Modelling function-valued processes with complex structure

Evandro Konzen

Thesis submitted for the degree of
Doctor of Philosophy

School of Mathematics, Statistics & Physics
Newcastle University
Newcastle upon Tyne
United Kingdom

May 2019

Abstract

Existing approaches to functional principal component analysis (FPCA) usually rely on nonparametric estimation of the covariance structure. When function-valued processes are observed on a multidimensional domain, the nonparametric estimation suffers from the curse of dimensionality, forcing FPCA methods to make restrictive assumptions such as covariance separability.

In this thesis, we discuss a general Bayesian framework on modelling function-valued processes by using a Gaussian process (GP) as a prior, enabling us to handle nonseparable and/or nonstationary covariance structure. The nonstationarity is introduced by a convolution-based approach through a varying kernel, whose parameters vary along the input space and are estimated via a local empirical Bayesian method. For the varying anisotropy matrix, we propose to use a spherical parametrisation, leading to unconstrained and interpretable parameters and allowing for interaction between coordinate directions in the covariance function. The unconstrained nature allows the parameters to be modelled as a nonparametric function of time, spatial location and even additional covariates.

In the spirit of FPCA, the Bayesian framework can decompose the function-valued processes using the eigenvalues and eigensurfaces calculated from the estimated covariance structure. A finite number of the eigensurfaces can be used to extract some of the most important information involved in data with complex covariance structure.

We also extend the methods to handle multivariate function-valued processes. The estimated covariance structure is shown to be important to analyse joint variation in the data and is further used in our proposed multiple functional partial least squares regression model. We show that the interaction between the scalar response variable and function-valued covariates can be explained by fewer terms than in a regression model which uses multivariate functional principal components.

Simulation studies and applications to real data show that our proposed approaches provide new insights into the data and excellent prediction results.

To my parents, Geraldo and Edir

Acknowledgements

I wish to express my gratitude and thanks to my supervisor, Dr. Jian Qing Shi, for the brilliant supervision during my PhD years. He has always encouraged me to explore my research interests and provided me excellent guidance. His dedication and energy towards the research project have motivated me since the beginning. Many thanks to Dr. Jian Qing Shi for being always available when I needed him.

I am also very grateful to the School of Mathematics, Statistics and Physics for providing me all the necessary facilities and resources to conduct this research. I am also grateful to the School for the funding which made this research possible. Many thanks to John and Dr. Michael Beaty for their help with computational issues.

I must also extend my gratitude to Amos, Maryam, Yingying and Omar, my office mates during most of my PhD years. They were all great. Thanks to geordies for the warm welcome when I moved to Newcastle. Thanks to Andrew for the football matches at St. James' Park.

I take this opportunity to thank my ex-supervisor and friend Dr. Flavio Ziegelmann. He has encouraged me to do PhD abroad and helped me to grow personally and academically.

Huge thanks to my friends from Brazil, Sandro, Magnus, Ederson, Jordana, Andre Luis, Raquel, Felipe and Christopher, for their patience and support.

I would like to thank Eliza, for her love, companionship and help. Without her help, I probably would not be where I am today.

I also wish to thank all my uncles, aunts, and cousins for their love and support.

Finally, I would like to express special thanks to my parents, Geraldo and Edir, for their love, patience, reassurance and encouragement despite the distance.

Contents

List of Figures

List of Tables

List of Abbreviations

CFVE	cumulative fraction of variance explained
FDA	functional data analysis
FPC	functional principal component
FPCA	functional principal component analysis
FPCR	functional principal component regression
FPLS	functional partial least squares
FPLSR	functional partial least squares regression
GP	Gaussian process
GPR	Gaussian process regression
KL	Karhunen-Loève orthogonal expansion
LLE	local likelihood estimation
MFPCA	multivariate functional principal component analysis
MFPCR	multivariate functional principal component regression
MFPLSR	multiple functional partial least squares regression
MGP	multivariate Gaussian process
MSLL	mean standardised log loss
NSGP	nonstationary Gaussian process
NSMGP	nonstationary multivariate Gaussian process
RMSE	root mean squared error
RSMSE	root standardised mean squared error
SD	subset of data
SMSE	standardised mean squared error
SNR	signal-to-noise ratio
SR	subset of regressors

List of Symbols

δ_{ij}	Kronecker delta. $\delta_{ij} = 1$ if $i = j$ and 0 otherwise
J	number of eigenfunctions (or principal components)
$k(\cdot, \cdot)$	covariance function $k(\cdot, \cdot; \boldsymbol{\theta})$
$\boldsymbol{K}_n$	$n \times n$ covariance matrix whose (i, j)-th entry is $\left[\boldsymbol{K}_n\right]_{ij} = k(\boldsymbol{t}_i, \boldsymbol{t}_j)$
$\otimes$	Kronecker product
m	subset size ($m < n$) used in approximate implementation methods
M	number of elements in the multivariate function-valued process $\boldsymbol{X}(\cdot)$
$\mu(\cdot)$	mean function of $X(\cdot)$
N	number of replicated scalar/functional observations
n	number of observations on each curve/surface
$\|\| \cdot \|\|$	norm of the vector $\boldsymbol{x} = (x_1, \ldots, x_p)^\top$, i.e. $\|\|\boldsymbol{x}\|\| = \left(\sum_{i=1}^p x_i^2\right)^{1/2}$ or norm of the function $f \in L^2(\mathcal{T})$, i.e. $\|\|f\|\| = \langle f, f \rangle^{1/2}$
$\mathcal{O}$	computational complexity
Ξ	(covariance) operator
$\boldsymbol{\Psi}_n$	$n \times n$ covariance matrix whose (i, j)-th entry is $\left[\boldsymbol{\Psi}_n\right]_{ij} = k(\boldsymbol{t}_i, \boldsymbol{t}_j) + \sigma_\varepsilon^2 \delta_{ij}$
σ_ε^2	variance of the measurement error ε
$\boldsymbol{\Sigma}(\boldsymbol{\tau})$	$\boldsymbol{\tau}$-varying anisotropy matrix
$\mathcal{T}$	input domain $\mathcal{T} \subset \mathbb{R}^Q$
$\boldsymbol{t}$	Q-dimensional input
t	unidimensional input
$X(\cdot)$	function-valued process with input $t \in \mathcal{T} \subset \mathbb{R}$ or $\boldsymbol{t} \in \mathcal{T} \subset \mathbb{R}^Q$
$\boldsymbol{X}(\cdot)$	multivariate function-valued process $\boldsymbol{X}(\cdot) = \left(X_1(\cdot), \ldots, X_M(\cdot)\right)^\top$
Y	scalar response variable

Chapter 1

Introduction

In functional data analysis (FDA), data are seen as discretely observed functions (of time, space, etc.) and are therefore called functional data. Such functions can be seen as realisations of a function-valued stochastic process $X(\boldsymbol{t})$, $\boldsymbol{t} \in \mathcal{T} \subset \mathbb{R}^Q$, which has mean function $\mu(\boldsymbol{t})$ and covariance function $\mathrm{Cov}\big[X(\boldsymbol{t}), X(\boldsymbol{t}')\big]$.

