated experiments for which the minimum p-value is smaller than the given experimentally observed p-value. Incorporation of the correlation structure results in a smaller adjusted p-value and, consequently, more powerful tests. As an extreme example, consider what would happen if all the p-values were perfectly correlated and uniformly distributed under the complete null hypothesis. In this case:

$$\mathbb{P}(\min_{1\le j\le k} \Lambda_j \le \lambda_j | H_0) = \mathbb{P}(\Lambda_i \le \lambda_i | H_0) = \lambda_i,$$

i.e. no adjustment is needed. This suggests that incorporation of correlation structures can be very important when correlations are extremely large, as is the case for the test studies in this thesis. Sometimes this probability can be computed analytically, but in situations involving dependence among the λ_i, such as the multiple test for the equality of covariance operators, the distribution of $(\min_{1\le j\le k} \Lambda_j | H_0)$ is usually intractable. In the case of interest, however, we have seen in Section 3.2.3 that it is possible to simulate vectors $\boldsymbol{\lambda}^*$ having the same null distribution as the original p-value vector $\boldsymbol{\lambda}$. Then, the adjusted p-values may be approximated by the distribution obtained by taking the minimum of those vectors. A more formal description of this process is given in the following algorithm:

Algorithm 6.1 (Single-step p-value adjustment for generic NPC method)**.**

1. *Perform the NPC Algorithms 3.2 and 3.3. The* $\hat{\boldsymbol{\lambda}}^{(b)}$, $b = 1, \dots, B$ *computed in step 3. of the second phase are vectors of p-values from the same distribution as the actual p-values* $\lambda_1, \dots, \lambda_k$ *under the complete null hypothesis.*

2. *For* $b = 1, \dots, B$ *compute* $p^{(b)} = \min_{1\le j\le k} \lambda_j^{(b)}$*;*

3. *The estimated adjusted p-value is*

$$\hat{\tilde{\lambda}}_i = \frac{\sum_b \mathbb{1}(p^{(b)} \le \lambda_i)}{B}. \tag{6.1}$$

Since the single-step adjusted p-values are all computed under the complete null hypothesis H_0, it is straightforward to prove that the FWEC is controlled. The proof given by Westfall and Young (1993) holds also in this particular case.

Proposition 6.1. *The FWE is controlled in the weak sense by the rule* $\tilde{\lambda}_i \le \alpha$*, where* $\tilde{\lambda}_i$ *are computed as in Algorithm* (6.1).

Proof. If a null hypothesis is rejected when $\tilde{\lambda}_i \leq \alpha$, then

$$\mathbb{P}(\text{Reject at least one } H_{0i}|H_0) = \mathbb{P}(\text{At least one } \tilde{\Lambda}_i \leq \alpha|H_0).$$

By definition, $\tilde{\lambda}_i \leq \alpha$ if and only if $\lambda_i \leq c_\alpha$, where c_α denotes the α quantile of the minimum p-value distribution. Formally, letting $\mathcal{L}$ denote the sample space of the minimum p-value,

$$c_\alpha = \max\{\lambda \in \mathcal{L}|\mathbb{P}(\min_{1\leq j\leq k} \Lambda_j|H_0) \leq \alpha\}.$$

In words, c_α is the largest observable value of the random variable $\min_{1\leq j\leq k} \lambda_j$for which the cumulative probability is at most α. Then we have

$$\mathbb{P}(\min_{1\leq j\leq k} \tilde{\Lambda}_j \leq \alpha|H_0) = \mathbb{P}(\min_{1\leq j\leq k} \Lambda_j \leq c_\alpha|H_0) \leq \alpha.$$

so that this simultaneous test procedure controls the FWEC at (or below) level α. □

It is important to note that, when using permutation approaches to estimate $(\min_{1\leq j\leq k} \Lambda_j|H_0)$, we can only approximate the control of the FWE and the closeness of the approximation depends upon the accuracy of the permutation-based estimates of the underlying distributions. Simulation analysis can be helpful when attempting to evaluate such approximations.

Since all probabilities have been computed under the complete null hypothesis, it might seem that partial error rates are not controlled. However, Westfall and Young (1993) introduced an additional condition that allowed to prove that the FWE is controlled also in the strong sense.

Condition 6.1 (Subset Pivotality)**.** *The distribution of $\boldsymbol{\lambda}$, vector of partial p-values, has the Subset Pivotality (SP) property if the joint distribution of the sub-vector $\{\lambda_i, i \in \bar{K}\}$ is identical under the restrictions $\bigcap_{i\in\bar{K}} H_{0i}$ and H_0, for all subsets $\bar{K} = \{i_1, \ldots, i_j\} \subseteq K$ of true null hypotheses.*

In many cases the analyses considered are approximate in an asymptotic sense, so Westfall and Young (1993) defined a less restrictive condition. Suppose the data **X** is assumed to come from a sequence of experiments involving an increasing amount of data. Indexing the experiments by $t = 1, 2, \ldots$, the p-value for testing H_{0i} is indexed correspondingly as λ_{it} and the entire vector is $\boldsymbol{\lambda}_t$.

Condition 6.2 (Asymptotic Subset Pivotality)**.** *Let $K = \{1, \ldots, k\}$ be the index set for the tested hypotheses H_{0i} and $\bar{K} = \{i_1, \ldots, i_j\}$ any set of true null hypotheses. The distribution of $\boldsymbol{\lambda}_t$ has the asymptotic subset pivotality property if the joint distribution of the sub-vector $\{\lambda_{it}, i \in K\}$ converges to the same limit law (as $t \to \infty$) under the restrictions $\bigcap_{i\in K} H_{0i}$ and H_0.*

While the subset pivotality condition is easily satisfied in many cases, researchers have questioned the assumption. For example, Romano et al. (2008) stated:

> *The condition of subset pivotality assumed in Westfall and Young (1993) is quite restrictive.*

However, Westfall and Troendle (2008) proved that in the permutational tests framework, in which everything will be embedded within the null hypotheses, under H_0, being all observations exchangeable, SP condition is always satisfied.

Then, the following proposition holds.

Proposition 6.2. *Under Condition 6.1, the FWE is controlled in the strong sense by the rule $\tilde{\lambda}_i \leq \alpha$, where $\tilde{\lambda}_i$ are computed as in Algorithm* (6.1).

We give here a special case of the original proof written by Westfall and Young, 1993.

Proof. Suppose that $\bar{K} = \{i_1, \dots, i_j\}$ is the collection of hypotheses H_{0i} which are true. Then

$$\begin{aligned} \mathbb{P}(\text{At least one } H_{0i} \text{ is rejected for } i \in \bar{K} | \cap_{i \in \bar{K}} H_{0i}) \quad &= \mathbb{P}(\min_{i \in \bar{K}} \tilde{\Lambda}_i \leq \alpha | \cap_{i \in K} H_{0i}) \\ &= \mathbb{P}(\min_{i \in \bar{K}} \Lambda_i \leq c_\alpha | \cap_{i \in K} H_{0i}) \\ \text{(SP)} \quad &= \mathbb{P}(\min_{i \in \bar{K}} \Lambda_i \leq c_\alpha | H_0) \\ &\leq \mathbb{P}(\min_{1 \leq i \leq k} \Lambda_i \leq c_\alpha | H_0) \\ &\leq \alpha \end{aligned}$$

□

Again, it is assumed that the adjusted p-values are evaluated without error. If this is not the case, then the method approximately controls the FWE in the strong sense.

6.3 Step-down methods

As mentioned at the beginning of the chapter, it is possible to improve the power of the procedure by making the adjusted p-values uniformly smaller, while retaining the same error rate protection. The idea is that, rather than adjusting all p-values according to the minimum p-value distribution, one should only adjust the minimum p-value using this distribution and then adjust the remaining p-values according to smaller and smaller sets of p-values. The effect of using restricted sets of p-values is to make the adjusted p-values smaller, thereby improving the power of the method.

Holm's sequentially rejective algorithm

Holm (1979) derived a simultaneous test procedure based on the Bonferroni inequality which strongly controls the FWE. Holm's sequentially rejective algorithm is based on the ordered p-values $\lambda_{(1)} \leq \ldots \leq \lambda_{(k)}$, corresponding to the hypotheses $H_{(1)}, \ldots, H_{(k)}$. The intuitive reasoning is as follows: once $H_{(1)}$ has been rejected using the Bonferroni critical value α/k, we should believe that $H_{(1)}$ is false. Thus, there are only $k-1$ hypotheses which might still be true, so that the critical value for for $H_{(2)}$ can be augmented to $\alpha/(k-1)$. If $H_{(2)}$ is rejected, we can use $\alpha/(k-2)$ for $H_{(3)}$, and so on. As in the case of the single-step Bonferroni method, adjusted p-values greater than one must be set to one. While the FWE is strongly protected using this step-down method, it is based on the Bonferroni probability inequality, and therefore is conservative.

Resampling method

Also in this case, the adjustments can be made less conservative by incorporating the precise dependence characteristics, as with the single-step adjusted p-values. Let the ordered p-values have indexes $r_1, \ldots, r_k$ so that $\lambda_{(1)} = \lambda_{r_1}, \lambda_{(2)} = \lambda_{r_2}, \ldots, \lambda_{(k)} = \lambda_{r_k}$. The step-down adjusted p-values are defined sequentially as follows:

$$\tilde{\lambda}_{(1)} = \mathbb{P}(\min_{j \in \{r_1, \ldots, r_k\}} \Lambda_j \leq \lambda_{(1)} | H_0)$$

$$\tilde{\lambda}_{(i)} = \max\{\tilde{\lambda}_{(i-1)}, \mathbb{P}(\min_{j \in \{r_i, \ldots, r_k\}} \Lambda_j \leq \lambda_{(j)} | H_0)\}, \quad i = 2, \ldots, k \tag{6.2}$$

The use of max operator insures that the order of the adjusted p-values is the same as that of the original p-values. The adjustments (6.2) are uniformly smaller than the single-step adjusted p-values, since the minima are taken over successively smaller sets. Westfall and Young (1993) proved that this procedure too, controls the FWE in the strong sense. We retrieve here the main ideas of the proof because they are the key to understand the methods that follow.

Proposition 6.3. *Under Condition 6.1, the FWE is controlled in the strong sense by the rule $\tilde{\lambda}_i \leq \alpha$, where $\tilde{\lambda}_i$ are computed as in* (6.2).

Proof. Let $K = \{1, \ldots, k\}$ be the index set for the tested hypotheses H_{0i}. To control the FWE in the strong sense, it is required that the probability of rejecting at least one true H_{0i} is no larger than α, no matter what subset of the $K_0 \subseteq K$ of hypotheses happen to be true. Suppose that $K_0 \neq \varnothing$ (if $K_0 = \varnothing$ then there can be no type I errors). Let $c_\alpha^{\bar{K}}$ denote the α quantile of

$(\min_{j\in\bar{K}}\Lambda_j|H_0)$:

$$c_\alpha^{\bar{K}} = \max\{\lambda \in S|\mathbb{P}(\min_{j\in\bar{K}}\Lambda_j|H_0) \le \alpha\}.$$

Define $K_i = \{r_i,\dots,r_k\},\ i = 1,\dots,k$. We have the following relationships:

$$\{\text{There is at least one type I error}\} = \{\text{Reject at least one } H_{0i}, i \in K_0\} \subseteq \{\min_{i\in K_0}\Lambda_i \le c_\alpha^{K_j}\}$$

where $j \le k-|K|+1$ is defined by $\min_{i\in K_0}\lambda_i = \lambda_{(j)} = \lambda_{r_j}$, i.e. it is the true hypothesis with the smallest p-value. Then

$$\begin{aligned}
\mathbb{P}(\text{There is at least one type I error}) &= \mathbb{P}(\text{Reject at least one } H_{0i}, i \in K_0|\cap_{i\in K_0} H_{0i})\\
&\le \mathbb{P}(\min_{i\in K_0}\Lambda_i \le c_\alpha^{K_j}|\cap_{i\in K_0} H_{0i})\\
(\text{SP}) \quad &= \mathbb{P}(\min_{i\in K_0}\Lambda_i \le c_\alpha^{K_j}|H_0).
\end{aligned}$$

Since $K_j \subseteq K$, we have $c_\alpha^{K_j} \le c_\alpha^K$, implying

$$\mathbb{P}(\min_{i\in K_0}\Lambda_i \le c_\alpha^{K_j}|H_0) \le \mathbb{P}(\min_{i\in K_0}\Lambda_i \le c_\alpha^{K_0}|H_0) \le \alpha$$

that is, the step-down adjustments strongly control the FWE. □

As in Section 6.2, it is assumed once again that the adjusted p-values are evaluated without error. When this is not true, then the resampling method approximately controls the FWE in the strong sense.

In view of this considerations, it is easy to see that this method is equivalent to iteratively use the NPC with the Tippett combining function, as explained in Pesarin and Salmaso (2010):

Algorithm 6.2 (Step-down method for Tippett combining function)**.**

Let $\lambda_{(1)},\dots,\lambda_{(k)}$ be the increasing ordered p-values corresponding to the set of minimal hypotheses.

1. $\tilde{\lambda}_{(1)} = \lambda_{(1),\dots,(k),Tippett}$
   - *If $\tilde{\lambda}_{(1)} \le \alpha$, reject the corresponding hypothesis $H_{0(1)}$ and continue;*
   - *Otherwise retain the hypotheses $H_{0(1)},\dots,H_{0(k)}$ and stop.*
2. $\tilde{\lambda}_{(i)} = \max\{\lambda_{(i),\dots,(k),Tippett}, \tilde{\lambda}_{(i-1)}\}$
   - *If $\tilde{\lambda}_{(i)} \le \alpha$, reject also $H_{0(i)}$ and continue;*

- *Otherwise retain the hypotheses $H_{0(i)}, \dots, H_{0(k)}$ and stop.*

Furthermore, Lehmann and Romano (2005) presented a more generic step-down method. The basic idea is to use the test statistics T instead of the p-values. So, for example, suppose that the individual tests H_{0i} are based on test statistics T_i with large values indicating evidence against the partial null hypotheses. The method is equivalent to that based on the Tippett combining function but allows to avoid the computations of the of the permutational distributions of the partial p-values.

Algorithm 6.3 (Step-down method for $\max T$ combining function)**.**

Let $T_{r_1} \geq T_{r_2} \geq \ldots \geq T_{r_k}$ denote the observed ordered test statistics and let $H_{(1)}, H_{(2)}, \dots, H_{(k)}$ be the corresponding hypotheses.

1. *Let $K = \{1, \dots, k\}$,*
   - *If $T_{r_1} \geq c_K(\alpha)$ reject $H_{(1)}$ and continue;*
   - *Otherwise retain the hypotheses $H_{0(1)}, \dots, H_{0(k)}$ and stop.*
2. *For $i = 2, \dots, k$, let K_i be the set of hypotheses not previously rejected,*
   - *If $T_{r_i} \geq c_{K_i}(\alpha)$ reject $H_{(i)}$ and continue;*
   - *Otherwise retain the hypotheses $H_{0(i)}, \dots, H_{0(k)}$ and stop.*

The problem is how to construct the $c_{\bar{K}}(\alpha)$ so that the FWE is controlled in the strong sense. The following holds (Lehmann and Romano, 2005):

Proposition 6.4. *Consider Procedure 5.2. The following two conditions are sufficient for controlling the FWE at level α:*

1. *monotonicity of the critical values: for any $\bar{K} \supset K_0$, $c_{\bar{K}}(\alpha) \geq c_{K_0}(\alpha)$;*
2. *weak control of the family-wise error rate at each step, i.e., when $\bar{K}$ is true,*

$$\mathbb{P}\{\max_{i \in \bar{K}}(T_i) \geq c_{\bar{K}}\} \leq \alpha.$$

In Solari et al. (2009) the critical value $c_{\bar{K}}(\alpha)$ is defined as the smallest qth value among the permutation distributions of $T_{\bar{K}} = \max_{i \in \bar{K}} T_i$,

$$c_{\bar{K}}(\alpha) = \{\max_{i \in \bar{K}} T_i^*(b), b = 1, \dots, B\}$$

with $q = B - \lfloor B\alpha \rfloor$. For this reason, in the following we will refer to this as the step-down method for the max T combining function. The so-defined critical values clearly satisfy the monotonicity requirement. In order to satisfy also the second requirement of Proposition 6.4, the considered model must guarantee the randomisation hypothesis. That is, under the null hypothesis, the distribution of the subset of data that is used for the calculation of $T_i, i \in \bar{K}$ must not be affected by the transformations considered (here, permutations). Thus we require exchangeability of the observed data.

6.4 Closed testing procedure

In case one wants to use another combining function and still perform post-hoc comparisons, a possibility is to use the closed testing procedure (Marcus et al., 1976). The idea behind the closure method is that one may reject any hypothesis H_{0i}, while controlling the FWE, when the test of H_i itself is significant and the test of every intersection hypothesis that includes H_{0i} is significant. Hence,

$$\tilde{\lambda}_i = \max(\lambda_i, \lambda_{ij}, \lambda_{ijh}, \ldots).$$

When using permutation tests, the algorithm to be used is the following:

Algorithm 6.4 (Closed testing for permutation tests)**.**

Consider the closure of the set, which is the set of all possible intersection hypotheses.

1. *Test all the hypotheses simultaneously by using permutation tests:*
   (a) *Calculate the statistics $T_{\bar{K}}$ for each non-empty $\bar{K} \subseteq \{1, \ldots, k\}$;*
   (b) *For $b = 1, \ldots, B$, perform the* b*th permutation and compute the statistics $T^{*(b)}_{\bar{K}}$ for each non-empty $\bar{K} \subseteq \{1, \ldots k\}$ on the* b*th permutation of the data;*
   (c) *Calculate the raw p-values as*

$$\lambda_{\bar{K}} = \frac{\sum_b \mathbb{1}(T^*_{\bar{K}}(b) \geq T_{\bar{K}})}{B};$$

2. *Reject any hypothesis H_{0i} when the test of H_{0i} itself is significant and the test of every intersection hypothesis that includes H_{0i} is significant.*

This method has two major drawbacks: it requires a greater number of computations and it is very conservative. However, it has proved useful in many cases when the use of the Tippett or max T combining functions is not suitable.

Chapter 7

Simulation studies

This chapter contains all the simulation studies that have been performed in order to assess the empirical power of the proposed methodology. In Section 7.1 are explained the two considered case studies. Section 7.2 is dedicated to the comparisons between the permutation strategies illustrated in Chapter 5. In Section 7.3, simulations are used in order to quantify the loss of empirical power due to the adjustment of partial p-values in order to control the FWE. In Section 7.4, we illustrate how the different distances introduced in Chapter 2 behave, when applied to the considered case studies. We will explain why we have chosen the square root distance and the max T combining function for the analysis of the wheel-running dataset. In Section 7.5 the test is applied to non-Gaussian data. The reported results show that the test performs well also in this case, because the assumption of gaussianity is not required. Because of the computational effort required in order to assess the empirical power of these tests, all simulations have been performed in a three-sample setting. However, in Section 7.6 we also show the empirical power of the test in some cases in which the number of sample groups is greater than three. Lastly, in Section 7.7, the proposed permutation test is compared, when possible, to the other tests that can be used for the problem of interest, presented in Chapter 4.

7.1 Case studies

The covariance operators used for the case studies are the same as in Pigoli et al. (2014). In particular, we select Σ_1 and Σ_2 as in Section 5.3, that is, equal to the sample covariance operators for the male and female subjects, respectively, of the Berkeley growth study dataset (Ramsay and Silverman, 2005), rescaled to lie in $[0,1]$. In most of the simulations reported in this chapter, synthetic data have been generated as follows: the first data sample has covariance operator Σ_1, the others have covariance equal to $\Sigma(\gamma)$. $\Sigma(\gamma)$ depends on Σ_1 and Σ_2

and varies according to the test case. The mean function is $\sin(x)$ for all the samples. In the first test case we consider

$$\Sigma(\gamma) = [(\Sigma_1)^{1/2} + \gamma\{(\Sigma_2)^{1/2}\tilde{R} - (\Sigma_1)^{1/2}\}][(\Sigma_1)^{1/2} + \gamma\{(\Sigma_2)^{1/2}\tilde{R} - (\Sigma_1)^{1/2}\}]'$$

where $\tilde{R}$ is the operator minimizing the Procrustes distance between Σ_1 and Σ_2 (Pigoli et al., 2014). In the second test case the covariance are equal up to a scaling factor:

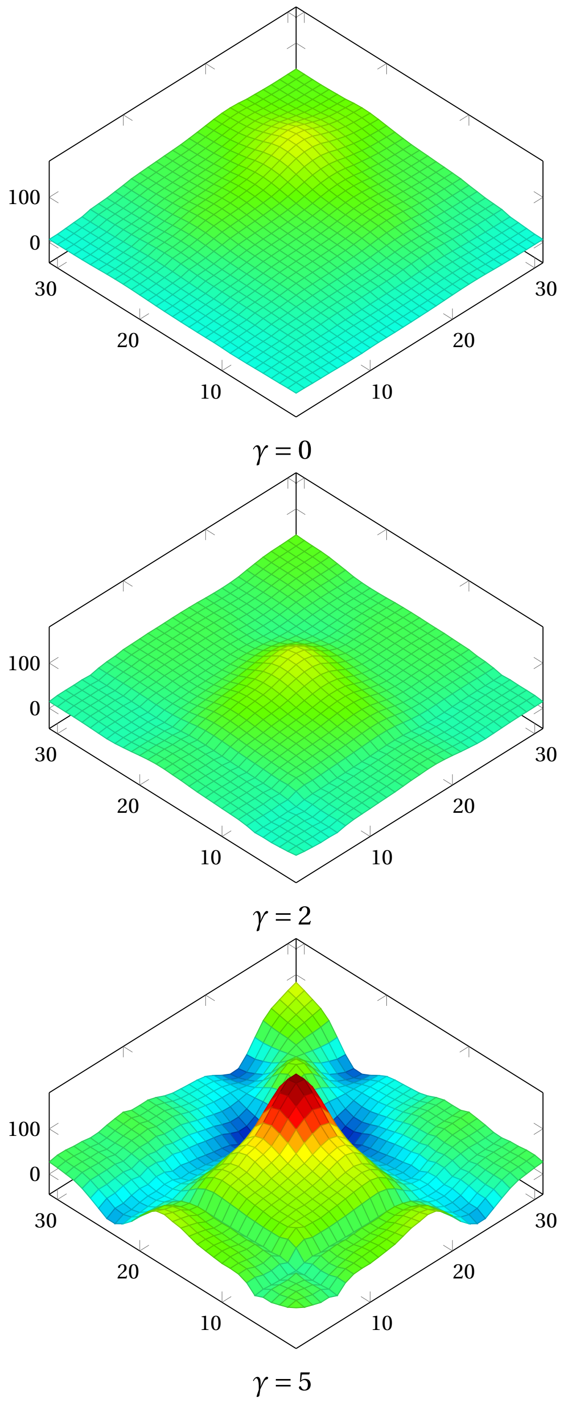
$$\Sigma(\gamma) = (1+\gamma)\Sigma_1.$$

Unless otherwise stated, the curves are simulated from a multivariate Gaussian process, generated on a grid of $p = 31$ points and the sample size of each group is $N = 20$. Each permutation test is performed with $B = 1000$ iterations of the MC Algorithm 5.2 and is repeated for 1000 different synthetic data samples.

7.2 Empirical power of synchronised, paired and pooled tests

The aim of this section is to evaluate the empirical power of the proposed test, when using the three permutation strategies introduced in Section 5.3: pooled, paired and synchronised. Figures 7.3 and 7.4 show the results obtained with the direct and $\max T$ combining functions respectively, for both case studies and three of the considered distances between covariance operators: square root, Procrustes and L^2 distance between the kernels (see Section 2.2.2). We have chosen not to take into account the spectral distance since, as it has already been pointed out by Pigoli et al. (2014), it is not suitable in the second test case, because it cannot discern between an operator and one of its multiples. The values of the empirical levels of the tests are reported in Table 7.1 for a more detailed analysis.

We can see that the global test is anti-conservative when paired permutations are applied, because the dependence among partial tests is not taken into account. Pooled and paired permutations, instead, lead to the right empirical level and have approximately the same empirical power in all cases. However, in the balanced case, we prefer to use synchronised permutations, because they allow for straightforward post-hoc comparisons.

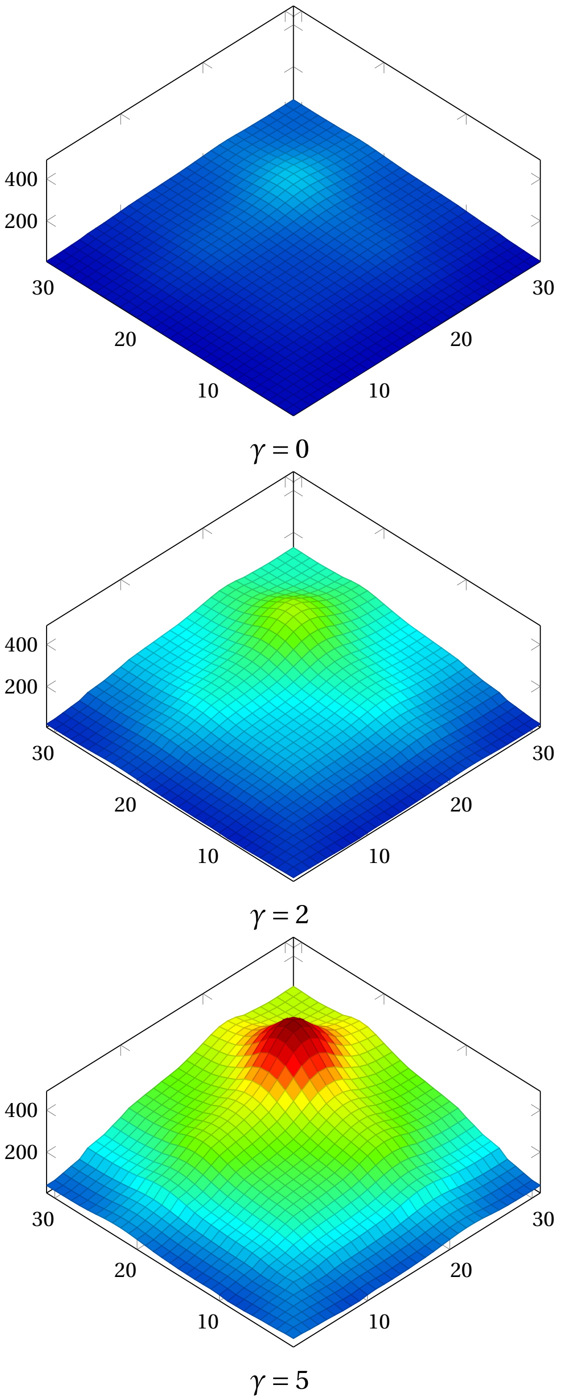


$\gamma = 0$

$\gamma = 2$

$\gamma = 5$

Figure 7.1: Covariance operators, test case 1.

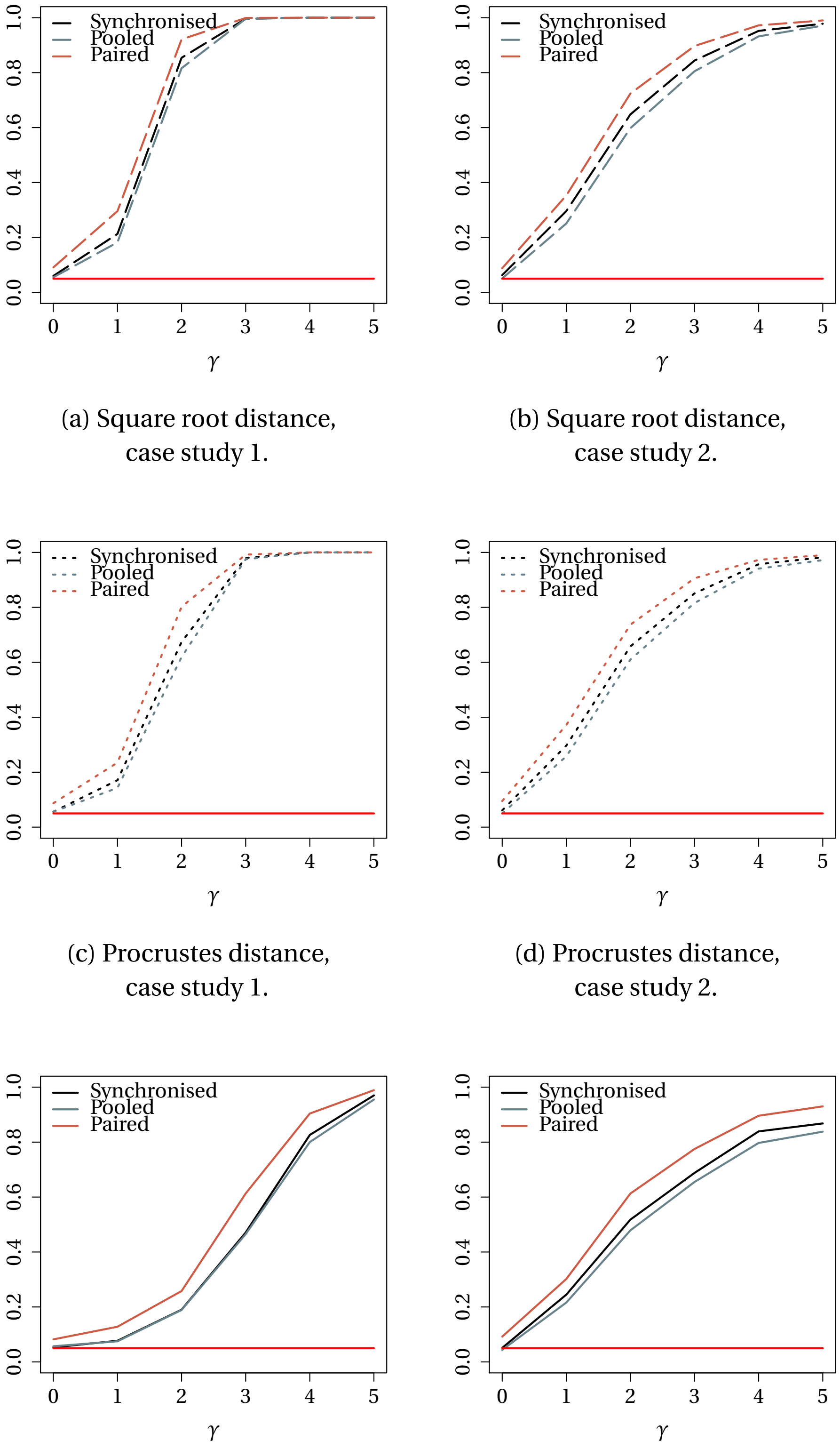


$\gamma = 0$

$\gamma = 2$

$\gamma = 5$

Figure 7.2: Covariance operators, test case 2.



(a) Square root distance, case study 1.

(b) Square root distance, case study 2.

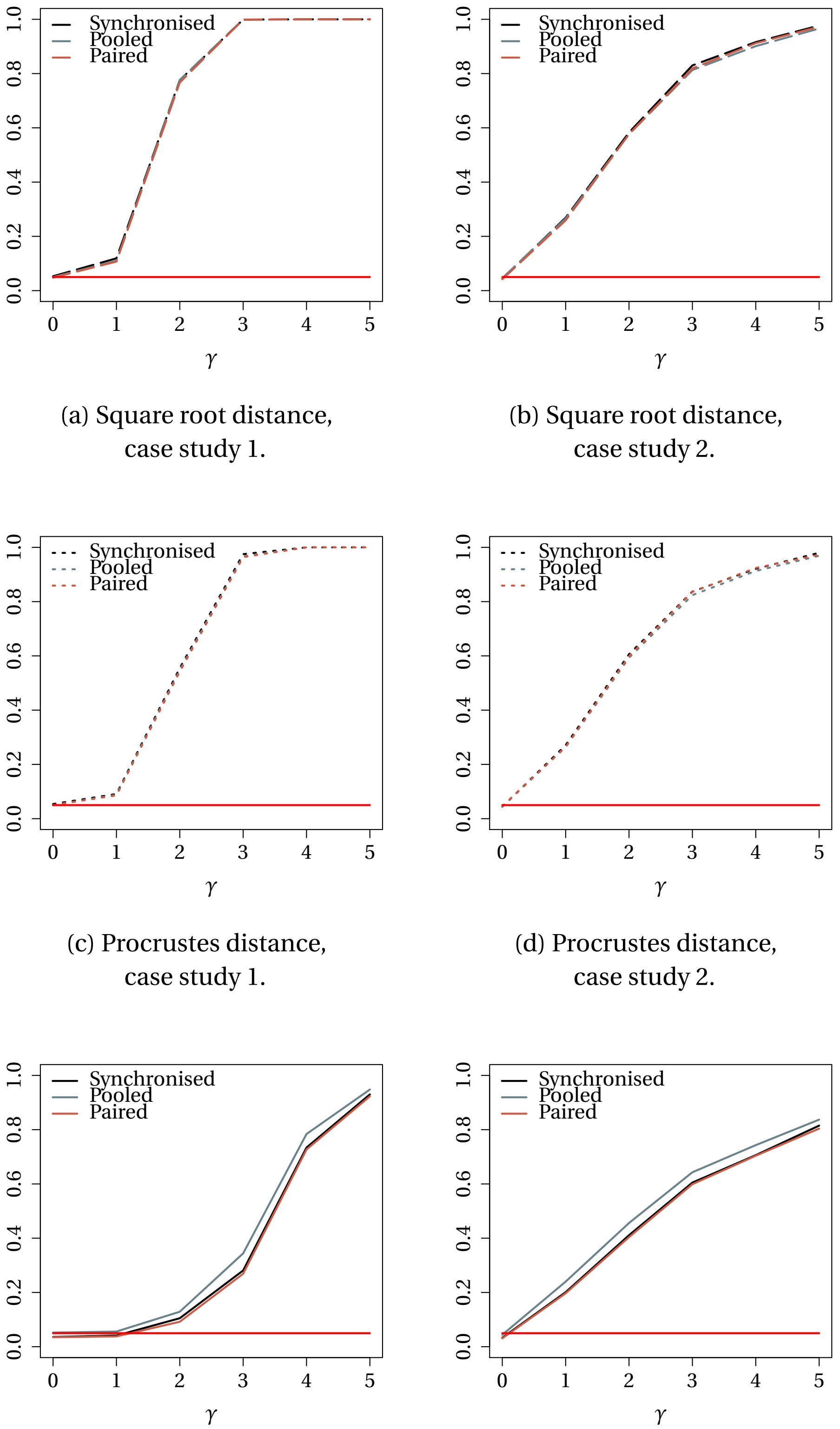
(c) Procrustes distance, case study 1.

(d) Procrustes distance, case study 2.

(e) Kernel L^2 distance, case study 1.

(f) Kernel L^2 distance, case study 2.

Figure 7.3: Comparison between pooled, paired and synchronised permutations. Direct combining function, case studies 1 and 2 (first and second row respectively).



(a) Square root distance, case study 1.

(b) Square root distance, case study 2.

(c) Procrustes distance, case study 1.

(d) Procrustes distance, case study 2.

(e) Kernel L^2 distance, case study 1.

(f) Kernel L^2 distance, case study 2.

Figure 7.4: Comparison between pooled, paired and synchronised permutations. $\max T$ combining function, case studies 1 and 2 (first and second row respectively).

Table 7.1: Rejection rates of $H_{0i}, i = 1,2,3$ and H_0, $\gamma = 0$, direct combining function.

Test	α	Distance	Case study 1	Case study 2
Paired	0.05	Procrustes	0.091	0.095
Paired	0.05	Square root	0.090	0.088
Paired	0.05	Kernel L^2	0.084	0.092
Pooled	0.05	Procrustes	0.051	0.050
Pooled	0.05	Square root	0.051	0.051
Pooled	0.05	Kernel L^2	0.058	0.044
Synchronised	0.05	Procrustes	0.064	0.061
Synchronised	0.05	Square root	0.061	0.063
Synchronised	0.05	Kernel L^2	0.058	0.051

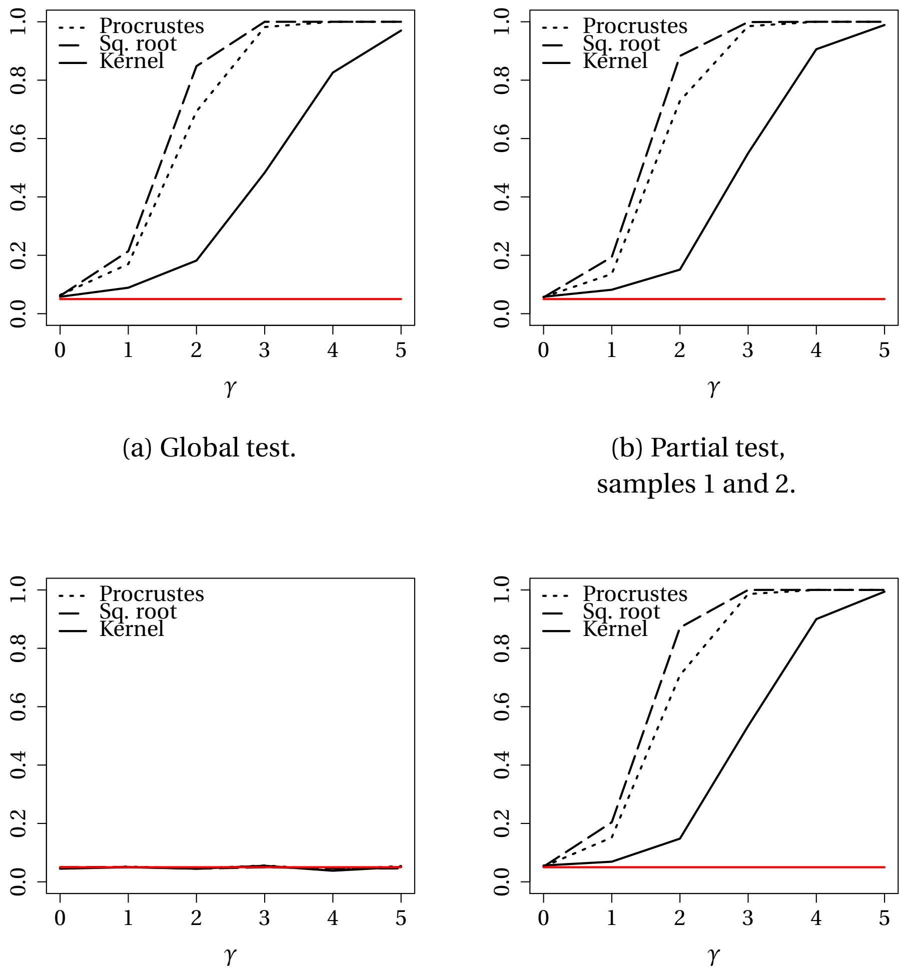
7.3 p-values adjustment

Having chosen to use synchronised permutations, here we show the empirical power of the partial tests with and without adjustment. In particular, Figures 7.5 and 7.6 show the empirical power of the global and partial tests done using the synchronised permutations and direct combining function, for the first and second case studies respectively. In Figures 7.7 and 7.8 are reported the same tests, where the partial p-values have been adjusted with the closed testing procedure (see Section 6.4). For Figures 7.9 and 7.10 the max T combining function has been used, coupled with the ad-hoc step-down procedure for the strong control of the FWE (see Section 6.3). The graphs of the analogous with unadjusted partial tests are not shown because the values of the global test are the same as in Figures 7.9a (7.10a) and the empirical power of the partial tests is that of Figures 7.5b, 7.5c and 7.5d (7.6b, 7.6c and 7.6d) for the first (second) case study.