Typical applications of FDA include data observed on the one-dimensional domain $\mathcal{T} \subset \mathbb{R}$ (e.g. time series, growth curves). Nowadays, there is an increasing interest in data observed on two-dimensional (e.g. spatial data, 2D images), three-dimensional (3D images), four-dimensional (fMRI data) or even higher dimensional domains. In the multidimensional settings, the nonparametric estimation of the covariance function suffers from the curse of dimensionality and FDA methods are often forced to make restrictive assumptions such as covariance separability (e.g. in Chen *et al.* (2017)). This limits the application to many types of data with complex covariance structure.

1.1 Aims

In this thesis, we discuss a general Bayesian framework on modelling function-valued processes by using a Gaussian process or other heavy-tailed processes as a prior, allowing nonseparable and/or nonstationary covariance structure.

Modelling nonstationary and/or nonseparable covariance structure

The nonstationarity is introduced by a convolution-based approach (Higdon *et al.*, 1999) via a varying kernel. In particular, the nonstationarity can be simply defined by a $Q \times Q$ varying anisotropy matrix $\boldsymbol{\Sigma}(\boldsymbol{t})$ and a standard deviation $\sigma(\boldsymbol{t})$, both varying along $\boldsymbol{t} \in \mathcal{T} \subset \mathbb{R}^Q$ or along $\boldsymbol{\tau} \in \mathcal{T}^* \subset \mathbb{R}^{Q^*}$, where $Q^* \leq Q$. A local empirical Bayesian

approach is used to estimate the hyperparameters involved in the modelling of covariance structure, including both fixed and varying coefficients. For the varying anisotropy matrix $\boldsymbol{\Sigma}(\boldsymbol{t})$, we propose to use a spherical parametrisation in order to have unconstrained and interpretable parameters. In addition, this parametrisation allows for interaction between coordinates of $\boldsymbol{t}$, thus producing a nonseparable covariance. The unconstrained nature of the parameters allows them to be modelled as a nonparametric function of time (or spatial location), time (or spatially) dependent covariates and even additional covariates.

The Bayesian framework provides an efficient approach for obtaining the predictive distribution for the unknown underlying regression function of the processes; in the meantime, it can also decompose the function-valued processes using the eigenvalues and eigensurfaces calculated from the estimated covariance structure. In the spirit of functional principal component analysis, a finite number of the eigensurfaces can be used to extract some of the most important and interpretable information involved in different types of data with complex structure.

Multivariate function-valued processes

Recent interest has been also given to the analysis of multivariate functional data, where the samples consist of vectors of functional data and are modelled by a multivariate random process $\boldsymbol{X}(\cdot) = \big(X_1(\cdot), \ldots, X_M(\cdot)\big)^\top$. In the multivariate case, a major difficulty consists in defining cross-covariance functions ensuring that resulting covariance matrices for the multivariate process are positive definite. Multivariate functional principal component analysis (MFPCA) (Happ & Greven, 2018) is a recent approach that addresses joint variation of multiple functions. In order to explicitly model the cross-covariance structure between multiple functions, we extend a bivariate convolved Gaussian processes model (Boyle & Frean, 2004) to the multivariate case, called multivariate GP (MGP), ensuring the positive-definiteness property. This is done by considering that each individual random process is represented as the sum of an independent latent process and another latent process common to all M random functions.

Multiple functional PLS regression model

The estimation of the covariance structure of multivariate random processes is also important for further statistical analysis, e.g. in a scalar-on-functions regression which involves multiple function-valued covariates. This has motivated us to apply our MGP methodology as a building block to extend the functional partial least squares regression model (Delaigle & Hall, 2012) to the case of multiple functional covariates. Our proposed model is hereafter called the multiple functional partial least squares regression (MF-

PLSR) model and is an alternative to the multivariate functional principal component regression (MFPCR) model.

1.2 Structure of the thesis

Chapter 2 provides an introduction to functional data analysis and related statistical concepts used throughout the thesis. It also describes recent developments and challenges of some FDA tools, namely the functional principal components and the scalar-on-function regression model.

Chapter 3 introduces a nonparametric Bayesian framework which uses a Gaussian process prior. Besides introducing Gaussian process regression (GPR) models and related inference procedures, we discuss properties and interpretation of commonly used parametric covariance function families. Decomposition of GP and asymptotic theory are also provided. The end of the chapter is dedicated to discussing approximate implementation methods used to reduce computational costs, showing results in a simulation study. Via another simulation study we discuss the decomposition of GP.

In Chapter 4, the focus is on modelling function-valued processes defined on a multidimensional domain. In order to avoid assumptions of stationarity and covariance separability, we propose a (semi)parametric approach for the estimation of the covariance function. Numerical simulation studies and an application to Canadian temperature are provided.

Chapter 5 presents the MGP to deal with multivariate function-valued processes. Through simulation studies, we show that the joint modelling improves prediction performance. In addition, we compare prediction results with the ones obtained by MFPCA. Finally, an application to human fertility data is given.

In Chapter 6, we propose the MFPLSR model, where the covariance function of multiple function-valued predictors is estimated by MGP. A simulation study compares MFPLSR and MFPCR in terms of prediction of the scalar response variable.

In Chapter 7, we highlight the main contributions, conclusions and future work.