It is immediate to see that, while unadjusted partial tests for multiple comparisons achieve the nominal level α, the methods of FWE control make them very conservative. Without the adjustment, the empirical power of the partial tests is, of course, the same as the pairwise permutations tests reported by Pigoli et al. (2014). When corrected by the closed testing procedure, the empirical power of the partial tests declines slightly. On the contrary, if the step-down procedure is used, the empirical power remains almost unchanged.

7.4 Choice of the distance and the combining function

Concerning the choice of the distances, it is evident that the test has greater empirical power when using Procrustes and square root distances. For the analysis of the wheel-running dataset, we have chosen to use the second because not only it performs, in some cases, better



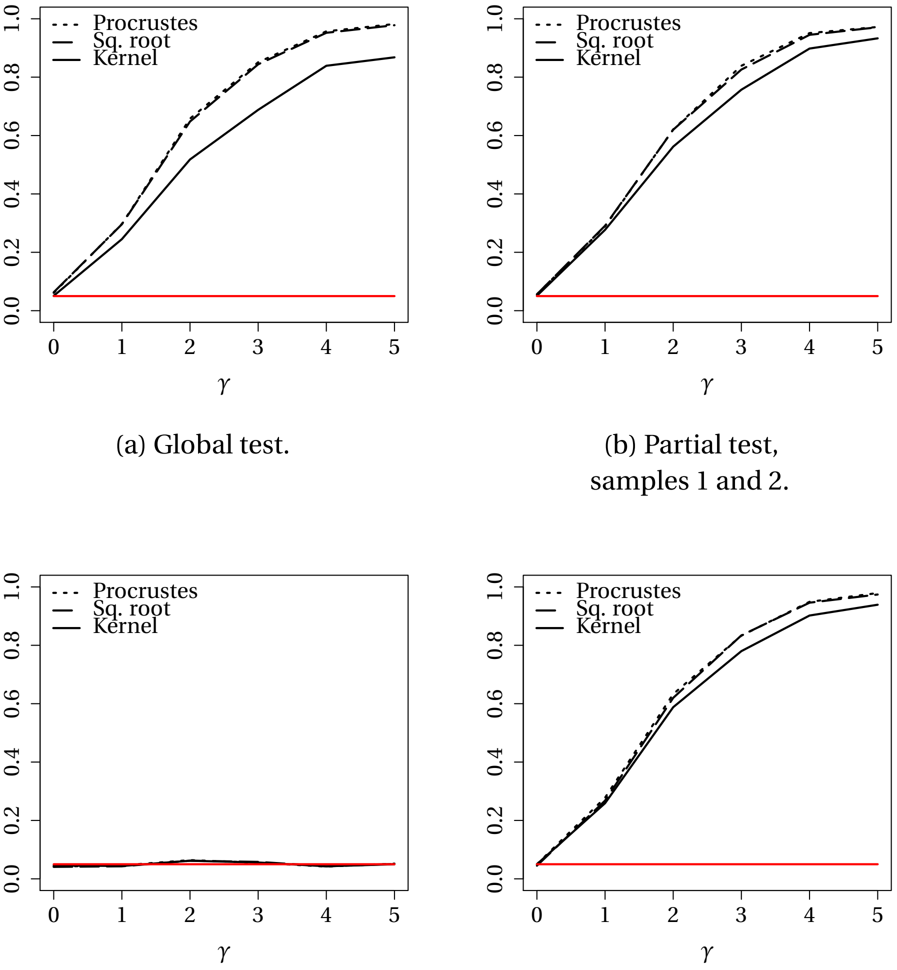
(a) Global test.

(b) Partial test, samples 1 and 2.

(c) Partial test, samples 2 and 3.

(d) Partial test, samples 1 and 3.

Figure 7.5: Empirical power of synchronised permutation global and partial tests applied to the first case study using direct combining function.



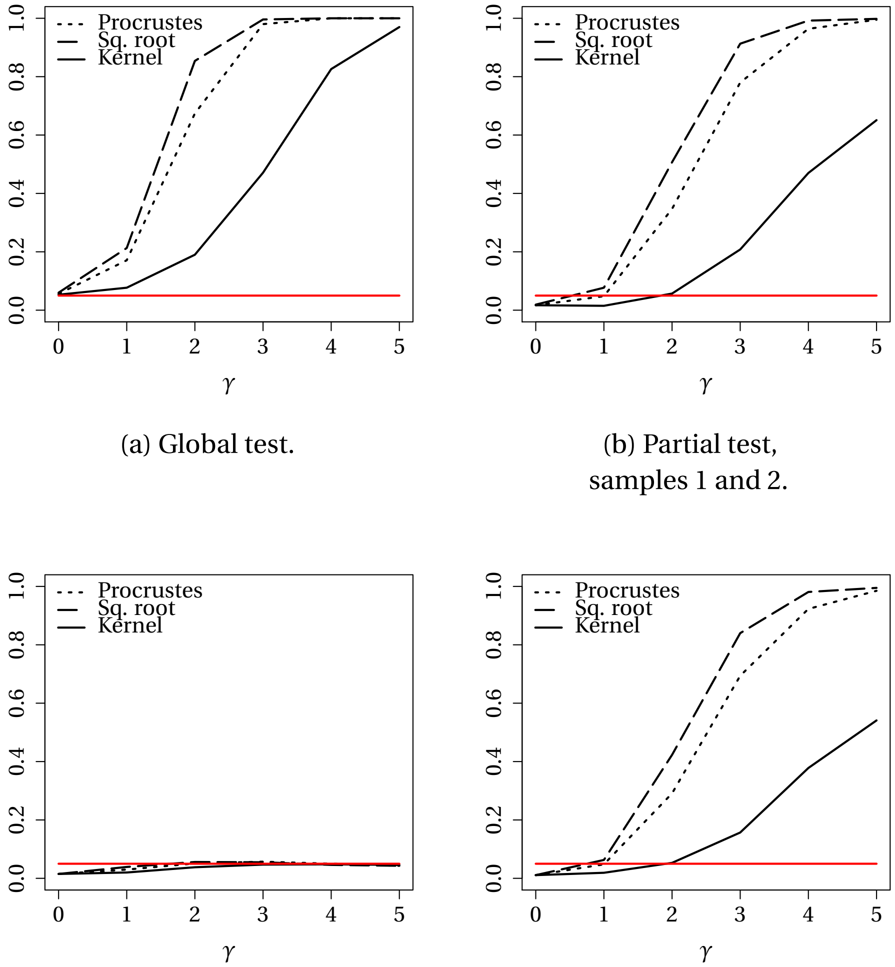
(a) Global test.

(b) Partial test, samples 1 and 2.

(c) Partial test, samples 2 and 3.

(d) Partial test, samples 1 and 3.

Figure 7.6: Empirical power of synchronised permutation global and partial tests applied to the second case study using direct combining function.



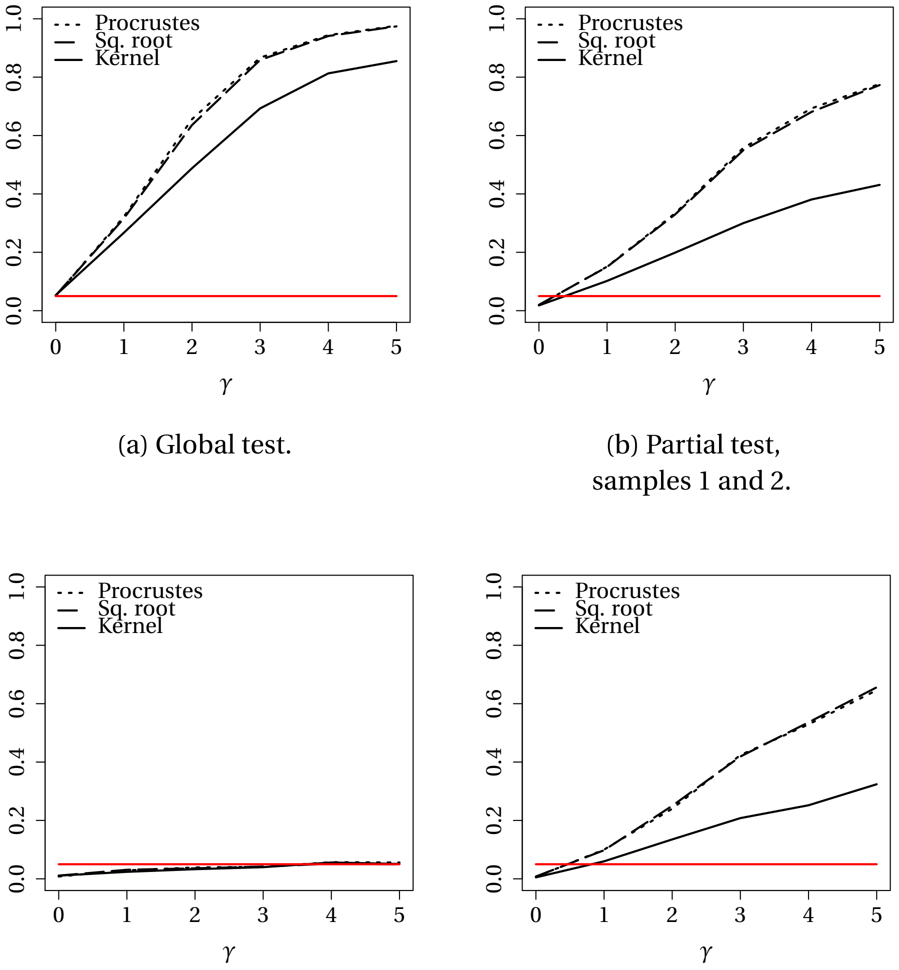
(a) Global test.

(b) Partial test, samples 1 and 2.

(c) Partial test, samples 2 and 3.

(d) Partial test, samples 1 and 3.

Figure 7.7: Empirical power of synchronised permutation global and partial tests applied to the first test case using direct combining function. P-values have been adjusted using the closed testing procedure.



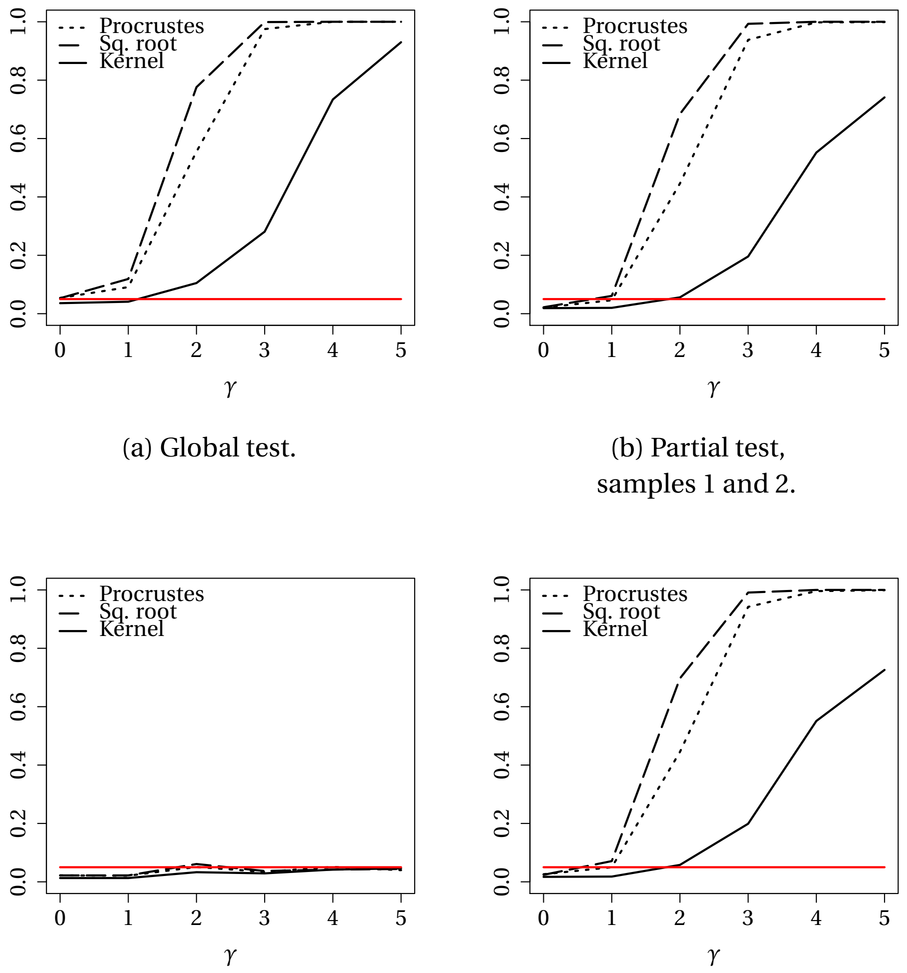
(a) Global test.

(b) Partial test,
samples 1 and 2.

(c) Partial test,
samples 2 and 3.

(d) Partial test,
samples 1 and 3.

Figure 7.8: Empirical power of synchronised permutation global and partial tests applied to the second test case using direct combining function. P-values have been adjusted using the closed testing procedure.



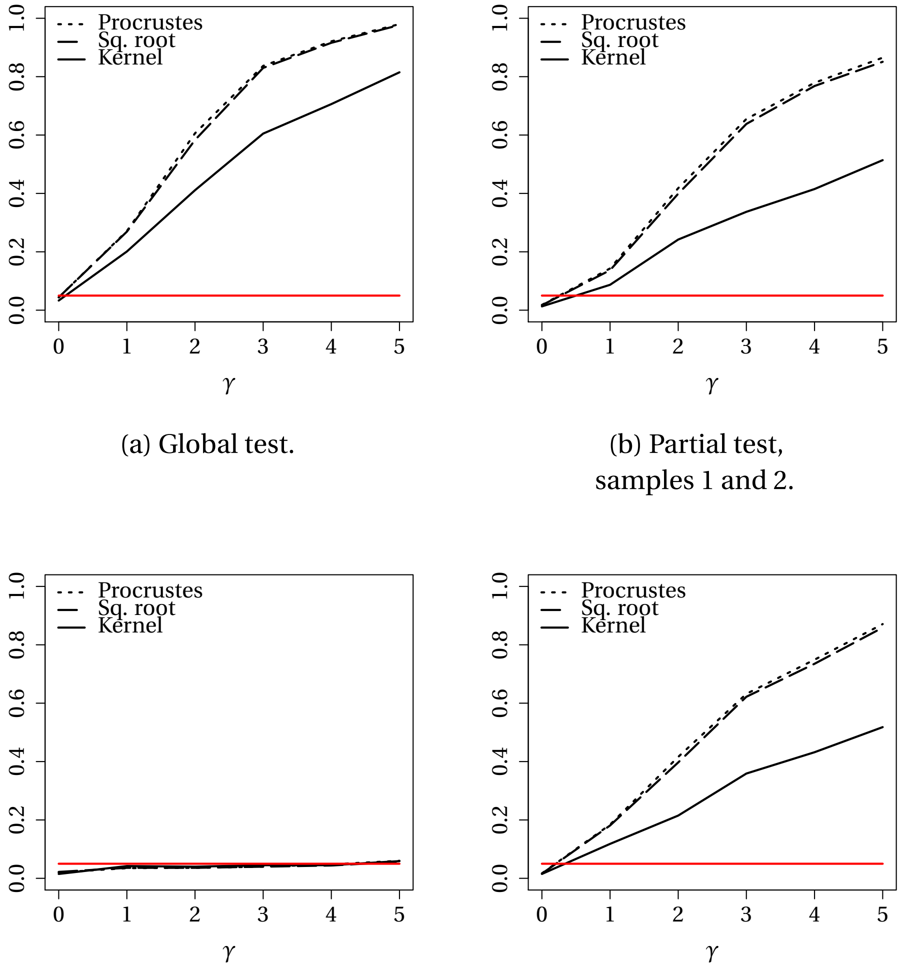
(a) Global test.

(b) Partial test, samples 1 and 2.

(c) Partial test, samples 2 and 3.

(d) Partial test, samples 1 and 3.

Figure 7.9: Empirical power of synchronised permutation global and partial tests applied to the first test case using $\max T$ combining function. P-values have been adjusted using the $\max T$ step-down procedure.



(a) Global test.

(b) Partial test, samples 1 and 2.

(c) Partial test, samples 2 and 3.

(d) Partial test, samples 1 and 3.

Figure 7.10: Empirical power of synchronised permutation global and partial tests applied to the first test case using max T combining function. P-values have been adjusted using the max T step-down procedure.

than the others, but it also has a lower computational cost. As regards the combining function, the choice should be based mainly on two aspects: the synergy of the partial tests that best suits the considered application and the computational effort required to compute the partial p-values $\lambda_1,\dots,\lambda_k$ and then to adjust them to control the FWE, obtaining $\tilde{\lambda}_1,\dots,\tilde{\lambda}_k$. For reasons of time, having to repeat the test thousands of times, in this chapter we have chosen to use only the combining functions that do not require to compute the empirical distributions of the partial p-values by means of the $\lambda_i^{(b)}$ (see Algorithm 3.3, step 3.), i.e. the max T and the direct combining functions. Between these two, the first requires a lower number of computations to adjust the partial p-values, because an ad-hoc step-down procedure is available. For this reason we chose to use this combining function also in the following. However, in different applications, one may be interested in studying different combinations of the partial tests and therefore use other functions. Depending on the size of the dataset, this may take from a few seconds up to a minute more, which in some cases are worth spending for a more accurate analysis.

7.5 Non-Gaussian data

All the simulations reported so far in this chapter have been done with synthetic data drawn from Gaussian processes. In this section we reproduce some of the simulations reported above, with data simulated from a multivariate t-Student with 4 degrees of freedom. In particular, Figures 7.11 and 7.12 are exactly the same as Figures 7.9 and 7.10, except for the data generating process.

The empirical power of the test in both cases is almost equal to the previous case. Indeed, the assumption of gaussianity is not needed in order to perform the test.