Chapter 2

Statistical concepts for function-valued processes

This chapter introduces the statistical concepts of functional data analysis framework used in this thesis. It also discusses important tools, their recent developments and challenges. Section 2.1 introduces the notion of functional data and related mathematical tools. Section 2.2 explains how we interpret functional data as stochastic processes. Finally, Sections 2.3 and 2.4 introduce two of the main tools of functional data analysis, namely the functional principal component analysis and the scalar-on-function regression model.

2.1 Functional data

Functional data analysis (FDA) deals with the analysis and theory of data that are in the form of functions, images, shapes, or more general objects (Wang *et al.*, 2016). We assume that the observed data $x_1, \ldots, x_n$ are realisations of an underlying continuous stochastic process X defined on $\mathcal{T} \subset \mathbb{R}^Q$, $Q \geq 1$. We will say that the functional data is *multidimensional* when $Q > 1$. In FDA, the terminology *multi-way* functional data is also used.

With the progress of technology, it has become common to find applications involving multiple functions (elements) per subject, and this has motivated the development of methodologies to consider simultaneous variation in the multiple elements. We will denote the M-variate functional data by $\boldsymbol{X}(\boldsymbol{t}) = \big(X_1(\boldsymbol{t}), \ldots, X_M(\boldsymbol{t})\big)^\top$, a topic discussed in Chapters 5 and 6.

To conduct inference and develop theory, the functional objects are elements of the space of all square integrable functions, L^2, a space whose main concepts are defined in

the next section.

2.1.1 Square integrable functions

We say that a function f is square integrable on $\mathcal{T}$ if $\int_{\mathcal{T}} f^2(\boldsymbol{t})d\boldsymbol{t} < \infty$. The set of all square integrable functions is denoted by L^2. It is very convenient to work on the space L^2 because it is associated with the inner product of two functions, f and g, by

$$\langle f, g\rangle = \int_{\mathcal{T}} f(\boldsymbol{t})g(\boldsymbol{t})d\boldsymbol{t}, \quad f, g \in L^2(\mathcal{T}). \tag{2.1}$$

This inner product is clearly analogous to the inner product of two vectors. If $\langle f, g\rangle = 0$, we say that functions f and g are orthogonal. The inner product (2.1) also provides the induced norm

$$||f|| = \sqrt{\langle f, f\rangle}, \quad f \in L^2(\mathcal{T}),$$

which is analogous to the length of a finite dimensional vector. If f and g are orthogonal and, in addition, $||f|| = ||g|| = 1$, they are said to be orthonormal.

Finally, the distance between two functions is the norm of their difference, that is, $d(f, g) = ||f - g||$. Therefore, the inner product (2.1) allows us to use analogous concepts of a finite dimensional Euclidean space to a function space.

2.1.2 Linear operators

In this section, we follow closely some definitions and understanding presented in Horváth & Kokoszka (2012) and Kokoszka & Reimherr (2017).

Linear transformations, often called linear operators or simply operators, are important in FDA because many types of regression involve linear transformations of functions to functions, functions to scalars and vectors to functions. A commonly used operator in FDA is the covariance operator Ξ, which transforms a function $\phi \in L^2$ to $\Xi(\phi)$ by solving the integral

$$\Xi(\phi)(\boldsymbol{t}) = \int \phi(\boldsymbol{t}')k(\boldsymbol{t}, \boldsymbol{t}')d\boldsymbol{t}',$$

where $k(\cdot, \cdot)$ is called the kernel of the operator Ξ. If $\int\int k^2(\boldsymbol{t}, \boldsymbol{t}')d\boldsymbol{t}d\boldsymbol{t}' < \infty$, the operator Ξ is called a Hilbert-Schmidt operator and is square integrable whenever ϕ is so.

Ξ is said to be *symmetric* if, for every ϕ and η, $\langle \Xi(\phi), \eta\rangle = \langle \phi, \Xi(\eta)\rangle$, and it is positive semi-definite if, for every $\phi \neq 0$, $\langle \Xi(\phi), \phi\rangle \geq 0$. These two definitions must be satisfied to show that the kernel $k(\cdot, \cdot)$ can be expanded into

$$k(\boldsymbol{t}, \boldsymbol{t}') = \sum_{j=1}^{\infty} \lambda_j \phi_j(\boldsymbol{t}) \phi_j(\boldsymbol{t}'), \tag{2.2}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ are the eigenvalues and $\phi_1, \phi_2, \ldots$ are the eigenfunctions of Ξ, that is, $\Xi(\phi_j) = \lambda_j \phi_j$. The result (2.2) is known as Mercer's theorem (Adler & Taylor, 2007).

2.2 Function-valued stochastic processes

Let $(\Omega, \mathcal{F}, P)$ be a probability space, where Ω is the set of all possible outcomes, $\mathcal{F}$ is a σ-algebra of subsets of Ω, and P is a probability measure over $\mathcal{F}$ (Bosq, 2000).

Let us consider a stochastic process $\{X_t(\boldsymbol{\omega}) : t \in \mathcal{T}\}$ defined on $(\Omega, \mathcal{F}, P)$, where $\mathcal{T} \subset \mathbb{R}$ for simplicity of exposition, and let $\mathcal{S}$ be a metric space. The mapping $X : \Omega \to \mathcal{S}$ is a random element in $\mathcal{S}$. If $\mathcal{S} = \mathbb{R}$, the real line, then X is a random variable. If $\mathcal{S} = \mathbb{R}^p$, the p-dimensional Euclidean space, then X is a random vector. If $\mathcal{S}$ is a function space (e.g. L^2), X is a random function.

The stochastic process $\{X_t(\boldsymbol{\omega}) : t \in \mathcal{T}\}$ can be thought of as random variables indexed by $t \in \mathcal{T}$. For a fixed t, $X_t(\boldsymbol{\omega})$ is a *random variable*. A stochastic process can also be interpreted as a random variable taking values in a function space (Bosq, 2000). That is, for a fixed $\boldsymbol{\omega}$, the *random function* (which is now observed as $\boldsymbol{\omega}$ is fixed) is a realisation of the stochastic process over $t \in \mathcal{T}$. In this case, note that the index set $\mathcal{T}$ of the stochastic process is the input space of the random function.

In this thesis, we use the terminology *function-valued (stochastic) processes* to highlight that the random variable takes values in a function space.

Note that stochastic processes described as above are defined on a unidimensional index set $\mathcal{T} \subset \mathbb{R}$ but can also be defined on a multidimensional index set $\mathcal{T} \subset \mathbb{R}^Q$, $Q > 1$, where they are also known as *random fields*.

We say that the stochastic process X is *integrable* if $\mathrm{E}||X|| = \mathrm{E}\big[\int X^2(\boldsymbol{t}) d\boldsymbol{t}\big]^{1/2} < \infty$ and *square integrable* if $\mathrm{E}||X||^2 = \mathrm{E}\big[\int X^2(\boldsymbol{t}) d\boldsymbol{t}\big] < \infty$ (Horváth & Kokoszka, 2012). These conditions ensure that the mean function and the covariance function exist and are defined, respectively, by

$$\mu(\boldsymbol{t}) = \mathrm{E}\big[X(\boldsymbol{t})\big]$$

and

$$\mathrm{Cov}\big[X(\boldsymbol{t}), X(\boldsymbol{t}')\big] = \mathrm{E}\big[\big(X(\boldsymbol{t}) - \mu(\boldsymbol{t})\big)\big(X(\boldsymbol{t}') - \mu(\boldsymbol{t}')\big)\big]. \tag{2.3}$$

We can estimate $\mu(\boldsymbol{t})$ by pooling the data from all individuals. If the sampling points

are densely, regularly observed, a common strategy is to simply use the empirical mean estimator, which is obtained by averaging over all the individuals and interpolating the resulting points. For sparse designs, though, Yao *et al.* (2005) suggest using a local linear smoother (Fan & Gijbels, 1996).

Likewise, for the covariance function (2.3), if the sampling points are densely, regularly observed, we could use the empirical covariance estimator and proceed with smooth interpolation of the sample estimates to estimate the covariance function in the entire interval. This estimator cannot be used if data are irregularly sampled and, even if that is not the case, its estimates are usually noisy. Therefore, other estimators for (2.3) have been proposed.

Two-dimensional kernel smoothing is usually suggested to nonparametrically fit the covariance function of functional data defined on $\mathcal{T} \subset \mathbb{R}$ (e.g. in Yao *et al.* (2005); Hall *et al.* (2006); Li *et al.* (2010)). A disadvantage is that smoothing estimates do not guarantee positive definiteness of the covariance function; to remedy this, eigendecomposition is used and negative eigenvalues are set to zero. Moreover, if the input space is multidimensional (i.e. $Q > 1$), one needs to use to a high-dimensional kernel smoothing, which results in slow computing and curse of dimensionality. To handle multidimensional function-valued processes, Chen *et al.* (2017) propose a tensor product representation, which will be discussed in Chapter 4. A parametric covariance function can also be used and is selected from many existing families, as we will see in Chapters 3 and 4.

The accurate estimation of covariance function (2.3) is of crucial importance to functional principal component analysis (FPCA), which is discussed in the next section.

2.3 Functional principal component analysis

According to the Karhunen-Loève (KL) orthogonal expansion (Wahba, 1990), the centred random function $X^c(\boldsymbol{t}) = X(\boldsymbol{t}) - \mu(\boldsymbol{t})$ can be decomposed into

$$X^c(\boldsymbol{t}) = \sum_{j=1}^{\infty} \phi_j(\boldsymbol{t})\xi_j, \tag{2.4}$$

where ξ_j are uncorrelated random variables and $\phi_j(\cdot)$ are the eigenfunctions of the covariance operator of X. That is, $\phi_j(\cdot)$ are the solutions to the eigenequations

$$\int k(\boldsymbol{t}, \boldsymbol{t}')\phi_j(\boldsymbol{t}')d\boldsymbol{t}' = \lambda_j\phi_j(\boldsymbol{t}), \quad \int \phi_i(\boldsymbol{t})\phi_j(\boldsymbol{t})d\boldsymbol{t} = \delta_{ij}, \tag{2.5}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ are the eigenvalues of $k(\cdot, \cdot)$ and δ_{ij} is the Kronecker delta. Each function $\phi_j(\cdot)$ is called functional principal component (FPC) and the random variable ξ_j is called its respective score.

As each deterministic function ϕ_j is normalised, the variance of X in the principal direction ϕ_j is simply $\mathrm{Var}\big[\xi_j\big] = \lambda_j$. In other words, λ_j quantifies how much variation of X is explained by ϕ_j. We can also show that

$$\mathrm{E}||X^c(\boldsymbol{t})||^2 = \sum_{j=1}^{\infty} \lambda_j,$$

which means that the variance of X is equal to the sum of the variances of the projections of X onto ϕ_j's.

Since λ_j's are arranged in nonincreasing order, we can calculate the cumulative fraction of variance explained (CFVE) by the first J eigenfunctions via

$$\mathrm{CFVE}_J = \frac{\sum_{j=1}^{J} \lambda_j}{\sum_{j=1}^{\infty} \lambda_j}. \tag{2.6}$$

In practice, as we can only estimate a finite number of FPCs, the denominator of (2.6) is replaced by $\sum_{j=1}^{J^*} \lambda_j$, where J^* is large.

One strategy to reduce eigenequation (2.5) to a matrix form is by representing the observed data as a linear combination of fixed (known) J basis functions (Ramsay & Silverman, 2005, Section 8.4). This provides estimated eigenvalues $\hat{\lambda}_j$ and the corresponding FPCs $\hat{\phi}_j$, where the maximum number of FPCs is J, the dimension of the basis. Some drawbacks of this strategy are clear: the estimated FPCs are sensitive to the basis functions used to represent the data and to the sparsity level of the observed data. Another strategy is to estimate the covariance function nonparametrically and take its eigenfunctions as the FPCs (see e.g. *principal component analysis through conditional expectation* (PACE) in Yao *et al.* (2005)).

2.3.1 Multivariate FPCA

Ramsay & Silverman (2005) propose the bivariate FPCA, which can be straightforwardly extended to the multivariate functional principal component analysis (MFPCA). In this approach, the functions $X_l(t),\ l = 1, \ldots, M$, are concatenated into a unique long curve for each individual of the sample. Next, FPCA is performed on the concatenated curves. As in FPCA, one obtains a vector of scores for each individual. However, this method might not work well as we often encounter different degrees of variability in different functional variables, which means that each functional variable may require a different number of

components.