7.6 Simulations with more than three groups

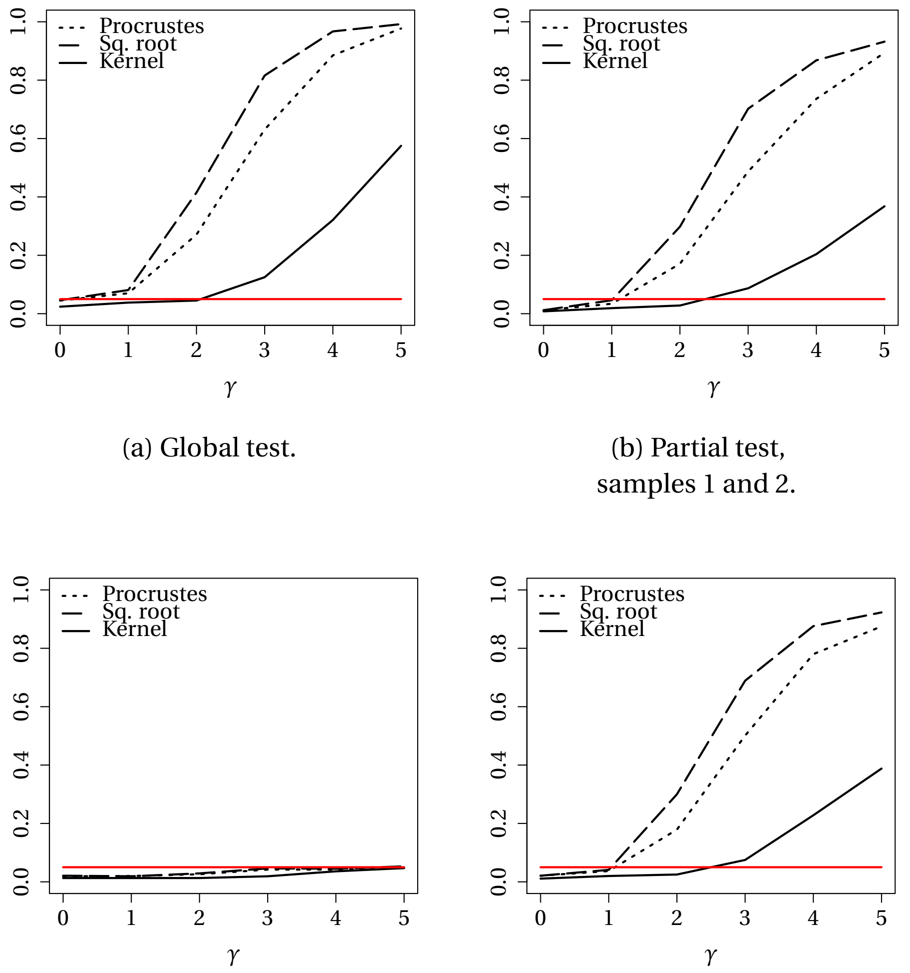
Figure 7.13 shows the empirical power of the global test when the number of groups is equal to 4, 6 and 8. The data samples are simulated as explained in Section 7.1. Figure 7.13a is related to synthetic data samples that have one group with fixed covariance operator Σ_1 and all the others have covariance equal to $\Sigma(\gamma)$. Figure 7.13b, instead, corresponds to the case of two groups of data samples with the same cardinality: one with covariance Σ_1 and the other with covariance $\Sigma(\gamma)$. As usual, γ varies from 0 to 5.

The graphics shows that the test has a great empirical power even for number of groups greater than three. In the first one, it decreases when the number of data samples grows, while the other one displays the opposite trend. In both cases, however, performances change slowly with respect to the number of groups. For this reason, we can suppose that they are almost the same for any reasonable number of data samples.

7.7 Comparison with similar tests

In this section we compare the permutation test to two of the other existing tests introduced in Chapter 4. Figure 7.14 shows the empirical power of the Levene's test and the empirical bootstrap for the two case studies. In the first row they are compared to the results obtained using the permutation test with the max T combining function, in the second row with the direct combining function. The test statistic used for the empirical bootstrap is the same as the one used for the permutation test (see Appendix A). Levene's test has been implemented using the permutation ANOVA. We have decided not to consider here the test of Boente et al. (2014) because its implementation would require to compute the infinite eigenvalues of the operator Υ (see Section 4.2) and is therefore challenging.

It is clear that the empirical bootstrap has approximately the same power as the permutation test. On the contrary, Levene's test performs very differently. As expected, it outperforms in the second case study, where it captures very well the differences in scale, but it does not compete with the others in the other cases. The non-asymptotic test of Kashlak et al. (2016) is much faster than the others and has almost the same power as the resampling-based methods. Thus, if data are supposed to be Gaussian, it is a valid alternative to them. Unfortunately, being based on the Gaussian assumption, it has a limited range of applications. However, the permutation test the only one that allows for straightforward computation of the partial p-values.



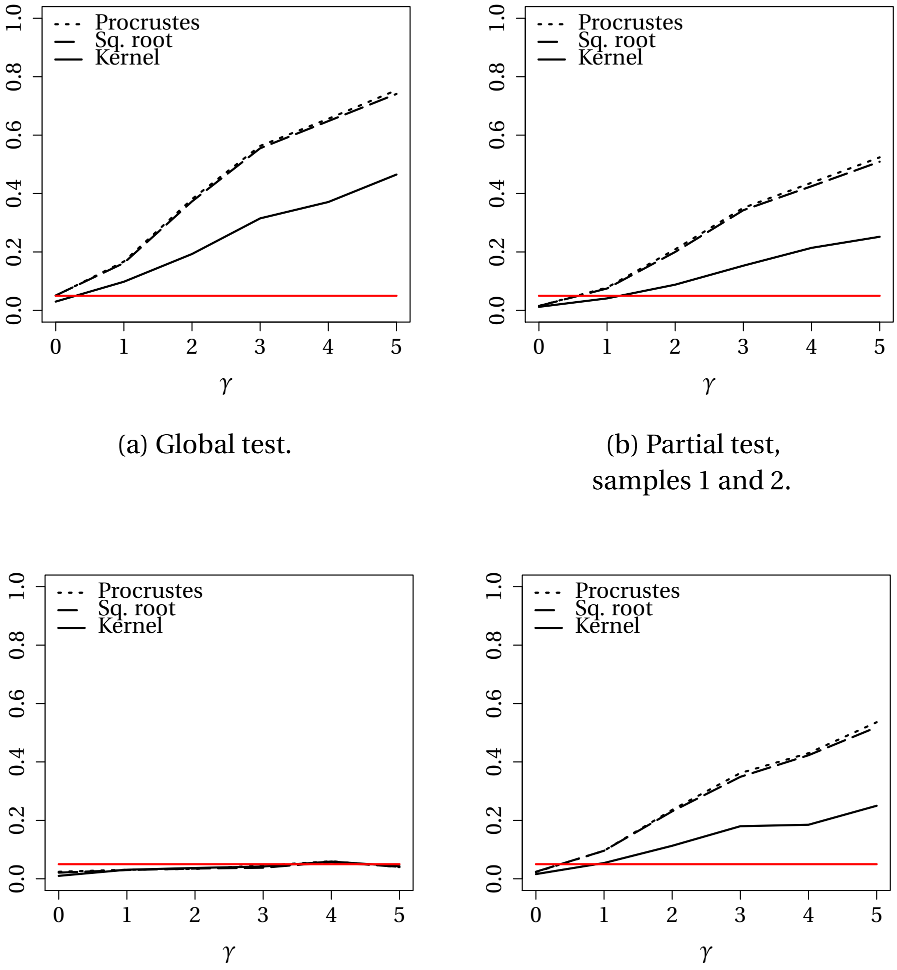
(a) Global test.

(b) Partial test, samples 1 and 2.

(c) Partial test, samples 2 and 3.

(d) Partial test, samples 1 and 3.

Figure 7.11: Synchronised permutation global and partial tests applied to the first case study using max T combining function. P-values have been adjusted using the max T step-down procedure.



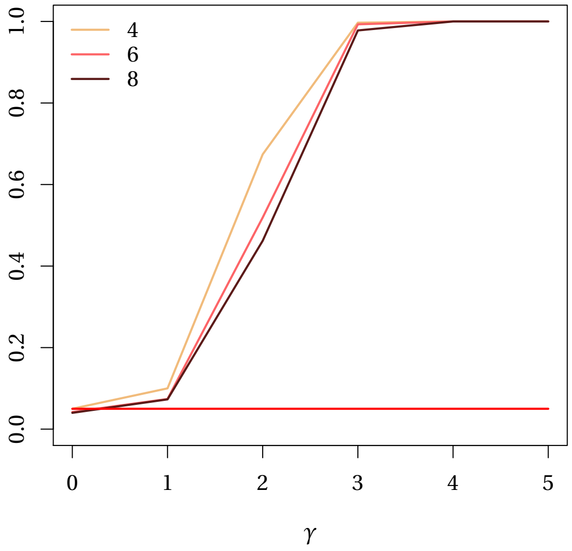
(a) Global test.

(b) Partial test, samples 1 and 2.

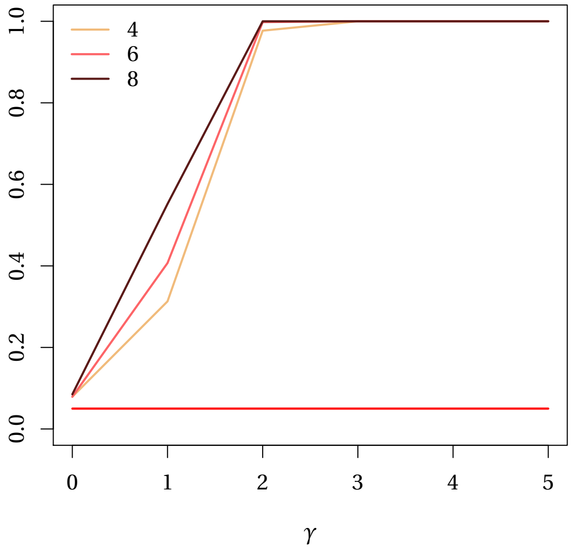
(c) Partial test, samples 2 and 3.

(d) Partial test, samples 1 and 3.

Figure 7.12: Synchronised permutation global and partial tests applied to the first case study using max T combining function. P-values have been adjusted using the max T step-down procedure.

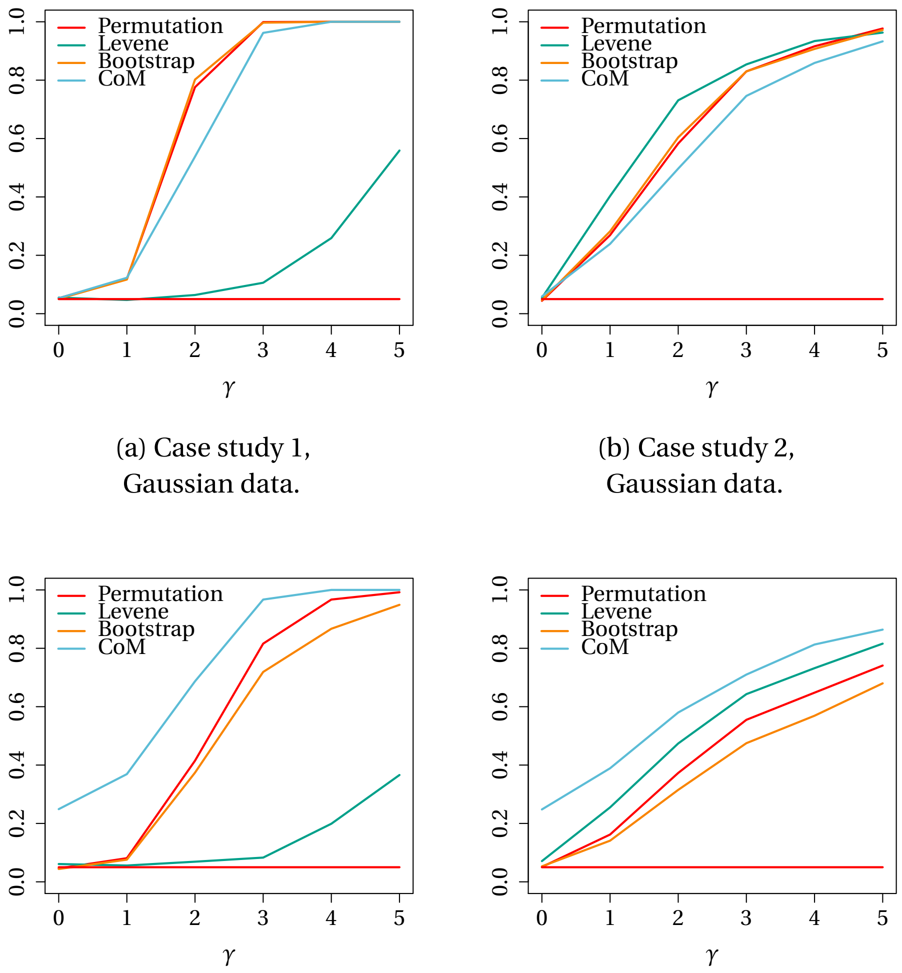


(a) One of the groups has a covariance operator different from all the others.



(b) Half of the groups have covariance equal to Σ_1, the others have covariance $\Sigma(\gamma)$.

Figure 7.13: Synchronised permutation global tests applied to the first case study using max T combining function, with an increasing number of data samples.



(a) Case study 1,
Gaussian data.

(b) Case study 2,
Gaussian data.

(c) Case study 1,
non-Gaussian data.

(d) Case study 2,
non-Gaussian.

Figure 7.14: Synchronised permutation (red), Levene's generalisation (blue), empirical bootstrap (orange) and concentration measure (green) global tests applied to the first (left) and second (right) case studies. In the first row are shown the results given by these methods when synthetic data are generated from a Gaussian process, the second row shows the values obtained when data are drawn from a multivariate t-Student. The combining function is $\max T$ and the p-values have been adjusted with the step-down procedure of Algorithm 6.3.

Chapter 8

Application to evolutionary biology

In this chapter we analyse the wheel-running activity dataset presented in Section 1.3. The objective of the experiment was to use selective breeding to study the genetics and evolution of locomotor behaviour in mice. Four replicate lines of laboratory house mice have been compared to other four random-bred lines maintained as controls. The selection criterion was the total number of revolutions run on days 5 and 6 of a 6-day test. In the selected lines, the highest-running male and female from each family were chosen as breeders. In Section 8.2 we use the proposed permutation methodology to test the equality of the covariance operators of the eight lines of mice used for the experiment, where each function represents the level of voluntary wheel-running of one mice.

Before that, however, we have to deal with the missing observation. In fact, we have seen in Chapter 5 that, ideally, one would want to have a balanced dataset, i.e. with the same number of observations in each sample, so that synchronised permutations can be used. In the wheel-running activity dataset, all groups are composed by 20 mice, but one of them died of unknown causes during the early stages of the experiment. For this reason, one group only has 19 observations and, in order to apply the synchronised permutations, we have to prove that the presence of a missing observation does not affect the inference.

8.1 Dealing with the missing observation

First, we introduce some basic concepts of the theory of missing data and make some preliminary assumptions. Then, following the guidelines given by Pesarin and Salmaso (2010), we give a new formulation of the test that takes into account the presence of missing data. Thanks to this, we are able to prove that it is possible to apply the proposed test to an unbalanced dataset with one missing observation, under certain assumptions on the process that generates the missing observations.

8.1.1 Assumptions on the process producing missing data

Rubin (1976) formalised the definition of data missing at random and gave the conditions in which it is appropriate to ignore the process producing missing data when making inferences about the distribution of observed data. Let us use θ to denote the parameter regulating the distribution of the observable variable and ξ to denote that of the missing data process; thus, the vector (θ,ξ) identifies the whole probability distribution of observed data within a family $\mathcal{P}$ of non-degenerate distributions. The ignorability of the missing data process depends on the method of inference and on three conditions which the data generating process must satisfy.

Definition 8.1. *The missing data are Missing At Random (MAR) if for each possible value of the parameter ξ, the conditional probability of the observed pattern of missing data given the missing data and the value of the observed data, is the same for all possible values of the missing data.*

Definition 8.2. *The observed data are Observed At Random (OAR) if for each possible value of the missing data and the parameter ξ, the conditional probability of the observed pattern of missing data given the missing data and the observed data, is the same for all possible values of the observed data. The parameter ξ is distinct from θ if there are no a priori ties, via parametric space restrictions or prior distributions, between ξ and θ.*

Definition 8.3. *If the missing data are MAR and the observed data are OAR, the missing data are Missing Completely At Random (MCAR).*

In this case, missingness does not depend on observed or unobserved values, and observed values may be considered as a random subsample of the complete dataset. In these situations, therefore, it is appropriate to ignore the process that causes missing data when making inferences on θ. We will assume in the following that this is true for the considered dataset.

8.1.2 Permutation tests for the equality of covariance operators with missing data

We show here how this applies to our case. Consider again a functional dataset of the form

$$\mathbf{X} = \{X_{ij},\ i = 1,\ldots,q,\ j = 1,\ldots,n_i\},$$

that consists of $q \geq 2$ samples or groups of size $n_i \geq 2$, with $n = \sum_i n_i$. The groups are related to q levels of a treatment and the data X_{ij} are supposed to independent and identically distributed with distributions $P_i \in \mathcal{P}$, $i = 1, \ldots, q$. In order to take into account that, for whatever reason, some of the data are missing, Pesarin and Salmaso (2010) suggested to consider the inclusion indicator associated to the considered dataset, that is

$$\mathbf{O} = \{O_{ij},\ i = 1, \ldots, q,\ j = 1, \ldots, n_i\},$$

where $O_{ij} = 1$ if X_{ij} has been observed and collected, $O_{ij} = 0$ otherwise. This indicator represents the observed configuration in the dataset. Hence, the whole set of observed data is summarized by the pair of associated matrices $(\mathbf{X}, \mathbf{O})$. We define by $\kappa_i = \sum_j O_{ij}$ the actual sample size of the observed data of each sample. Assuming that data are jointly exchangeable under the null hypothesis with respect to the groups, we would like to perform the following test

$$H_0 : \{(\mathbf{X}_1, \mathbf{O}_1) \overset{d}{=} \ldots \overset{d}{=} (\mathbf{X}_q, \mathbf{O}_q)\} \quad \text{against} \quad H_1 : \{H_0 \text{ is not true}\}.$$

Thus, the hypotheses and assumptions are such that the permutation testing principle applies. If P represents the joint multivariate distribution of $(\mathbf{X}_i, \mathbf{O}_i)$, $i = 1, \ldots, q$ under the null hypothesis, we may write

$$P = P_{\mathbf{O}} * P_{\mathbf{X}|\mathbf{O}}.$$

With this in mind, we can break down the null hypothesis in the following way:

$$H_0 : \{[O_1 \overset{d}{=} \ldots \overset{d}{=} O_q] \cap [X_1 \overset{d}{=} \ldots \overset{d}{=} X_q|\mathbf{O}]\} = \{H_0^{\mathbf{O}} \cap H_0^{\mathbf{X}|\mathbf{O}}\}.$$

Furthermore, if the missing data are MCAR, we may, according to Rubin (1976), proceed conditionally with respect to the observed inclusion indicator and ignore $H_0^{\mathbf{O}}$, assuming that $\mathbf{O}$ does not provide any discriminative information about treatment effects. Thus, as sub-hypotheses on $\mathbf{O}$ are true by assumption, $H_0^{\mathbf{O}} : \{\mathbf{O}_1 = \cdots = \mathbf{O}_q\}$ may be ignored. Hence, we may write the null hypothesis in the simpler form

$$H_0 = H_0^{\mathbf{X}|\mathbf{O}} = \{\mathbf{X}_1 \overset{d}{=} \ldots \overset{d}{=} \mathbf{X}_q|\mathbf{O}\}.$$

Now, consider a vector of partial test statistics $\mathbf{T}$ based on functions of sample valid data and denote its permutation distribution as $F(\mathbf{t}|(\mathbf{X}, \mathbf{O}))$, $\mathbf{t} \in \mathbb{R}^k$. The set of permutations $\mathbf{O}^*$ of $\mathbf{O}$, that is the set of possible permuted inclusion indicators according to the random attribution of data to the q groups, induces a partition into sub-orbits on the whole permutation sample space $(\mathbf{X}, \mathbf{O})_{/(\mathcal{X}, \mathcal{O})}$. These sub-orbits are characterized by points which exhibit the same matrix of permutation actual sample sizes of valid data $\boldsymbol{\kappa}^* = \kappa_1, \ldots, \kappa_k$. Then, if the permutation sub-distributions of the partial test statistics are invariant with respect to the sub-orbits induced by

$\mathbf{O}^*$, then we may evaluate $F[\mathbf{t}|(\mathbf{X},\mathbf{O})]$ by a simple MC procedure, that is, ignoring the missing values. Indeed, the distributional invariance with respect to permuted inclusion indicators $\mathbf{O}^*$ of sample totals implies that the equality

$$F[\mathbf{t}|(\mathbf{X},\mathbf{O})] = F[\mathbf{t}|(\mathbf{X},\mathbf{O}^*)]$$

holds for every $\mathbf{t} \in \mathbb{R}^k$, for every permutation $\mathbf{O}^*$ of $\mathbf{O}$ and for all datasets $\mathbf{X}$. This is true because in the case of the tests for covariance operators, the test statistic T of the global test is a combination of the partial test statistics of the pairwise comparisons between the groups. These, in turn, depend only on the distances between covariance operators and their permutations. We can suppose that, under the null hypothesis, the permutation distribution of the $d(S_i^*, S_j^*)$ depend essentially on the number κ_i, $i = 1,\ldots,q$ of summands. Thus, just like in the case of the multivariate analysis of variance studied in Pesarin and Salmaso (2010), the previous distributional equality becomes equivalent to

$$F[\mathbf{t}|(\mathbf{X},\boldsymbol{\kappa})] = F[\mathbf{t}|(\mathbf{X},\boldsymbol{\kappa}^*)], \tag{8.1}$$

where $\boldsymbol{\kappa}^*$ represents the vector of permutation of actual sample sizes of valid data associated to $\mathbf{O}^*$. Hence, we would like our partial test statistics to be invariant with respect to $\boldsymbol{\kappa}^*$ and for all $\mathbf{X}$. Now, suppose we are in the balanced case, i.e. $n_1 = \cdots = n_q = \bar{n}$ and one observation is missing in one of the groups, say group a, where $1 \leq a \leq q$. In the wheel-running dataset, for instance, $q = 8$, $\bar{n} = 20$ and one observation is missing in group 1. All the pairwise comparisons between groups i, with $1 \leq i \leq q$, $i \neq a$ and group j, with $1 \leq j \leq q$, $j \neq i, a$ are not affected by problem of missing data since $\kappa_i^* = \kappa_j^* = \bar{n}$. As regarding the others, at each iteration of the algorithm, we could have $\kappa_a^* = \bar{n}$ and $\kappa_j^* = \bar{n} - 1$ or viceversa, depending on the permutation. However, since distances are symmetric, this two cases are permutationally equivalent under the null hypothesis and Equation (8.1) is always satisfied. For this reason, we can simply redefine the pseudo-data matrix given in Equation (5.1) as follows:

$$\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_1 & \ldots & \mathbf{x}_{q-1} \\ \mathbf{x}_2 & \mathbf{x}_3 & \ldots & \mathbf{x}_q \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^1 & \ldots & x_1^{q-1} \\ x_2^1 & x_2^1 & \ldots & x_2^{q-1} \\ \vdots & \vdots & & \vdots \\ \text{NA} & \text{NA} & \ldots & x_{\bar{n}}^{q-1} \\ & & & \\ x_1^2 & x_1^3 & \ldots & x_1^q \\ x_2^2 & x_2^3 & \ldots & x_2^q \\ \vdots & \vdots & & \vdots \\ x_{\bar{n}}^2 & x_{\bar{n}}^3 & \ldots & x_{\bar{n}}^q \end{bmatrix}$$

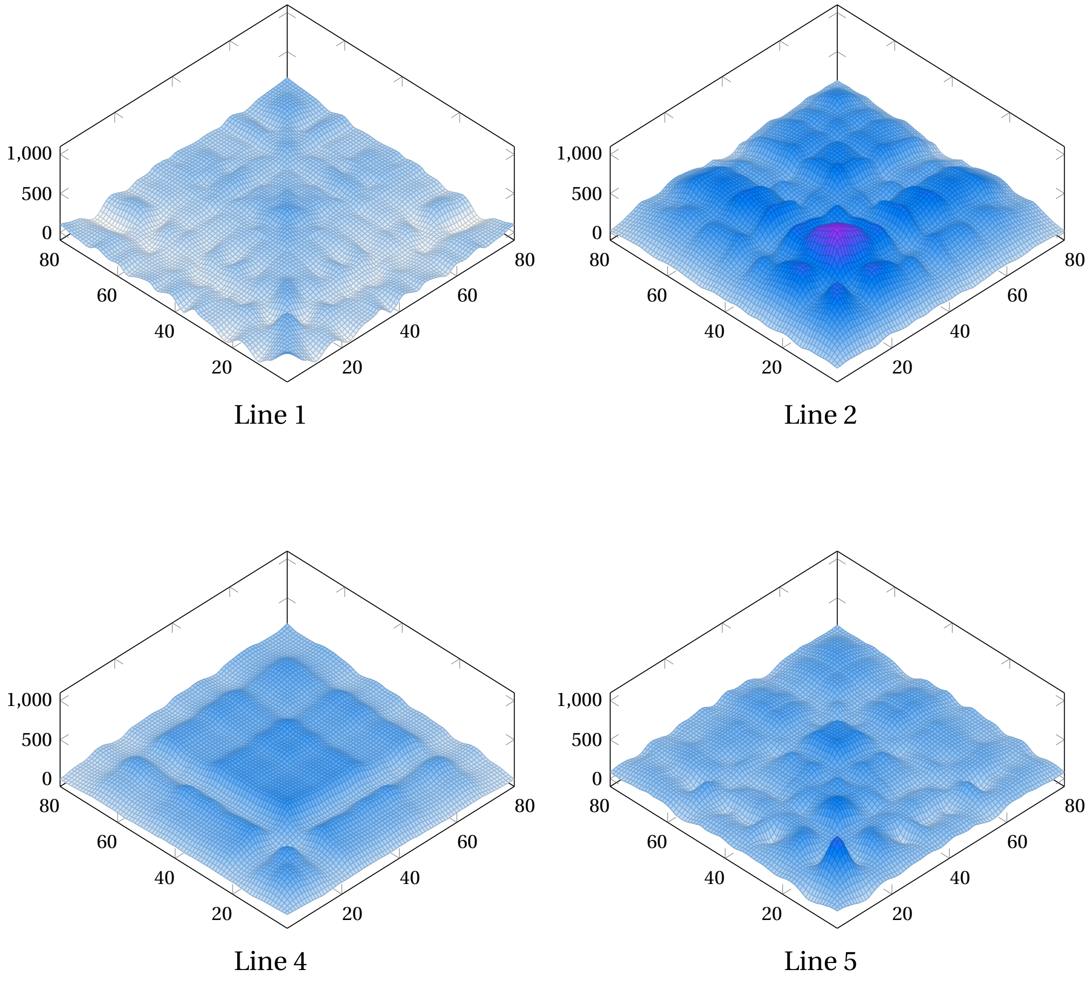
and apply the synchronised permutations as usual. At each iteration of Algorithm 5.2 the sample covariance of each pseudo-group is computed only with the available data. The situation becomes more complicated when the number of missing data becomes greater than one, since the vector κ of actual sample sizes can assume other values.

8.2 Testing the equality of the covariance operators

The aim of this analysis is to check if the covariance operators of the eight groups of mice are the same and, if this is not the case, to find out what are the mice lines that have different covariances. Figures 8.1 and 8.2 illustrate the estimated covariance operators of the aligned and smoothed data reported in Section 2.3.2. Covariances, in this experiment, represent the wheel-running behaviour of mice. The graphical representation already shows some differences between the control and selected lines. In fact, while the selected groups present an high peak around the diagonal in correspondence of week 20, this is less evident for the covariances of the control groups 2 and 5 and completely absent in groups 1 and 4. From the exploratory data analysis performed in Sections 2.3.1 and 2.3.2, it is evident that the mean of revolutions per week is greater in the selected groups. However, the covariance operators give a precious complementary information about the rise and falls of voluntary wheel-running during the life of laboratory mice. We want to test the hypothesis

$$H_0 : \{\Sigma_1 = \cdots = \Sigma_8\} \quad \text{against} \quad H_1 : \{\text{at least one of equalities is not true}\}.$$

To this end, we use the MC Algorithm 5.2 to obtain an estimate of the permutation test proposed in Section 5.2. We have shown in the previous section that synchronised permutations can be used, even if one of the observations is missing. We use the square root distance between covariance operators as partial test statistic and we choose the max T combining function. We set the number of iterations B to 1000. The global test indicates that there is strong evidence to reject the null hypothesis. This is due mainly to the first group of mice for which almost all the partial null hypotheses are rejected and, when using the max T combining function, we reject H_0 even if only one of the partial tests is rejected. In Figure 8.3 are reported the p-values of each partial test between two families of mice adjusted with the step-down method.



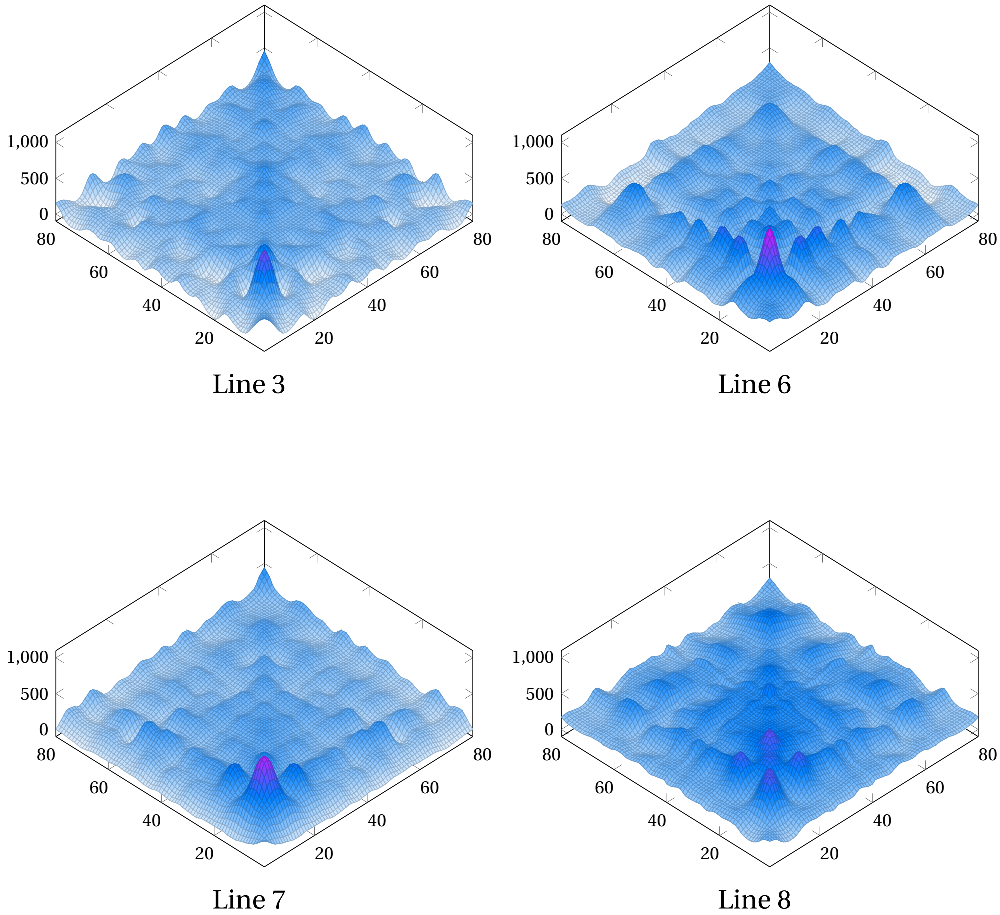
Line 1

Line 2

Line 4

Line 5

Figure 8.1: Sample covariance operators, control lines. All values are divided by 10^6.



Line 3

Line 6

Line 7

Line 8

Figure 8.2: Sample covariance operators, selected lines. All values are divided by 10^6.

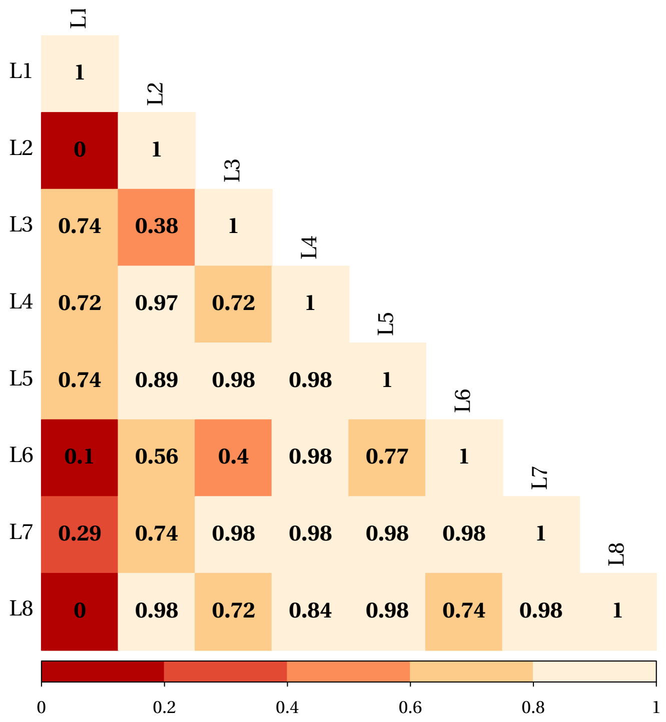


Figure 8.3: Partial p-values of the synchronised permutation test on the covariance operators of the aligned data. For each $i = 1,\dots,k$ and $j = 1,\dots,k, i \neq j$, the value reported in row i, column j corresponds to the adjusted p-value of the test $H_0 : \{\Sigma_i = \Sigma_j\}$ against $H_1 : \{\Sigma_i \neq \Sigma_j\}$. The global p-value of the test is the minimum of the partial p-values and therefore is less than $1/1000$.

Chapter 9

Conclusions

The aim of this work was to study how the two-sample permutation test for the equality of covariance operators suggested by Pigoli et al. (2014) could be extended to the case of multiple samples. To this end, we have introduced in Chapter 3 the non-parametric combination methodology of Pesarin and Salmaso (2010) that enabled us to generate Algorithm 5.2 to perform the test of interest.

The permutation approach has many advantages. Not only it allows to use any metric, but also permits to choose among a wide range of combining functions. Besides, it makes no assumption on the process generating the functional data. Indeed, it is so general that it could be applied, for instance, also to the comparison of the mean functions, provided that apposite distance are used.

The empirical power of the test has been assessed through the use of simulation studies, reported in Chapter 7. Synthetic data sets have been generated from Student-t and Gaussian processes, with the number of samples ranging from three to ten. The simulations have been performed with different test statistics, varying both the distances between covariance operators and the combining functions, and with three distinct types of permutations. These have shown that, among the distances proposed by Pigoli et al. (2014), those that better capture the differences between covariance operators are the square root and the Procrustes distances. Simulation studies also helped verify that the best results are obtained when the permutations applied in Algorithm 5.2 are synchronised.

Moreover, we have used simulation studies to compare the new method with the existing ones. We evinced that the permutation approach has the same empirical power as the bootstrap proposed by Paparoditis and Sapatinas (2014). However, the former allows for more flexibility in the choice of the test statistics. On the contrary, these two methods perform very differently with respect to the generalisation of the Levene's test (Anderson, 2006). In fact, it captures very well the differences in scale, but lacks of precision in the other cases. Finally, the test by Kashlak et al. (2016) is a valid alternative if data are Gaussian, but cannot be used otherwise.

In addition to that, we wanted to find a way to carry out pairwise comparisons between the data samples, if the global null hypothesis is rejected. The great advantage of using the NPC methodology is that no computational effort is needed to compute the partial p-values of the tests. In order to control the family-wise error rate of the whole procedure, in Chapter 6 we have reviewed different ways of correcting the p-values and we have indicated which are the most suitable, according to the combining function used in Algorithm 5.2.

Furthermore, in Chapter 8 we have shown the practical utility of the test by applying it to the evolutionary biology data set of Swallow et al. (1998) presented in the Introduction. The data set is composed of eight groups, each corresponding to a line of laboratory mice. The permutation test for the equality of the covariance operators of the eight lines of mice evidenced that only one of the groups presents a covariance that is significantly different from the others. Instead, there are no substantial dissimilarities in the covariances of the other lines, even if some of the corresponding families of mice were bred differently.

All the functions needed to run the permutation test and are implemented in R and are now available on CRAN (Cabassi and Kashlak, 2016). In particular, the function `ksample_perm` allows to perform Algorithm 5.2 with different distances and combining functions. The package also enables users to make inference on covariance operators through the non-asymptotic methods of Kashlak et al. (2016), if the global null hypothesis is rejected. For further details, refer to Appendix A.

9.1 Future work

We have mentioned earlier that, when the post-hoc comparisons are of interest, it is better to use synchronised permutations. Unfortunately, these can only be performed when the considered data samples have the same cardinality. Therefore, the possibility to apply synchronised permutations with non-balanced groups would make this test more efficient in many situations. For the moment, we have proven that this method can be applied also if one observation is missing. We believe that extending this proof to other, more general cases when the number of missing observations is greater than one could make the test even more useful.

Another possible improvement the implementation in C++ of the functions provided in the R package. In fact, permutation tests are computationally heavy especially in the case of functional data analysis, where the number of variables p is usually high. For this reason, the tests can require a significant amount of time when applied to large data sets.

Finally, it could also be interesting to develop an extension of this method to the two-way ANOVA framework, analysing the effect of two different variables and their interaction on the covariance operators. In that framework, however, the ideas used for similar problems by Pesarin and Salmaso (2010) could not be applied. In fact they are based on the linearity of the test statistics employed in the univariate and multivariate cases, that does not hold in this case. Consequently, new approaches should be explored.

Appendix A

Documentation of the R package

The R package “fdcov” has been developed in collaboration with Adam B. Kashlak. It is intended to be a tool that makes available to R users all the functions needed in order to perform the permutation tests proposed in this thesis as well as the non-asymptotic test introduced by Kashlak et al. (2016). Even if the tests are based on very different principles, we wanted to reunite in a single place all the code related to the statistical inference on covariance operators of functional data. We hope that this will make it easier for end-users to choose which approach to adopt, according to their needs.

The functions related to the permutation tests are `ksample.perm` and `perm.plot`. The first one takes as input the data set and the group labels and performs the permutation test for the equality of the covariance operators of the groups. Many parameters have default values but can be also set by the user. These are: the number of iterations of the Monte Carlo Algorithm 5.2, the type of permutations, the distance between covariance operators and the combining function. Moreover, it is also possible to perform the post-hoc comparisons. In that case, the user can decide to let the function automatically adjust the p-values to control the family-wise error rate of the whole procedure or to be given the raw p-values. The other function plots the partial p-values in a matrix like the one in Figure 8.3.