Chiou *et al.* (2014) propose an approach in order to cope with these problems: they consider that each of the M functions may have different variation and extend FPCA to the multivariate case by using cross-covariance functions estimated nonparametrically through a local linear plane, so that their approach take into account the dependence among the functions. Nevertheless, the nonparametric estimation may suffer from the *curse of dimensionality*.

Berrendero *et al.* (2011) suggest an alternative way to reduce the dimension of multivariate functional data. The main aim is to summarize the vector of functions for each individual employing a very small number of functions which retain most of the information from M original functions. This is done by looking for curves which are obtained after finding principal components based on the $M \times M$ covariance matrix of the M functional variables at each time t independently. As they show in their first simulation study, the curves that summarise the data tend to be quite rough when the random functions are weakly correlated, something which is difficult to explain. Therefore, time dependence should be considered or some regularisation should be applied in order to obtain smooth summary curves. In practice, this is important as we often encounter low correlated functions in settings with a large M.

Chiou & Müller (2014) propose a linear manifold model which identifies linear combinations of the components of multivariate functional data and is determined by varying-coefficient functions that describe time varying relationships between those components. However, as in Berrendero *et al.* (2011), they obtain curves that summarise the data, rather than scores as in the Ramsay & Silverman (2005)'s FPCA approach.

A recent approach called MFPCA (Happ & Greven, 2018) can be applied to a more general case, allowing to include functional variables irregularly sampled and also observed on different dimensional domains. In this approach, each observation consists of $M > 2$ functions $X_1, \ldots, X_M$, where each one may be defined on different domains, $\mathcal{T}_1, \ldots, \mathcal{T}_M$, with possible different dimensions. The article suggests an estimation strategy to calculate multivariate FPCs and scores based on their univariate counterparts.

In Chapter 5, we use a convolution-based approach where cross-covariance functions are explicitly modelled. This is achieved by assuming that multiple functional variables are constructed from the same source. Each functional variable, though, can also have independent features. This framework is especially important to guarantee positive definiteness of the covariance function of the multivariate response.

2.4 Scalar-on-function regression model

We usually classify functional regression models into three types: *scalar-on-function* regression (where there are a scalar response and functional covariates); *function-on-scalar* regression (functional response and scalar covariates); and *function-on-function* regression (functional response and functional covariates). In this section, we focus on the first type. Recent and comprehensive overviews of it can be found in Morris (2015) and Reiss *et al.* (2016).

In the scalar-on-function regression model, we aim to predict a scalar response Y given a functional covariate $X(t),\ t \in \mathcal{T} \subset \mathbb{R}$. Let $X_i^c(t) = X_i(t) - \mu(t),\ i = 1, \ldots, N$, be the i-th realisation of the centred functional variable and Y_i the corresponding scalar response variable. Then the model is given by

$$Y_i = a + \int_{\mathcal{T}} b(t) X_i^c(t) dt + \varepsilon_i, \qquad i = 1, \ldots, N, \tag{2.7}$$

where a is a scalar parameter, $b(t)$ is the regression coefficient function (or slope function), and ε is a scalar error term satisfying $\mathrm{E}\big[\varepsilon | X\big] = 0$.

If each individual curve x_i is observed at n time points $t_{i,1}, \ldots, t_{i,n}$, then a first idea to estimate (2.7) might be to use the discrete observed values of each functional observation $x_i(t)$ and to fit the model

$$y_i = a + \sum_{j=1}^{n} \gamma_j x_i^c(t_{ij}) + \varepsilon_i, \qquad i = 1, \ldots, N.$$

However, as n is typically large, this may lead to multicollinearity problems. We actually often have $n > N$, which makes it impossible to obtain estimates of the slope parameters γ_j by standard linear regression techniques.

One way to reduce (2.7) is by taking the expansions of X_i^c and b in terms of orthogonal basis functions $\phi_1, \phi_2, \ldots$:

$$X_i^c(t) = \sum_{j=1}^{\infty} \xi_{ij} \phi_j(t) \tag{2.8}$$

and

$$b(t) = \sum_{k=1}^{\infty} \nu_k \phi_k(t). \tag{2.9}$$

In particular, for the first J elements of the expansion (2.8),

$$\xi_{ij} = \int_{\mathcal{T}} X_i^c(t) \phi_j(t) dt, \qquad j = 1, 2, \ldots, J,$$

minimises $||X_i{}^c(t) - \sum_{j=1}^{J} \xi_{ij}\phi_j(t)||$ (Weidmann, 1980, p.38). The term ξ_{ij} is called the j-th score for the i-th functional observation. Similarly, for expansion (2.9),

$$\nu_k = \int_{\mathcal{T}} b(t)\phi_k(t)dt, \qquad k = 1, 2, \ldots, J.$$

Hence, our model (2.7) becomes

$$\begin{aligned} Y_i &= a + \int_{\mathcal{T}} \sum_{j,k} \nu_k \phi_k(t)\xi_{ij}\phi_j(t)dt + \varepsilon_i \\ &= a + \sum_{j,k} \nu_k \xi_{ij} \int_{\mathcal{T}} \phi_k(t)\phi_j(t)dt + \varepsilon_i \\ &= a + \sum_{j=1}^{J} \nu_j \xi_{ij} + \varepsilon_i, \end{aligned}$$

since $\phi_1, \ldots, \phi_J$ are orthogonal functions. Therefore, the scalar response Y_i can be written simply as a linear combination of the scores $\xi_{i1}, \ldots, \xi_{iJ}$. Once estimated those first J scores, we can easily see that $\nu_1, \ldots, \nu_J$ can be obtained by regressing Y_i on the scores.

We could use basis ϕ_j independently of the data (e.g. B-splines or Fourier basis systems). However, we cannot guarantee that the first terms of those basis functions will explain most of the variation in X and $b(t)$. For this reason, one can instead use the information available in the data to construct the basis functions.

A common alternative is to use FPCs ϕ_j, that is, the eigenfunctions of the covariance operator of X obtained by eq. (2.5). This leads to what we call functional principal component regression (FPCR).

As FPCs only take into account the variation of the functional predictor X, we cannot guarantee that the first few terms will provide good predictions for a scalar response Y. Therefore, functional partial least squares (FPLS) basis might be an appropriate alternative as it considers the covariation between X and Y when constructing the basis functions. The functional regression model that uses FPLS basis is called the functional partial least squares regression (FPLSR). Theory about FPLSR is found in Delaigle & Hall (2012), which we will discuss in Chapter 6.

In Chapter 6, we further propose an extension of model (2.7) to the case involving $M > 1$ functional covariates, given by

$$Y_i = a + \sum_{l=1}^{M} \int_{\mathcal{T}} b_l(t) X_{l,i}^c(t)dt + \varepsilon_i, \qquad i = 1, \ldots, N.$$

Analogously to the case involving one functional covariate, this regression equation can be

solved by expanding the M-variate $\boldsymbol{X}(\cdot) = \big(X_1(\cdot), \ldots, X_M(\cdot)\big)^\top$ in terms of multivariate FPC basis or multivariate FPLS basis, which lead, respectively, to the multivariate functional principal component regression (MFPCR) and to the multiple functional partial least squares regression (MFPLSR).

Chapter 3

Gaussian process regression model

In this chapter, we discuss a nonparametric Bayesian framework for modelling function-valued processes by using a Gaussian process prior. We introduce the Gaussian process regression model in Section 3.1, where we describe why they have become popular and cite some recent developments. Inference procedures on the GPR model are shown in Section 3.2. Section 3.3 discusses commonly used families of parametric covariance functions, their properties and their interpretable parameters. Asymptotic theory is provided in Section 3.4. Approximate implementation methods to reduce computational costs and implementation issues are discussed in Section 3.5. Section 3.6 presents a simulation study for analysing the decomposition of a GP and for comparing approximate implementation methods.

3.1 Introduction to Gaussian process regression

Let us consider the following nonlinear functional regression model or process regression model:

$$X(\boldsymbol{t}) = f(\boldsymbol{t}) + \varepsilon(\boldsymbol{t}), \quad \varepsilon(\boldsymbol{t}) \sim N(0, \sigma_\varepsilon^2), \tag{3.1}$$

where $\boldsymbol{t} \in \mathcal{T} \subset \mathbb{R}^Q$ and the unknown nonlinear regression function f is a mapping $f : \mathbb{R}^Q \to \mathbb{R}$. We assume that the additive noise $\varepsilon(\boldsymbol{t})$ has normal distribution, but we could assume it has a different distribution (e.g. generalised Gaussian process regression models in Wang & Shi (2014)).

A variety of models has been proposed to estimate the unknown function f. Popular models are based on the approximation $f(\boldsymbol{t}) = \sum_{j=1}^{J} \alpha_j \phi_j(\boldsymbol{t})$, where ϕ_j are, for example, smoothing splines (Wahba, 1990). One of the major difficulties of these nonparametric approaches is the curse of dimensionality problem in the estimation process when $\boldsymbol{t}$ is multidimensional.

From the Bayesian perspective, the function f is treated as an unknown process (an unknown random function defined in a functional space analogue of a random unknown parameter defined in a conventional Bayesian approach). Therefore, we need to specify a prior distribution over the (random) function f to make probabilistic inference about f. One way to do this is by using a Gaussian process (GP) prior.

The Gaussian process (O'Hagan, 1978; Rasmussen & Williams, 2006; Shi & Choi, 2011) is defined as a stochastic process parametrised by its mean function

$$\mu(\cdot) : \mathcal{T} \to \mathbb{R},\ \mu(\boldsymbol{t}) = \mathrm{E}\big[f(\boldsymbol{t})\big],$$

and its covariance function

$$k(\cdot,\cdot) : \mathcal{T}^2 \to \mathbb{R},\ k(\boldsymbol{t},\boldsymbol{t}') = \mathrm{Cov}\big[f(\boldsymbol{t}), f(\boldsymbol{t}')\big].$$

From now on, we will write the GP as

$$f(\cdot) \sim GP\big(\mu(\cdot), k(\cdot,\cdot)\big). \tag{3.2}$$

GP can be seen as a generalisation of the multivariate Gaussian distribution to the infinite-dimensional setting. When we use a GP prior (3.2) for the random function f, (3.1) is referred to as Gaussian process regression (GPR) model. In this case, for any finite n and $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n \in \mathcal{T}$, the joint distribution of $\boldsymbol{x} = \big(x(\boldsymbol{t}_1), \ldots, x(\boldsymbol{t}_n)\big)^\top$ is an n-variate Gaussian distribution with mean vector $\boldsymbol{\mu}_n = \big(\mu(\boldsymbol{t}_1), \ldots, \mu(\boldsymbol{t}_n)\big)^\top$ and covariance matrix $\boldsymbol{\Psi}_n$ whose (i,j)-th entry is given by $\big[\boldsymbol{\Psi}_n\big]_{ij} = k(\boldsymbol{t}_i, \boldsymbol{t}_j) + \delta_{ij}\sigma_\varepsilon^2,\ i,j = 1, \ldots, n$ where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

When it is difficult to specify a fixed mean function $\mu(\cdot)$ in (3.2), we may use a mean function $\mu(\boldsymbol{t}) = \boldsymbol{h}(\boldsymbol{t})^\top\boldsymbol{\beta}$, where $\boldsymbol{h}(\boldsymbol{t})$ contains a set of function-valued covariates and $\boldsymbol{\beta}$ is inferred from the data (Rasmussen & Williams, 2006, Section 2.7); or a mean function $\mu(\boldsymbol{t}) = \boldsymbol{u}^\top\boldsymbol{\beta}(\boldsymbol{t})$, involving a set of scalar covariates and a varying coefficient (Shi *et al.*, 2007). However, as we will focus on the covariance structure, we will use the same mean function estimated via local linear smoother (Yao *et al.*, 2005) as it is commonly made in FDA. Other mean models can also be used.

GPR models have become popular for a number of reasons. Firstly, a wide class of nonlinear functions f can be modelled by choosing a suitable prior specification for $k(\cdot,\cdot)$. Other prior distributions can be used for robust heavy-tailed processes (Shah *et al.*, 2014; Wang *et al.*, 2017; Cao *et al.*, 2018). This enables us to estimate the covariance structure directly based on the data. In addition, the applicability of GPR models can be readily

extended to random processes defined on dimensions higher than two. Finally, these models allow to easily quantify the variability of predictions.