The data set taken as example was developed by a joint collaboration between Andreas Buja, Werner Stuetzle and Martin Maechler, and used as illustration by Kashlak et al. (2016). The description of the data set is available at http:www-stat.stanford.edu/ElemStatLearn. Data can be downloaded from the R package `fds` (Shang and Hyndman, 2013).

For all details regarding the package, please refer to the documentation reported hereafter, also available at https://cran.r-project.org/web/packages/fdcov/.

Package ‘fdcov’

June 28, 2016

Title Analysis of Covariance Operators

Version 1.0.0

Author Alessandra Cabassi [aut],
Adam B Kashlak [aut, cre],
Davide Pigoli [ctb]

Maintainer Adam B Kashlak <ak852@cam.ac.uk>

Description Provides a variety of tools for the analysis of covariance operators.

Depends R (>= 3.2.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1

Imports matlab,
corrplot

Suggests fds

R topics documented:

fdcov-package *Analysis of Covariance Operators.*

Description

fdcov provides a variety of tools for the analysis of covariance operators.

Details

This package contains a collection of tools for performing statistical inference on functional data specifically through an analysis of the covariance structure of the data. It includes two methods for performing a k-sample test for equality of covariance in ksample.perm and ksample.com. For supervised and unsupervised learning, it contains a method to classify functional data with respect to each category's covariance operator in classif.com, and it contains a method to cluster functional data, cluster.com, again based on the covariance structure of the data.

The current version of this package assumes that all functional data is sampled on the same grid at the same intervals. Future updates are planned to allow for the below methods to interface with the fda package and its functional basis representations of the data.

Author(s)

Alessandra Cabassi <alessandra.cabassi@mail.polimi.it>, Adam B Kashlak <ak852@cam.ac.uk>

References

Kashlak, Adam B, John AD Aston, and Richard Nickl (2016). "Inference on covariance operators via concentration inequalities: k-sample tests, classification, and clustering via Rademacher complexities", April, 2016 (in review)

Pigoli, Davide, John AD Aston, Ian L Dryden, and Piercesare Secchi. "Distances and inference for covariance operators." Biometrika (2014): asu008.

classifier-com *Functional data classifier via concentration inequalities*

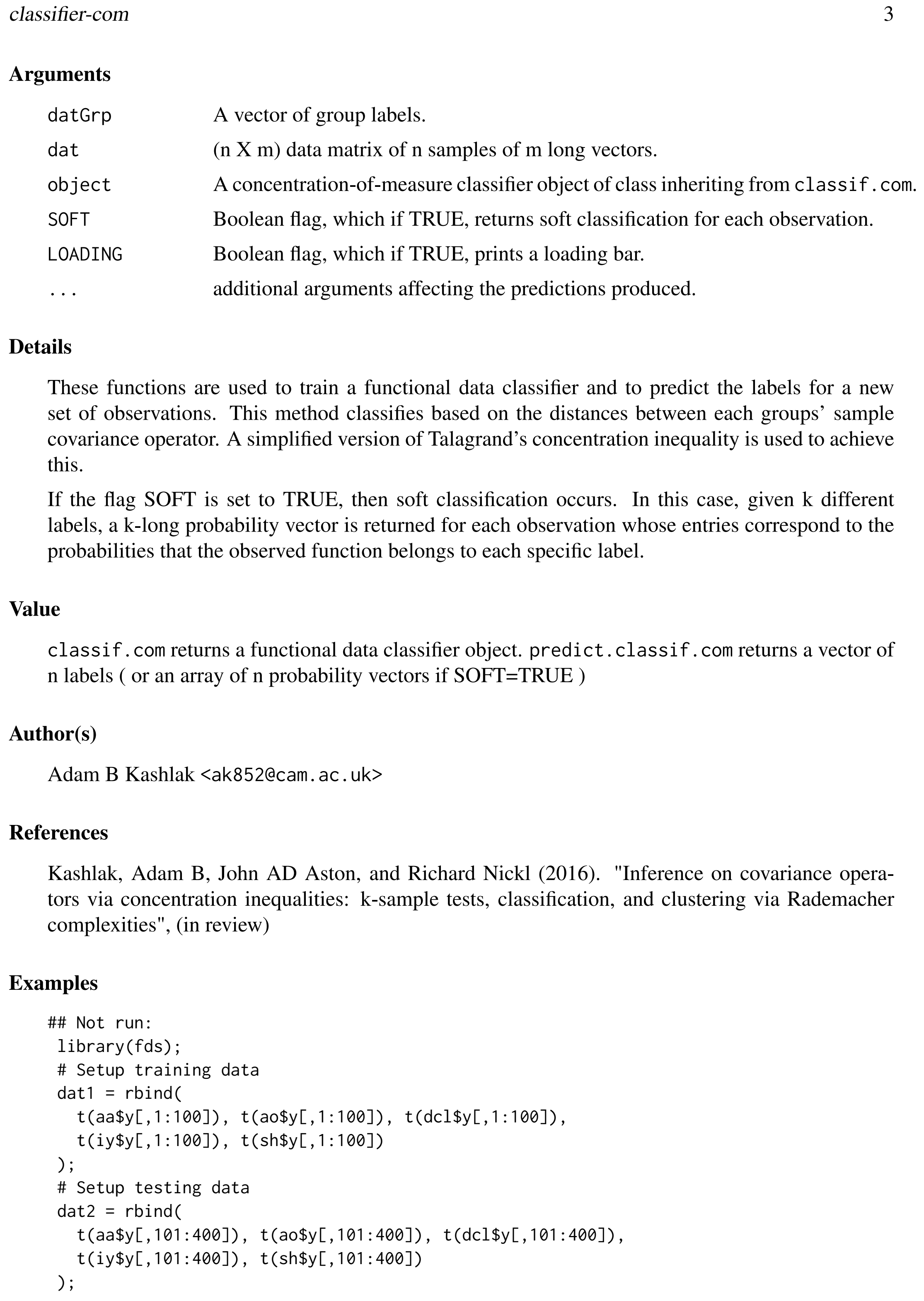
Description

classif.com trains a covariance operator based functional data classifier that makes use of concentration inequalities. predict.classif.com uses the previously trained classifier to classify new observations.

Usage

```
classif.com(datGrp, dat)

## S3 method for class 'classif.com'
predict(object, dat, SOFT = FALSE, LOADING = FALSE,
  ...)
```



Arguments

`datGrp`	A vector of group labels.
`dat`	(n X m) data matrix of n samples of m long vectors.
`object`	A concentration-of-measure classifier object of class inheriting from `classif.com`.
`SOFT`	Boolean flag, which if TRUE, returns soft classification for each observation.
`LOADING`	Boolean flag, which if TRUE, prints a loading bar.
`...`	additional arguments affecting the predictions produced.

Details

These functions are used to train a functional data classifier and to predict the labels for a new set of observations. This method classifies based on the distances between each groups' sample covariance operator. A simplified version of Talagrand's concentration inequality is used to achieve this.

If the flag SOFT is set to TRUE, then soft classification occurs. In this case, given k different labels, a k-long probability vector is returned for each observation whose entries correspond to the probabilities that the observed function belongs to each specific label.

Value

`classif.com` returns a functional data classifier object. `predict.classif.com` returns a vector of n labels (or an array of n probability vectors if SOFT=TRUE)

Author(s)

Adam B Kashlak <`ak852@cam.ac.uk`>

References

Kashlak, Adam B, John AD Aston, and Richard Nickl (2016). "Inference on covariance operators via concentration inequalities: k-sample tests, classification, and clustering via Rademacher complexities", (in review)

Examples

```
## Not run:
 library(fds);
 # Setup training data
 dat1 = rbind(
   t(aa$y[,1:100]), t(ao$y[,1:100]), t(dcl$y[,1:100]),
   t(iy$y[,1:100]), t(sh$y[,1:100])
 );
 # Setup testing data
 dat2 = rbind(
   t(aa$y[,101:400]), t(ao$y[,101:400]), t(dcl$y[,101:400]),
   t(iy$y[,101:400]), t(sh$y[,101:400])
 );
```

```
  datgrp = gl(5,100);
  clCom = classif.com( datgrp, dat1 );
  grp = predict( clCom, dat2, LOADING=TRUE );
  acc = c(
    sum( grp[1:300]==1 ), sum( grp[301:600]==2 ), sum( grp[601:900]==3 ),
    sum( grp[901:1200]==4 ), sum( grp[1201:1500]==5 )
  )/300;
  print(rbind(gl(5,1),signif(acc,3)));

## End(Not run)
```

cluster.com — *Functional data clustering via concentration inequalities*

Description

`cluster.com` clusters sets of functional data via their covariance operators making use of an EM style algorithm with concentration inequalities.

Usage

```
cluster.com(dat, labl = NULL, grpCnt = 2, iter = 30, SOFT = FALSE,
  PRINTLK = TRUE, LOADING = FALSE, IGNORESTOP = FALSE)
```

Arguments

`dat`	(n X m) data matrix of n samples of m long vectors.
`labl`	An optional vector of n labels to group curves. (see Details)
`grpCnt`	Number of clusters into which to split the data.
`iter`	Number of iterations for EM algorithm.
`SOFT`	Boolean flag for whether or not category probabilities should be returned.
`PRINTLK`	Boolean flag, which if TRUE, prints likelihood values for each iteration.
`LOADING`	Boolean flag, which if TRUE, prints a loading bar.
`IGNORESTOP`	Boolean flag, which if TRUE, will ignore early stopping conditions and cause the EM algorithm to run for the total amount of desired iterations.

Details

This function clusters individual curves or sets of curves by considering the distance between their covariance operator and each estimated category covariance operator. The implemented algorithm reworks the concentration inequality based classification method `classif.com` into an EM style algorithm. This method iteratively updates the probability of a given observation belonging to each of the k categories. These probabilities are in turn used to update the category means. This process continues until either the total number of iterations is reached or a computed likelihood begins to decrease signaling the arrival of a local optimum.

If the argument labl is NULL, then every curve is clustered separately. If labl contains factors used to group the curves, then each set of curves is classified as one group. For example, if you have multiple speakers and multiple speech samples from each speaker, you can group the data from each speaker together in order to cluster based on each speakers' covariance operator rather than based on each speech sample individually.

If the flag SOFT is set to TRUE, then soft clustering occurs. In this case, given k different labels, a k-long probability vector is returned for each observation whose entries correspond to the probability that the observed function belongs to a specific label.

Value

`cluster.com` returns a vector a labels with one entry for each row of data corresponding to one of the k categories (or an array of probability vectors if SOFT=TRUE).

Author(s)

Adam B Kashlak <`ak852@cam.ac.uk`>

References

Kashlak, Adam B, John A D Aston, and Richard Nickl (2016). "Inference on covariance operators via concentration inequalities: k-sample tests, classification, and clustering via Rademacher complexities", in review

Examples

```
## Not run:
 # Load phoneme data
 library(fds);
 # Setup data to be clustered
 dat  = rbind( t(aa$y[,1:20]),t(iy$y[,1:20]),t(sh$y[,1:20]) );
 # Cluster data into three groups
 clst = cluster.com(dat,grpCnt=3);
 matrix(clst,3,20,byrow=TRUE);

 # cluster groups of curves
 dat  = rbind( t(aa$y[,1:40]),t(iy$y[,1:40]),t(sh$y[,1:40]) );
 lab  = gl(30,4);
 # Cluster data into three groups
 clst = cluster.com(dat,labl=lab,grpCnt=3);
 matrix(clst,3,10,byrow=TRUE);

## End(Not run)
```

`ksample.com` *k-sample test for equality of covariance operators*

Description

`ksample.com` performs a k-sample test for equality of covariance operators using concentration inequalities.

Usage

```
ksample.com(dat, grp, p = 1, alpha = 0.05, scl1 = 1, scl2 = 1)
```

Arguments

`dat`	(n X m) data matrix of n samples of m long vectors.
`grp`	n long vector of group labels.
`p`	p-Schatten norm in [1,Inf], Default is 1. (see Details)
`alpha`	the desired size of the test, Default is 0.05.
`scl1`	scales the deviation part of the concentration inequality. (see Details)
`scl2`	scales the Rademacher part of the concentration inequality. (see Details)

Details

This function tests for the equality of k covariance operators given k sets of functional data. It makes use of Talagrand's concentration inequality in the Banach space setting. The argument p specifies the p-Schatten norm used in the test. As detailed in Kashlak et al (2016), the most power is achieved using the trace class norm (p=1), which is the default value.

This test is inherently conservative as it constructed by concatenating many concentration inequalities together. Consequently, the method may be tuned by adjusting the arguments scl1 and scl2 to achieve the desired empirical size for the users specific data set. Otherwise, it can be used as a quick first pass before a more powerful but more computational test, such as specifically `ksample.perm`, is run. More information on tuning this method can be found in the reference.

Value

Boolean value for whether or not the test believes the alternative hypothesis is true. (i.e. Does there exist at least two categories of the k whose covariance operators are not equal?)

Author(s)

Adam B Kashlak `<ak852@cam.ac.uk>`

References

Kashlak, Adam B, John AD Aston, and Richard Nickl (2016). "Inference on covariance operators via concentration inequalities: k-sample tests, classification, and clustering via Rademacher complexities", (in review)

Examples

```
# Load in phoneme data
library(fds)
# Setup data arrays
dat1 = rbind( t(aa$y)[1:20,], t(sh$y)[1:20,] );
dat2 = rbind( t(aa$y)[1:20,], t(ao$y)[1:20,] );
dat3 = rbind( dat1, t(ao$y)[1:20,] );
# Setup group labels
grp1 = gl(2,20);
grp2 = gl(2,20);
grp3 = gl(3,20);
# Compare two disimilar phonemes (should return TRUE)
ksample.com(dat1,grp1);
# Compare two similar phonemes (should return FALSE)
ksample.com(dat2,grp2);
# Compare three phonemes (should return TRUE)
ksample.com(dat3,grp3);
```

ksample.perm *Multiple-sample permutation test for the equality of covariance operators of functional data*

Description

The method performs a test for the equality of the covariance operators of multiple data samples. It can also perform all of the pairwise comparisons between the groups and compute a p-value for each of them. This feature is useful when the global null hypothesis is rejected, so one may want to find out which samples have different covariances.

Usage

```
ksample.perm(dat, grp, iter = 1000, perm = "sync", dist = "sq",
  adj = TRUE, comb = "tipp", part = FALSE, cent = FALSE, load = FALSE)
```

Arguments

`dat`	n X p data matrix of n samples of p long vectors.
`grp`	n long vector of group labels.
`iter`	Number of permutations. Defaults to 1000.
`perm`	Type of permutation, can be 'sync' (if all the data samples are of the same size) or 'pool'. Defaults to 'sync'
`dist`	Distance between covariance operators. Can be 'sq' (square-root distance), 'tr' (trace distance),'pr' (Procrustes distance), 'hs'(Hilbert-Schmidt distance) or 'op' (operator distance). Defaults to 'sq'.
`adj`	p-value adjustment. Defaults to TRUE.

`comb` Can be 'tipp' (for Tippett), 'maxT', 'dire' (direct), 'fish' (Fisher) or 'lipt' (Liptak). Defaults to 'tipp'.

`part` If FALSE, the function computes only the global p-value; otherwise it computes also all the p-values corresponding to the pairwise comparisons. Defaults to FALSE.

`cent` If FALSE, the mean functions of the groups are supposed to be different, therefore data are centred before performing the test. Defaults to FALSE.

`load` Boolean flag, which if TRUE, prints a loading bar.

Value

If `part` is set to FALSE, the output is the p-value associated to the global test. If `part` is TRUE, the function returns also all the p-values of the pairwise comparisons.

Author(s)

Alessandra Cabassi `<alessandra.cabassi@mail.polimi.it>`

References

Pigoli, Davide, John A. D. Aston, Ian L. Dryden, and Piercesare Secchi (2014). "Distances and inference for covariance operators." Biometrika: asu008.

Examples

```
## Not run:
## Phoneme data

library(fdcov)
library(fds)

# Create data set
data(aa); data(ao); data(dcl);data(iy);data(sh)
dat = cbind(aa$y[,1:20],ao$y[,1:20],dcl$y[,1:20],iy$y[,1:20],sh$y[,1:20])
dat = t(dat)
grp = c(rep(1,20),rep(2,20),rep(3,20),rep(4,20),rep(5,20))

# Test the equality of the covariance operators
p = ksample.perm(dat, grp, iter=100, part = TRUE)
p$global # global p-value
p$partial # partial p-values

## End(Not run)
```

perm.plot *Plot partial p-values*

Description

perm.plot plots all of the partial comparison p-values in a matrix.

Usage

```
perm.plot(p, k, lab = NULL, save = FALSE, name = "pvalues.eps")
```

Arguments

p	Output of function perm.test, if part = TRUE.
k	Number of groups, must be greater than 2.
lab	Group labels. Defaults to 1, 2, ..., k.
save	Boolean variable that indicates if the plot must be saved as an .eps. Defaults to FALSE.
name	If save is TRUE, this is the filename of the plot. Defaults to pvalues.eps.

Value

perm.plot plots the partial p-values in a matrix.

Author(s)

Alessandra Cabassi <alessandra.cabassi@mail.polimi.it>

References

Pigoli, Davide, John A. D. Aston, Ian L. Dryden, and Piercesare Secchi (2014). "Distances and inference for covariance operators." Biometrika: asu008.

Examples

```
## Not run:
## Phoneme data

library(fdcov)
library(fds)

# Create data set
data(aa); data(ao); data(dcl);data(iy);data(sh)
dat=cbind(aa$y[,1:20],ao$y[,1:20],dcl$y[,1:20],iy$y[,1:20],sh$y[,1:20])
dat=t(dat)
grp=c(rep(1,20),rep(2,20),rep(3,20),rep(4,20),rep(5,20))

# Test the equality of the covariance operators
```

```
p=ksample.perm(dat,grp,iter=100,only.glob=FALSE)

# Plot partial p-values
perm.plot(p,5, lab=c('aa','ao','dcl','iy','sh'))

## End(Not run)
```

Index

Appendix B

Code

We report here the code produced for this work. In Section B.1 are given all the functions implemented for the permutation tests. Section B.2 contains the functions used in order to control the family-wise error rate, according to the different methods explained in Chapter 6. The code presented in the first two sections can be downloaded from the package webpage, https://cran.r-project.org/web/packages/fdcov/. Lastly, Sections B.3 and B.4 are dedicated to the empirical bootstrap and the generalisation of the Levene's test respectively. The non-asymptotic test based on the concentration measure of Kashlak et al. (2016), instead, has been run using the function contained in the R package "fdcov".