Many recent developments have been made in GPR analysis, including variational GP (Tran *et al.*, 2015), distributed GP (Deisenroth & Ng, 2015), manifold GP (Calandra *et al.*, 2016), linearly constrained GP (Jidling *et al.*, 2017), convolutional GP (van der Wilk *et al.*, 2017), and deep GP (Dunlop *et al.*, 2018). Some studies investigate connections between GPs with frequentist kernel methods based on reproducing kernel Hilbert spaces (Kanagawa *et al.*, 2018). Finally, many extensions and adaptations have been suggested to apply GPR models to different types of data, such as big data (Liu *et al.*, 2018), binary times series (Sung *et al.*, 2017), large spatial data (Zhang *et al.*, 2019), and mixed functional and scalar data in nonparametric functional regression (Wang & Xu, 2019).

3.2 Inference

For a given set of observed data $\mathcal{D} = \{\boldsymbol{x}, \boldsymbol{t}\} = \{(x_i, t_{i1}, \ldots, t_{iQ}),\ i = 1, \ldots, n\}$, a GPR model for (3.1) can be written as

$$\begin{aligned} x_i | f_i &\overset{\text{i.i.d.}}{\sim} N(f_i, \sigma_\varepsilon^2), \\ (f_1, \ldots, f_n) &\sim GP\big(\mathbf{0}, k(\cdot, \cdot; \boldsymbol{\theta})\big), \end{aligned} \tag{3.3}$$

where $k(\cdot, \cdot; \boldsymbol{\theta})$ contains the hyperparameter $\boldsymbol{\theta}$. Thus, the marginal distribution of $\boldsymbol{x}$ given $\boldsymbol{\theta}$ is

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \int p(\boldsymbol{x}|\boldsymbol{f}) p(\boldsymbol{f}|\boldsymbol{\theta}) d\boldsymbol{f},$$

where $p(\boldsymbol{x}|\boldsymbol{f}) = \prod_{i=1}^{n} \zeta(f_i)$, with $\zeta(f_i)$ denoting the normal probability density function with mean f_i and variance σ_ε^2, and $\boldsymbol{f} = \big(f(t_1), \ldots, f(t_n)\big)^\top \sim N(\mathbf{0}, \boldsymbol{K}_n)$, where $\big[\boldsymbol{K}_n\big]_{ij} = k(\boldsymbol{t}_i, \boldsymbol{t}_j),\ i, j = 1, \ldots, n$.

Given that the marginal distribution of $\boldsymbol{x}$ is $N(\mathbf{0}, \boldsymbol{\Psi}_n)$, where $\boldsymbol{\Psi}_n = \boldsymbol{K}_n + \sigma_\varepsilon^2 \boldsymbol{I}_n$, the marginal log-likelihood of $\boldsymbol{\theta}$ is given by

$$\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = -\frac{1}{2} \log |\boldsymbol{\Psi}_n(\boldsymbol{\theta})| - \frac{1}{2} \boldsymbol{x}^\top \boldsymbol{\Psi}_n(\boldsymbol{\theta})^{-1} \boldsymbol{x} - \frac{n}{2} \log 2\pi. \tag{3.4}$$

The estimates of $\boldsymbol{\theta}$ obtained by maximising (3.4) are called empirical Bayes estimates as they are obtained by using observed data (Carlin & Louis, 2008).

As we will see in the next section, the values of the hyperparameters in $\boldsymbol{\theta}$ control several properties of the covariance function and, consequently, determine the behaviour of the regression curve. As discussed in (Shi & Choi, 2011, Chapter 3), unless we have

a very good prior knowledge of the values of the hyperparameters, the selection of each hyperparameter should be done carefully.

We can define a hyperprior distribution for $\boldsymbol{\theta}$. In this case, our knowledge about $\boldsymbol{\theta}$ is updated as more data are observed. In fact, finding the mode of the posterior density is a way to find what we call the *maximum a posteriori* (MAP) estimate of $\boldsymbol{\theta}$. When we use a non-informative or a uniform prior distribution, the MAP estimates are precisely the same as the empirical Bayes estimates (Shi & Choi, 2011).

Instead of assuming a probability structure for the hyperparameters $\boldsymbol{\theta}$, in this thesis we always use empirical Bayes estimates using the data to estimate them.

Predictive distribution

Let us consider the GPR model (3.3). The marginal distribution of $\boldsymbol{x}$ is $N(\mathbf{0}, \boldsymbol{\Psi}_n)$, where $\boldsymbol{\Psi}_n = \boldsymbol{K}_n + \sigma_\varepsilon^2 \boldsymbol{I}_n$. Therefore, we can easily make predictions of test data at locations $\boldsymbol{t}$ given the observed data $\mathcal{D}$. The posterior distribution $p(f(\boldsymbol{t})|\mathcal{D})$, for any arbitrary $\boldsymbol{t}$, also has multivariate normal distribution with

$$\mathrm{E}\left[f(\boldsymbol{t})|\mathcal{D}\right] = \boldsymbol{k}_n^\top(\boldsymbol{t})(\boldsymbol{K}_n + \sigma_\varepsilon^2 \boldsymbol{I}_n)^{-1}\boldsymbol{x}, \tag{3.5}$$

$$\mathrm{Var}\left[f(\boldsymbol{t})|\mathcal{D}\right] = k(\boldsymbol{t},\boldsymbol{t}) - \boldsymbol{k}_n^\top(\boldsymbol{t})(\boldsymbol{K}_n + \sigma_\varepsilon^2 \boldsymbol{I}_n)^{-1}\boldsymbol{k}_n(\boldsymbol{t}), \tag{3.6}$$

where $\boldsymbol{x} = \big(x(\boldsymbol{t}_1), \ldots, x(\boldsymbol{t}_n)\big)^\top$, and $\boldsymbol{k}_n(\boldsymbol{t}) = \big(k(\boldsymbol{t}_1, \boldsymbol{t}), \ldots, k(\boldsymbol{t}_n, \boldsymbol{t})\big)^\top$.

However, the predictive distribution becomes much more complicated for non-Gaussian data (see e.g. Wang & Shi (2014)). Therefore, we may consider using the decomposition methods detailed below.