B.1 Multiple-sample permutation test

Main function

```r
#' Multiple-sample permutation test for the equality of covariance operators of functional data
#'
#' The method performs a test for the equality of the covariance operators of multiple data samples.
#' It can also perform all of the pairwise comparisons between the groups and compute a p-value for each of them.
#' This feature is useful when the global null hypothesis is rejected, so one may want to find out which samples have different covariances.
#'
#' @param dat n X p data matrix of n samples of p long vectors.
#' @param grp n long vector of group labels.
#' @param iter Number of permutations. Defaults to 1000.
```

```
#’ @param perm Type of permutation, can be ’sync’ (if all the data samples are of the same size) or ’pool’. Defaults to ’sync’
#’ @param dist Distance between covariance operators. Can be ’sq’ (square-root distance), ’tr’ (trace distance),’pr’ (Procrustes distance), ’hs’(Hilbert-Schmidt distance) or ’op’ (operator distance). Defaults to ’sq’.
#’ @param adj p-value adjustment. Defaults to TRUE.
#’ @param comb Can be ’tipp’ (for Tippett), ’maxT’, ’dire’ (direct), ’fish’ (Fisher) or ’lipt’ (Liptak). Defaults to ’tipp’.
#’ @param part If FALSE, the function computes only the global p-value; otherwise it computes also all the p-values corresponding to the pairwise comparisons. Defaults to FALSE.
#’ @param cent If FALSE, the mean functions of the groups are supposed to be different, therefore data are centred before performing the test. Defaults to FALSE.
#’ @param load Boolean flag, which if TRUE, prints a loading bar.
#’
#’ @return If \code{part} is set to FALSE, the output is the p-value associated to the global test. If \code{part} is TRUE, the function returns also all the p-values of the pairwise comparisons.
#’
#’ @author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}
#’
#’ @references Pigoli, Davide, John A. D. Aston, Ian L. Dryden, and Piercesare Secchi (2014). "Distances and inference for covariance operators." Biometrika: asu008.
#’
#’ @examples
#’ \dontrun{
#’ ## Phoneme data
#’
#’ library(fdcov)
#’ library(fds)
#’
#’ # Create data set
#’ data(aa); data(ao); data(dcl);data(iy);data(sh)
#’ dat = cbind(aa$y[,1:20],ao$y[,1:20],dcl$y[,1:20],iy$y[,1:20],sh$y[,1:20])
#’ dat = t(dat)
#’ grp = c(rep(1,20),rep(2,20),rep(3,20),rep(4,20),rep(5,20))
#’
#’ # Test the equality of the covariance operators
#’ p = ksample.perm(dat, grp, iter=100, part = TRUE)
#’ p$global # global p-value
#’ p$partial # partial p-values
#’ }
```

```
#' \dontshow{
#' library(fdcov)
#' library(fds)
#'
#' # Create data set
#' data(aa); data(ao); data(dcl);data(iy);data(sh)
#' dat = cbind(aa$y[,1:20],ao$y[,1:20],dcl$y[,1:20],iy$y[,1:20],sh$y[,1:20])
#' dat = t(dat)
#' grp = c(rep(1,20),rep(2,20),rep(3,20),rep(4,20),rep(5,20))
#'
#' # Test the equality of the covariance operators
#' p = ksample.perm(dat,grp,iter=2)
#' }
#' @export

ksample.perm = function(dat, grp, iter = 1000, perm = 'sync', dist = 'sq', adj
    = TRUE, comb = 'tipp', part = FALSE, cent = FALSE, load = FALSE){

    table_groups = table(grp) # groups table
    C = length(table_groups) # number of groups
    if(C < 2) stop('The number of groups must be at least 2.')
    if(C == 2) return(list(global = twosample.perm(dat, grp, iter, dist, load)))


    ### Step 1: data centring

    if(cent == FALSE){
        nonalign_dat = dat
        for(i in 1:length(grp)){
            dat[i,] = nonalign_dat[i,]-colMeans(nonalign_dat[grp == grp[i],],na.
                rm=TRUE)
        }
    }

    ### Steps 2, 3 and 4: apply 'iter' permutations and compute the test
        statistic for each permuted data set

    T = switch (perm,
                sync = perm.sync(dat, grp, iter, dist, load),
                pool = perm.pool(dat, grp, iter, dist, load),
                stop('The selected permutation strategy is not available'))



    ### Global p-values

    if (comb == ’tipp’ | comb == ’fish’ | comb == ’lipt’ | adj == FALSE) P =
        perm.t2p(T)

    P.glob = switch (comb,
                        tipp = mean(min(P[1,]) ≥apply(P[-1,],1,min)),
                        maxT = mean(max(T[1,]) ≤apply(T[-1,],1,max)),
                        dire = mean(sum(T[1,]) ≤apply(T[-1,],1,sum)),
                        fish = mean(comb.fish(P[1,]) ≤apply(P[-1,],1,comb.fish)),
                        lipt = mean(comb.lipt(P[1,]) ≤apply(P[-1,],1,comb.lipt)),
                        stop(’The selected combining function is not available’))

    ### Partial p-values

    if(part == TRUE){
        if(adj == TRUE){ # Corrected p-values
            P.part = switch (comb,
                                tipp = FWE.tipp(P),
                                maxT = FWE.maxT(T),
                                dire = FWE.clos(T, comb, load),
                                fish = FWE.clos(P, comb, load),
                                lipt = FWE.clos(P, comb, load))
        }else{P.part = matrix(P[1,],nrow = C*(C-1)/2,ncol = 1) # Raw p-values
        }

        # Group names definition
        name = character()
        cont = 1
        for(i in 1:(C-1)){
            for(j in (i+1):C){
                name[cont] = paste(i,j,sep="-")
                cont = cont+1
            }
        }

        P.part = data.frame(dist = T[1,],p_value = P.part, signif = perm.sig(P.
            part))
        rownames(P.part) = name

        return(list(global = P.glob,partial = P.part))
    }
    else{return(list(global = P.glob))}
}
```

```
# Synchronised permutations of a data set
#
# @param dat n x p data set matrix
#
# @return This function returns a matrix containing the partial test statistics of each pairwise comparison for each permutation applied to the data set
#
# @author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}
#

perm.sync = function(dat, grp, iter, dist, load = FALSE){

    table_groups = table(grp) # groups table
    C = length(table_groups) # number of groups
    K = C*(C-1)/2 # number of partial tests
    p = dim(dat)[2] # number of samplings per function

    T = array(5,dim = c((iter+1),K)) # test statistics vector initialisation

    ### Step 2: compute test statistic

    cont = 1
    for(i in 1:(C-1)){
        for(j in (i+1):C){ # for each pair of groups
            # compute test statistic for initial data
            T[1,cont] = distCov(cov(dat[grp==i,],use=’pairwise’),cov(dat[grp==j,],use=’pairwise’),dist)
            cont = cont+1
        }
    }

    ### Steps 3 and 4: apply iter permutations and compute the test statistic for each permuted data set

    # Build pseudomatrix

    n = (dim(dat)[1])/C
    X = array(0,dim = c((2*n),p,K)) # matrix 2*n X p X number of groups
    cont = 1
    for(i in 1:(C-1)){ # for each pair of groups
        for(j in (i+1):C){
            X[,,cont] = rbind(dat[grp==i,],dat[grp==j,]) # fill in the matrix
            cont = cont+1
        }
    }


    # Apply 'iter' permutations to the pseudomatrix and compute test statistics

    if(load) pb = txtProgressBar(min = 0, max = iter, style = 3) # create progress bar
    for(bb in 2:(iter+1)){
        X.perm = X[sample(2*n),,] # select permutation
        cont = 1
        for(i in 1:(C-1)){ # for each pair of groups
            for(j in (i+1):C){ # compute test statistic for the permuted dataset
                T[bb,cont] = distCov(cov(X.perm[c(1:n),,cont],use = 'pairwise'), cov(X.perm[-c(1:n),,cont],use = 'pairwise'),dist)
                cont = cont+1
            }
        }
        if(load) setTxtProgressBar(pb, bb-1) # update progress bar
    } # end iter
    if(load) close(pb) # close progress bar

    return(T)
}

# Pooled permutations of a data set
#
# @param dat n x p data set matrix
#
# @return This function returns a matrix containing the partial test statistics of each pairwise comparison for each permutation applied to the data set
#
# @author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}
#

perm.pool = function(dat, grp, iter, dist, loading){

    table_groups = table(grp) # groups table
    C = length(table_groups) # number of groups
    K = C*(C-1)/2 # number of partial tests
    p = dim(dat)[2] # number of samplings per function
    N = dim(dat)[1] # total number of observations

    T = array(5,dim=c((iter+1),K)) # test statistics vector initialisation

    ### Step 2: compute test statistic

    cont=1
    for(i in 1:(C-1)){
        for(j in (i+1):C){ # for each pair of groups
            # compute test statistic for initial data
            T[1,cont] = distCov(cov(dat[grp==i,],use=’pairwise’),cov(dat[grp==j,],use=’pairwise’),dist)
            cont = cont+1
        }
    }

    ### Steps 3 and 4: apply iter permutations and compute the test statistic for each permuted data set

    if(loading) pb = txtProgressBar(min = 0, max = iter, style = 3) # create progress bar
    for(bb in 2:(iter+1)){

        dat.perm = dat[sample(N),] # apply permutation
        cont = 1

        for(i in 1:(C-1)){
            for(j in (i+1):C){ # compute test statistic for the permuted dataset

                T[bb,cont] = distCov(cov(dat.perm[grp==i,],use=’pairwise’),cov(dat.perm[grp==j,],use=’pairwise’),dist)
                cont = cont+1

            }
        }
        if(loading) setTxtProgressBar(pb, bb-1) # update progress bar
    } # end iter
    if(loading) close(pb) # close progress bar

    return(T)
}
```

p-value computation

```
# Permutation p-value
#
# Given a vector of test statistics, where the first one is considered to be the one corresponding to the observed data set, computes the p-value of the test
#
# @param t Vector of test statistics
# @param extr To set which values are more extreme than the one observed. Can be \code{greater} or \code{lesser}.
#
# @return This function returns the p-value of the test, that is the percentages of test statistics in the vector that are more extreme than the one observed
#
# @author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}
#
perm.pval = function(t, extr = 'greater'){
     if(extr == 'greater') p = mean(t ≥t[1],na.rm=TRUE)
     else p = mean(t ≤t[1],na.rm=TRUE)
     return(p)
    }
```

```
# Liptak combining function
#
# @param v Vector of p-values
#
# @return This function returns a global p-value computed according to the Fisher combining function
#
# @author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}
#
# @references Pesarin, Fortunato, and Luigi Salmaso (2010). Permutation tests for complex data: theory, applications and software. John Wiley and Sons.
#
# @export
#
comb.lipt = function(v, iter){

    q = rep(0,length(v))
    for (i in 1:length(v)){
       p = (v[i] + 1/(2*iter))/(1 + 1/iter)
       q[i] = qnorm(p)
    }

    return(-sum(q))
}
```

```
# Fisher combining function
#
# @param v Vector of p-values
#
# @return This function returns a global p-value computed according to the Fisher combining function
#
# @author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}
#
# @references Pesarin, Fortunato, and Luigi Salmaso (2010). Permutation tests for complex data: theory, applications and software. John Wiley and Sons.
#

comb.fish = function(v){

    return(-2*sum(log(v)))

}
```

```
# Convert test statistics into p-values
#
# @param T Matrix of test statistics
#
# @return Matrix containing p-values
#
# @author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}
#

perm.t2p = function(T){

  oth = 2

  B = dim(T)[1]-1
  p = dim(T)[2]

  rango = function(x){
    r=1-rank(x[-1],ties.method="min")/B+1/B
    return(c(mean(x[-1]≥x[1]),r))
  }

  P=apply(T,oth,rango)
  return(P)
}
```

Distances between covariance operators

```
# General distances function
#
# @param mat1 First covariance matrix
# @param mat2 Second covariance matrix
# @param dist Distance between covariance operators. Can be ’sq’ (square-root),
     ’tr’ (trace),’pr’ (Procrustes), ’hs’(Hilbert-Schmidt) or ’op’ (operator).
#
# @return Distance.
#
distCov = function( mat1, mat2, type )
{
   switch( type,
     sq = distSqrt(mat1,mat2),
     tr = distTrac(mat1,mat2),
     pr = distProc(mat1,mat2),
     hs = distHsno(mat1,mat2),
     op = distOper(mat1,mat2)
   );
}
```

```
# Trace Class distance
#
# @param mat1 First covariance matrix
# @param mat2 Second covariance matrix
#
# @export
#
distTrac = function( mat1, mat2 )
{
  return( pschnorm( mat1-mat2,1 ) );
}
```

```
# Hilbert-Schmidt distance
#
# @param mat1 First covariance matrix
# @param mat2 Second covariance matrix
#
```

```
# @export
#
distHsno = function( mat1, mat2 )
{
  return( pschnorm( mat1-mat2, 2 ) );
}
```

```
# Operator norm distance
#
# @param mat1 First covariance matrix
# @param mat2 Second covariance matrix
#
# @export
#
distOper = function( mat1, mat2 )
{
  return( pschnorm( mat1-mat2, -1 ) );
}
```

```
# Square Root distance
#
# @param mat1 First covariance matrix
# @param mat2 Second covariance matrix
#
# @export
#
distSqrt = function( mat1, mat2 )
{
  smat1 = sqrtMat(mat1);
  smat2 = sqrtMat(mat2);
  return( pschnorm( smat1-smat2, 2 ) );
}
```

```
# Procrustes distance
#
# @param mat1 First covariance matrix
# @param mat2 Second covariance matrix
#
# @export
#
distProc = function( mat1, mat2 )
{
  smat1 = sqrtMat(mat1);
  smat2 = sqrtMat(mat2);
  matC = t(smat2)%*%smat1;
  svdC = svd(matC);
  matR = svdC$u%*%t(svdC$v);
  return( pschnorm( smat1-smat2%*%matR, 2 ) );
}
```

```
# Hilbert-Schmidt (Frobenius) Norm
#
# @param sig covariance matrix (i.e. symmetric positive definite)
#
# @param HS Norm of sig
#
# @export
#
hsnorm = function( sig )
{
  return(sqrt(sum(abs(sig)^2)));
}
```

```
# p-Schatten Norm
#
# @param sig covariance matrix (i.e. symmetric positive definite)
# @param p [1,Inf] or 1/2
#
# @return p-Schatten Norm of sig
#
# @export
#
pschnorm = function( sig, p )
{
  if( p==2 )
    return( hsnorm(sig) );
  if(p==1/2)
    return( sqrt(pschnorm(sig,1)) );
  eigval = eigen( sig, symmetric=TRUE, only.values=TRUE );
  if( p==-1||is.infinite(p) )
    return( max( abs(eigval$values) ) );
  return( sum(abs(eigval$values)^p)^(1/p) );
}
```

```
# Computes Square Root of matrix A
#
# @param A matrix
#
# @return Square root of A
#
```

```
# @export
#
sqrtMat = function(A)
{
  eig = eigen( A );
  val = eig$values;
  #val = pmax( eig$values, rep(0,nrow(A)) );
  d = sqrt( as.complex(val) );
  D = diag( d );
  V = eig$vectors;
  return( V%*%D%*%t(V) );
}
```

Plot

```
#' Plot partial p-values
#'
#' \code{perm.plot} plots all of the partial comparison p-values in a matrix.
#'
#' @param p Output of function perm.test, if part = TRUE.
#' @param k Number of groups, must be greater than 2.
#' @param lab Group labels. Defaults to 1, 2, ..., k.
#' @param save Boolean variable that indicates if the plot must be saved as an .
    eps. Defaults to FALSE.
#' @param name If \code{save} is TRUE, this is the filename of the plot.
    Defaults to \code{pvalues.eps}.
#'
#' @return \code{perm.plot} plots the partial p-values in a matrix.
#'
#' @author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}
#'
#' @references Pigoli, Davide, John A. D. Aston, Ian L. Dryden, and Piercesare
    Secchi (2014). "Distances and inference for covariance operators."
    Biometrika: asu008.
#'
#' @examples
#' \dontrun{
#' ## Phoneme data
#'
#' library(fdcov)
#' library(fds)
#'
#' # Create data set
#' data(aa); data(ao); data(dcl);data(iy);data(sh)
#' dat=cbind(aa$y[,1:20],ao$y[,1:20],dcl$y[,1:20],iy$y[,1:20],sh$y[,1:20])
#' dat=t(dat)
#' grp=c(rep(1,20),rep(2,20),rep(3,20),rep(4,20),rep(5,20))
#'
#' # Test the equality of the covariance operators
#' p=ksample.perm(dat,grp,iter=100,only.glob=FALSE)
#'
#' # Plot partial p-values
#' perm.plot(p,5, lab=c('aa','ao','dcl','iy','sh'))
#' }
#'
#' @export

perm.plot = function(p, q, lab = NULL, save = FALSE, name = 'pvalues.eps'){

    if(q<3) stop('The number of groups comparisons must be at least 3.')

    # put the p-values in a q X q matrix
    pmatrix = matrix(1,q,q)
    cont = 1
    for(i in 1:(q-1)){
        for(j in (i+1):q){
            pmatrix[i,j] = p$partial$p_value[cont]
            cont = cont+1
        }
    }

    if(!is.null(lab)){ # if there are no group labels in input
        colnames(pmatrix) = rownames(pmatrix) = lab # assign labels "1", ..., "q"
    }

    # select palette
    col = c(0,0,0,0,'#b30000','#e34a33','#fc8d59','#fdcc8a','#fef0d9')

    # plot partial p-values in a matrix
    corrplot::corrplot(t(pmatrix),method = "color",type = "lower",tl.col='black',addCoef.col = "black", is.corr = FALSE, cl.lim = c(0,1), col = col, tl.pos = 'ld')

   # save plot in an external .eps file
   if(save){
        setEPS()
        postscript(name)
        corrplot::corrplot(t(pmatrix),method = "color",type = "lower",tl.col = ’black’,addCoef.col = "black",
           is.corr = FALSE, cl.lim = c(0,1), col=col, tl.pos = ’ld’)
        dev.off()
    }

}
```

B.2 Multiplicity control

Step-down max T procedure

```
# Step-down Tippett procedure for strong FWE control
#
# @param T (iter+1) X k matrix of permutation test statistics
#
# @return The global p-value and a vector of adjusted p-values
#
# @author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}

FWE.maxT = function(T){

  ord = order(T[1,],decreasing=TRUE) # put the vector of observed test staiistics in decreasing order and store the order
  T.ord = T[,ord] # put the columns of matrix T in the new order ’ord’

  k = dim(T)[2] # number of tests
  p.ris = array(5,dim=c(k,1)) # create vector of adjusted p-values

  # Compute smallest p-value
  Tcomb = apply(T.ord,1,max) # combine vectors of p-values with max comb. fct.
  p.ris[1] = p.glob=mean(Tcomb[-1] ≥Tcomb[1]) # the first adjusted p-value corresponds with the global p-value

  # Compute the other p-values
  if(k>2){ # apply general step-down algorithm for p-value adjustement
    for(j in 2:(k-1)){
        Tcomb = apply(T.ord[,j:k],1,max)
        p.ris[j] = max(mean(Tcomb[-1] ≥Tcomb[1]),p.ris[(j-1)])
      }
  }


  # Compute greatest p-value
  Tcomb = T.ord[,k]
  p.ris[k] = max(mean(Tcomb[-1] ≥Tcomb[1]),p.ris[k-1]) # last adjusted p-value

  # Put the ajusted p-values in the correct order
  p.ris[ord] = p.ris

  rownames(p.ris) = colnames(T)
  return(p.ris)
}
```

Step-down Tippett procedure

```
# Step-down Tippett procedure for strong FWE control
#
# @param P (iter+1) X k matrix of permutation p-values
#
# @return A vector of adjusted p-values
#
# @author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}

FWE.tipp = function(P){

    ord = order(P[1,],decreasing=FALSE) # sort the observed p-values in increasing order and store the order
    P.ord = P[,ord] # put the columns of matrix P in the new order orde

    k = dim(P)[2] # number of tests
    p.ris = array(5,dim=c(k,1)) # create vector of adjusted p-values

    Pcomb = apply(P.ord,1,min) # combine vectors of p-values with Tippett's comb. fct.
    p.ris[1] = p.glob = mean(Pcomb[-1]≤Pcomb[1]) # first adjusted p-value corresponds with the global p-value

    if(k>2){ # apply tippett step-down algorithm for p-value adjustement
        for(j in 2:(k-1)){
            T = apply(P.ord[,j:k],1,min)
            p.ris[j] = max(mean(Pcomb[-1]≤Pcomb[1]),p.ris[(j-1)])
        }
    }

    p.ris[k] = max(P.ord[1,k],p.ris[k-1]) # last adjusted p-value
    p.ris[ord] = p.ris # put the ajusted p-values in the right order
    rownames(p.ris) = colnames(P)

  return(p.ris)
}
```

Closed testing procedure

```
# Closed testing procedure for strong FWE control
#
# @param T (iter+1) X k matrix of permutation test statistics
# @param comb Combining function
#
# @return A vector of adjusted p-values
#
# @author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}
#
# @export

FWE.clos = function(T, comb = 'dire', loading = FALSE){

    #library(matlab)

    iter = dim(T)[1] # number of permutations
    k = dim(T)[2] # number of partial tests
    rows = ncycles = 2^k # total number of tests

    x = matrix(0,rows,k)

    if(loading) pb = txtProgressBar(min = 0, max = k, style = 3) # create
        progress bar
    for (i in 1:k){
        nreps = rows/ncycles
        ncycles = ncycles/2
        zo = matrix(c(0,1),nreps,2,byrow=TRUE)
        zoc = rbind(as.matrix(zo[,1]),as.matrix(zo[,2]))
        settings = matlab::repmat(zoc,c(1,ncycles))
        x[,k-i+1] = settings
        if(loading) setTxtProgressBar(pb, k) # update progress bar
    }
    if(loading) close(pb) # close progress bar
    x = x[-1,]

    print(T[1:10,])

    T2 = matrix(0,iter,(rows-1))
    for(j in 1:(rows-1)){
        # for (i in 1:(iter+1)){
            T2[,j] = switch(comb,
                                dire = apply(as.matrix(T[,x[j,]==1]),1,sum),
                                fish = apply(as.matrix(T[,x[j,]==1]),1,comb.fish),
                                lipt = apply(as.matrix(T[,x[j,]==1]),1,comb.lipt),
                                warning(’The selected combining function is not available’))
        # }
    }

    print(T2[1:5,1:5])

    rawP = apply(T2,2,perm.pval)
    adjP = rep(5,k)
    for(l in 1:k) adjP[l] = max(rawP[x[,l]==1])

    return(adjP)
}
```

B.3 Empirical bootstrap

```
# Empirical bootstrap multiple sample test for the equality of covariance operators
#
#@param x dataset
#@param y group labels
#@param B number of permutations
#@param permutation type of permutation, ’synchro’, ’paired’ or ’pooled’
#
#@return list(global=P.glob) global p-value
#@return if(onlyglob==FALSE) list(global=P.glob, partial=P.part) global and partial p-values
#
#@author Alessandra Cabassi i \email{alessandra.cabassi@mail.polimi.it}

bootstrap = function(x, y, B=1000, mean=FALSE, distance=’sq’, adjust=TRUE, combfun=’tippett’, onlyglob=TRUE){

    table_groups=table(y) # groups table
    C=length(table_groups) # number of groups
    K=C*(C-1)/2 # number of partial tests
    p=dim(x)[2] # number of samplings per function
    N=length(y) # total number of observations

    # Group names definition
    name=character()
    cont=1
    for(i in 1:(C-1)){
        for(j in (i+1):C){
            name[cont]=paste(i,j,sep="-")
            cont=cont+1
        }
    }

    # calculate the sample mean functions in each population and the residual functions
    group_mean = matrix(NA,N,p)
    residual = matrix(NA,N,p)
    pseudo_obs=matrix(NA,N,p)

    for(i in 1:N){
        group_mean[i,]=colMeans(x[y==y[i],],na.rm=TRUE)
        residual[i,] =x[i,]-group_mean[i,]
    }

    T=array(0,dim=c((B+1),K)) # test statistics vector initialization

    cont=1
    for(i in 1:(C-1)){
        for(j in (i+1):C){ # for each pair of groups
            # compute test statistic for initial data
             T[1,cont] = distCov(cov(dat[grp==i,],use=’pairwise’),cov(dat[grp==j,],use=’pairwise’),dist)
            cont=cont+1
        }
    }

    for(bb in 2:(B+1)){# B iterations
        pseudo_obs = group_mean + residual[sample(1:N, N, replace=TRUE),]
        cont=1
        for(i in 1:(C-1)){ # for each pair of groups
            for(j in (i+1):C){ # apply permutation and compute test statistic
                T[bb,cont] = distCov(cov(pseudo_obs[grp==i,],use=’pairwise’),cov(pseudo_obs[grp==j,],use=’pairwise’),dist)
                cont=cont+1
            }
        }
    }
    return(compute_pvalue(P, T, C, combfun, adjust,onlyglob))
}#end MC algorithm
```

B.4 Generalisation of Levene's test

```
# Generalisation of Levene's test: ANOVA analysis on the Euclidean distances from individual points to the group centroid
#
#@param x dataset
#@param y group labels
#@param B number of permutations
#
#@return list(global=P.glob, partial=P.part) # global and partial p-values
#
#@author Alessandra Cabassi \email{alessandra.cabassi@mail.polimi.it}
#

anderson_test=function(x,y,B=1000){

    table_groups=table(y) # groups table
    C=length(table_groups) # number of groups
    K=C*(C-1)/2 # number of partial tests
    p=dim(x)[2] # number of samplings per function
    N=length(y) # total number of observations

    z=rep(0,N)
    for(i in 1:N){
        group=y[i]
        z[i]=Fdist(x[i,],colMeans(x[y==group,],na.rm=TRUE))
    }

    fit = aov(z ~y)
    T0 = summary(fit)[[1]][1,4]

    T_stat = numeric(B)

    for(perm in 1:B){
        perm = sample(1:N) # choose permutation
        z_perm = z[perm] # apply permutation
        fit_perm = aov(z_perm ~y) # use anova test on permuted data
        T_stat[perm] = summary(fit_perm)[[1]][1,4] # save test statistic
    } ## end MC algorithm

    p_val = sum(T_stat≥T0)/B # p-value
}
```

List of algorithms

List of Figures

Notation

t Element of $I \in \mathbb{R}$. 8

J Number of points of the measurement grid. 8

I Compact subset of $\mathbb{R}$. 8

$\mathbb{R}$ Set of real numbers. 8

X Random variable. 8

x Observation of random variable X. 8

n Sample size. 8

$\mathcal{H}$ Generic Hilbert space. 9

$\mathbb{P}$ Probability measure. 9

μ Mean function. 9

$\mathbb{E}$ Expectation operator. 9

Σ Covariance operator. 9

y Generic element of an Hilbert space. 9

m Sample mean. 9

S Sample covariance operator. 10

L^2 Hilbert space of square integrable functions. 10

l Lebesgue measure. 10

γ Variance. 10

Γ Autocovariance. 10

σ Hilbert-Schmidt kernel of covariance operator Σ. 10

s Element of $I \in \mathbb{R}$. 10

c Sample variance function. 11

C Sample variance function. 11

B_1 Closed ball of unitary radius in $L^2(I)$. 11

K Compact bounded linear operator from $L^2(I)$ to $L^2(I)$. 11

ρ Singular value of a compact bounded linear operator K. 12

$\mathbb{N}$ Set of natural numbers. 12

ϕ Eigenvalue of a compact bounded linear operator K. 12

$\mathcal{S}$ Space of the trace class operators on $L^2(I)$. 12

$\mathcal{L}$ Space of the linear bounded operators on $L^2(I)$. 13

v Eigenfunctions of the covariance operator Σ. 14

U Unitary bounded linear operator on $L^2(I)$. 14

$'$ Adjoint of a linear operator. 14

D Linear operator from $L^2(I)$ to $L^2(I)$ such that $\Sigma = DD'$. 14

$\mathcal{O}$ Space of unitary operators on $L^2(I)$. 14

V_p p-dimensional subspace of $L^2(I)$. 15

ϵ Observational error of curve x at time t_j: $y_j = x(t_j) + \epsilon_j$. 17

F Smoothing operator. 17

h Warping function. 20

$\mathcal{I}$ Set of absolutely continuous functions on $[0,1]$. 22

$\mathcal{X}$ Sample space of X. 28

P Distribution of random variable X. 28

$\mathcal{P}$ Family of probability distributions P. 28

A Generic element of the σ-algebra $\mathcal{A}$. 28

$\mathcal{A}$ σ-algebra of events in $\mathcal{X}$. 28

$\mathbf{X}$ Random data set. 28

$*$ Permuted vector of observations or labels. 28

$\mathbf{u}$ Vector of data set unit labels. 28

Δ Stochastic treatment effect. 28

$\overset{d}{>}$ Stochastic dominance. 28

$\mathcal{X}^n_{/\mathbf{X}}$ Conditional reference space given data set $\mathbf{X}$ (also indicated as $\mathcal{X}_{/\mathbf{X}}$). 29

M Cardinality of $\mathcal{X}_{/\mathbf{X}}$. 29

$\mathbb{1}$ Indicator function. 29

$\overset{d}{=}$ Equality in distribution. 29

$\mathbf{x}$ Observed data set. 30

T Test statistic. 30

$\mathcal{T}_{\mathbf{X}}$ Set of all possible values assumed by T as $\mathbf{X}^*$ varies in $\mathcal{X}_{/\mathbf{X}}$. 30

$\mathbf{T}$ Vector of test statistics $(T_1, \dots, T_k) : \mathcal{X}^n \to \mathbb{R}^k$. 30

α Significance level of a statistic test. 30

T^0 Test statistic evaluated on data set $\mathbf{X}$. 30

$\Lambda_{\mathbf{X}}$ Set of the step points of the significance level function associated to $\mathbf{X}$. 30

$L_{\mathbf{X}}$ Significance level function of a permutation test based on data set $\mathbf{X}$. 30

λ p-value. 31

B Number of Monte Carlo iterations. 31

$\hat{}$ Quantity estimated via Monte Carlo. 32

k Number of sub-hypotheses considered for the NPC method. 32

Ψ Combining function for the NPC method. 34

$\mathcal{C}$ Set of all combining functions satisfying Properties (3.1), (3.2) and (3.3). 34

Φ Inverse of the normal cumulative distribution function. 34

$\mathbf{c}$ Sample centroid. 40

$\otimes$ Tensor product in $L^2(I)$. 41

$\mathcal{F}$ Hilbert space of Hilbert-Schmidt operators. 41

Υ Covariance operator of U. 41

$\tilde{\otimes}$ Tensor product in $\mathcal{F}$. 41

$\xrightarrow{d}$ Convergence in distribution. 42

ι Eigenvalues of Υ. 42

Z Standard normal random variable. 42

$\mathcal{N}$ Normal distribution. 44

ε Residual function of x with respect to the mean μ. 44

Ξ Bounded linear operator mapping $L^2(I)$ to $L^2(I)$ s.t. $\|\Gamma\|_{\frac{p}{1-p}} \leq 1$. 47

ν Number of units exchanged between two data samples. 55

Λ Random variable associated to the p-value λ. 61

ξ Parameter(s) of the distribution of the missing data process. 90

$\mathbf{O}$ Set of observation indicators corresponding to dataset $\mathbf{X}$. 91

O Inclusion indicator corresponding to random variable X. 91

κ Actual sample size of valid data. 91

Abbreviations

FDA Functional Data Analysis. 7

MDA Multivariate Data Analysis. 7

OODA Object Oriented Data Analysis. 7

PENSSE Penalised Sum of Squared Errors. 18

MSE Mean Squared Error. 19

GCV Generalised Cross-Validation measure. 20

SRSF Square-Root Slope Function. 22

MC Monte Carlo. 31

NPC Non-Parametric Combination. 32

ANOVA Analysis of Variance. 32

UI Union-Intersection. 51

CSPs Constrained Synchronized Permutations. 55

USPs Unconstrained Synchronized Permutations. 55

FWE Family-Wise Error Rate. 59

FWEC Family-Wise Error Rate in the weak sense. 60

FWEP Family-Wise Error Rate in the strong sense. 60

STPs Simultaneous Test Procedures. 61

SP Subset Pivotality. 63

MAR Missing At Random. 90

OAR Observed At Random. 90

MCAR Missing Completely At Random. 90

Bibliography

Anderson, M. J. (2006). Distance-based tests for homogeneity of multivariate dispersions. *Biometrics*, 62(1):245–253.

Boente, G., Rodriguez, D., and Sued, M. (2014). A test for the equality of covariance operators. *arXiv preprint arXiv:1404.7080*.

Bonferroni, C. E. (1936). *Teoria statistica delle classi e calcolo delle probabilità*. Libreria internazionale Seeber.

Bosq, D. (2012). *Linear processes in function spaces: theory and applications*, volume 149. Springer Science & Business Media.

Cabassi, A. and Kashlak, A. B. (2016). *fdcov: Analysis of Covariance Operators*. R package version 1.0.0.

Cardot, H., Ferraty, F., Mas, A., and Sarda, P. (2003). Testing hypotheses in the functional linear model. *Scandinavian Journal of Statistics*, 30(1):241–255.

Casella, G. and Berger, R. L. (2002). *Statistical inference*, volume 2. Duxbury Pacific Grove, CA.

Craven, P. and Wahba, G. (1978). Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403.

Cuevas, A., Febrero, M., and Fraiman, R. (2004). An anova test for functional data. *Computational statistics & data analysis*, 47(1):111–122.

Dauxois, J., Pousse, A., and Romain, Y. (1982). Asymptotic theory for the principal component analysis of a vector random function: some applications to statistical inference. *Journal of multivariate analysis*, 12(1):136–154.

De Boor, C. (2002). Spline basics. *Handbook of Computer Aided Geometric Design, Elsevier*, 13:141–164.

Diaconis, P. and Efron, B. (1983). Computer intensive methods in statistics. Technical Report 196, Stanford University. Division of Biostatistics.

Dryden, I. L., Koloydenko, A., and Zhou, D. (2009). Non-euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *The Annals of Applied Statistics*, pages 1102–1123.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Annals of Statistics*, 7:1–26.

El Karoui, N. (2008). Operator norm consistent estimation of large-dimensional sparse covariance matrices. *The Annals of Statistics*, pages 2717–2756.

Fan, J. and Lin, S.-K. (1998). Test of significance when data are curves. *Journal of the American Statistical Association*, 93(443):1007–1021.

Ferraty, F. and Vieu, P. (2006). *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media.

Fremdt, S., Steinebach, J. G., Horvath, L., and Kokoszka, P. (2013). Testing the equality of covariance operators in functional samples. *Scandinavian Journal of Statistics*, 40(1):138–152.

Giné, E. and Nickl, R. (2015). Mathematical foundations of infinite-dimensional statistical models. *Cambridge Series in Statistical and Probabilistic Mathematics*.

Good, P. I. (2005). Permutation, parametric, and bootstrap tests of hypotheses. *Journal of the American Statistical Association*, 100(472):1457–1458.

Green, P. J. and Silverman, B. W. (1993). *Nonparametric regression and generalized linear models: a roughness penalty approach*. CRC Press.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models*, volume 43. CRC Press.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70.

Horváth, L. and Kokoszka, P. (2012). *Inference for functional data with applications*, volume 200. Springer Science & Business Media.

Kashlak, A. B., Aston, J. A., and Nickl, R. (2016). Inference on covariance operators via concentration inequalities: k-sample tests, classification, and clustering via rademacher complexities. *arXiv preprint arXiv:1604.06310*.

Lehmann, E. L. and Romano, J. P. (2005). *Testing Statistical Hypotheses*. Springer Texts in Statistics.

Levene, H. (1960). Robust tests for equality of variances. *Contributions to probability and statistics: Essays in honor of Harold Hotelling*, 2:278–292.

Lipták, T. (1958). On the combination of independent tests. *Magyar Tud Akad Mat Kutato Int Kozl*, 3:171–197.

Manly, B. F. (2006). *Randomization, bootstrap and Monte Carlo methods in biology*, volume 70. CRC Press.

Marcus, R., Eric, P., and Gabriel, K. R. (1976). On closed testing procedures with special reference to ordered analysis of variance. *Biometrika*, 63(3):655–660.

Marron, J. S., Ramsay, J. O., Sangalli, L. M., Srivastava, A., et al. (2014). Statistics of time warpings and phase variations. *Electronic Journal of Statistics*, 8(2):1697–1702.

Panaretos, V. M., Kraus, D., and Maddocks, J. H. (2010). Second-order comparison of gaussian random functions and the geometry of dna minicircles. *Journal of the American Statistical Association*, 105(490):670–682.

Paparoditis, E. and Sapatinas, T. (2014). Bootstrap-based k-sample testing for functional data. *arXiv preprint arXiv:1409.4317*.

Pesarin, F. and Salmaso, L. (2010). *Permutation tests for complex data: theory, applications and software*. John Wiley & Sons.

Pigoli, D., Aston, J. A., Dryden, I. L., and Secchi, P. (2014). Distances and inference for covariance operators. *Biometrika*, 101(2):409–422.

Quenouille, M. H. (1949). Approximate tests of correlation in time-series. *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(1):68–84.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ramsay, J. O. and Dalzell, C. (1991). Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 539–572.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer Series in Statistics.

Robinson, D. T. (2012). *Functional data analysis and partial shape matching in the square root velocity framework*. PhD thesis, The Florida State University.

Romano, J. P., Shaikh, A. M., and Wolf, M. (2008). Formalized data snooping based on generalized error rates. *Econometric Theory*, 24(02):404–447.

Roy, S. N. (1953). On a heuristic method of test construction and its use in multivariate analysis. *The Annals of Mathematical Statistics*, pages 220–238.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Sen, P. K. (2007). Union–intersection principle and constrained statistical inference. *Journal of Statistical Planning and Inference*, 137(11):3741–3752.

Shang, H. L. and Hyndman, R. J. (2013). *fds: Functional data sets*. R package version 1.7.0.

Shen, Q. and Faraway, J. (2004). An f test for linear models with functional responses. *Statistica Sinica*, pages 1239–1257.

Šidák, Z. (1967). Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62(318):626–633.

Solari, A., Salmaso, L., Pesarin, F., and Basso, D. (2009). *Permutation Tests for Stochastic Ordering and ANOVA*. Springer New York.

Swallow, J. G., Carter, P. A., and Garland Jr, T. (1998). Artificial selection for increased wheel-running behavior in house mice. *Behavior genetics*, 28(3):227–237.

Talagrand, M. (1996). New concentration inequalities in product spaces. *Inventiones mathematicae*, 126(3):505–563.

Tucker, J. D. (2016). *fdasrvf: Elastic Functional Data Analysis*. R package version 1.6.1.

Tucker, J. D., Wu, W., and Srivastava, A. (2013). Generative models for functional data using phase and amplitude separation. *Computational Statistics & Data Analysis*, 61:50–66.

Wang, H., Marron, J., et al. (2007). Object oriented data analysis: Sets of trees. *The Annals of Statistics*, 35(5):1849–1873.

Westfall, P. H. and Troendle, J. F. (2008). Multiple testing with minimal assumptions. *Biometrical Journal*, 50(5):745–755.

Westfall, P. H. and Young, S. S. (1993). *Resampling-based multiple testing: Examples and methods for p-value adjustment*, volume 279. John Wiley & Sons.

Zhu, K. (2007). *Operator theory in function spaces*. American Mathematical Society.