Once the covariance function $k(\cdot,\cdot)$ is estimated, we can estimate the corresponding eigenfunctions $\phi(\cdot)$ via the Nyström method for approximating eigenfunctions. Then a finite GPR approximation can be obtained as in (2.4) by using only the first J eigenfunctions. This allows us to make predictions at any arbitrary location $\boldsymbol{t}$ given observed data $\boldsymbol{x}$ and a finite number of components $\phi_j(\cdot),\ j = 1, \ldots, J$, similarly as in FPCA.

3.3 Covariance functions

The specification of the covariance function in (3.2) is important because it fixes the properties of the underlying function f that we want to infer. In this section, we discuss properties and popular families of covariance functions and make interpretation on their parameters. The selection of covariance functions is widely discussed in Rasmussen & Williams (2006) and Shi & Choi (2011). The validity and further classes of covariance

functions can be seen in many references (see, for example, Abrahamsen (1997); Shi & Choi (2011)).

Properties

In this subsection, we again use the notation $k(\boldsymbol{t}, \boldsymbol{t}') = \mathrm{Cov}\big[X(\boldsymbol{t}), X(\boldsymbol{t}')\big]$. In addition, let $\boldsymbol{h} = \boldsymbol{t} - \boldsymbol{t}'$ (or the unidimensional counterpart $h = t - t'$) be the separation vector (value).

A zero mean stochastic process X is *strongly stationary* if the distribution of $\big(X(\boldsymbol{t}_1), \ldots, X(\boldsymbol{t}_n)\big)^\top$ is the same as that of $\big(X(\boldsymbol{t}_1 + \boldsymbol{h}), \ldots, X(\boldsymbol{t}_n + \boldsymbol{h})\big)^\top$. It is *weakly stationary* if $k(\boldsymbol{t}, \boldsymbol{t} + \boldsymbol{h})$ only depends on the separation vector $\boldsymbol{h}$. Strong stationarity implies weak stationarity. The converse in general is not true, but if X is a GP, then it will be. Therefore, for GPs the weak stationarity and the strong stationarity are equivalent.

A covariance function is said to be *isotropic* if $k(\boldsymbol{t}, \boldsymbol{t} + \boldsymbol{h})$ only depends on the distance $||\boldsymbol{h}||$. When it also depends on the direction of $\boldsymbol{h}$, then we have a stationary process with a *anisotropic* covariance function. For example, a *separable* covariance function $k(\boldsymbol{t}, \boldsymbol{t} + \boldsymbol{h}) = \sigma^2 g_1(h_1) g_2(h_2)$, where g_1 and g_2 are valid correlation functions, is stationary, but it can be anisotropic as it depends on the direction $\boldsymbol{h} = (h_1, h_2)^\top$.

For the two-dimensional setting $\boldsymbol{t} = (s, \tau)^\top \in \mathcal{T} \subset \mathbb{R}^2$, if the covariance function cannot be factorised as $k(s, \tau, s', \tau') = k_1(s, s') k_2(\tau, \tau')$, then it is called *nonseparable*. The geometric anisotropic covariance function (Banerjee *et al.*, 2015) given by

$$k(\boldsymbol{t}, \boldsymbol{t}') = \sigma^2 g\big((\boldsymbol{t} - \boldsymbol{t}')^\top \boldsymbol{B} (\boldsymbol{t} - \boldsymbol{t}')\big), \tag{3.7}$$

where g is a valid correlation function, allows the covariance to be nonseparable. If the off-diagonal elements in $\boldsymbol{B}$ are nonzero, the covariance function (3.7) is nonseparable. The same idea of separability is extended to the cases of higher dimensions.

Varying the hyperparameters

Let us first discuss how different values for hyperparameters of a covariance function can be used to model a rich variety of curves.

We illustrate this by taking as an example the commonly used squared exponential covariance function, defined as

$$\mathrm{Cov}\big[X(\boldsymbol{t}_i), X(\boldsymbol{t}_j)\big] = \sigma^2 \exp\Big\{ -\frac{1}{\gamma} ||\boldsymbol{t}_i - \boldsymbol{t}_j||^2 \Big\} + \sigma_\varepsilon^2 \delta_{ij}, \quad \gamma > 0. \tag{3.8}$$

The hyperparameters σ^2, σ_ε^2 and γ are called the signal variance, the noise variance and the length-scale, respectively. In spatial statistics, these hyperparameters are called

the *partial sill*, the *nugget effect* and the *range parameter*, respectively. We will also use $\omega = 1/\gamma$ (the *decay parameter*) to simplify notation.

By considering the noise variance σ_ε^2, the covariance function (3.8) is used for the noisy curves $X(\boldsymbol{t})$ rather than for the underlying function f (see eq. (3.1)).

Whereas the value of σ^2 controls the vertical scale of variation of f, the values of σ_ε^2 and γ determine the smoothness of the sample paths. When we increase σ_ε^2 or decrease γ, in both cases we have rougher curves. However, the estimated values of these two hyperparameters can indicate whether the roughness degree comes from the underlying function f (signal) or from the noise.

From now on, to simplify the exposition, we will show examples of covariance functions without the measurement error term.

Powered exponential

The squared exponential covariance function (3.8) is a particular case of the powered exponential class of covariance functions when $\gamma = 2$. This class has the form

$$k(\boldsymbol{t}, \boldsymbol{t}') = \nu \exp\Big\{ -\omega ||\boldsymbol{t} - \boldsymbol{t}'||^\gamma \Big\}, \quad \nu > 0, \quad \omega \geq 0, \quad 0 < \gamma \leq 2. \tag{3.9}$$

and allows to model rougher curves than the squared exponential covariance function does.

Rational quadratic

The rational quadratic class can be seen as a *scale mixture* of squared exponential covariance functions with different length scale parameters (Rasmussen & Williams, 2006) and is given by

$$k(\boldsymbol{t}, \boldsymbol{t}') = \Big(1 + s_\alpha \omega ||\boldsymbol{t} - \boldsymbol{t}'||^2\Big)^{-\alpha}, \quad \alpha, \omega \geq 0. \tag{3.10}$$

We can see that the squared exponential is a particular case of the rational quadratic class as $\alpha \to \infty$. When $s_\alpha = 20^{1/\alpha} - 1$, (3.10) is called *Cauchy* covariance kernel (Shi & Choi, 2011).

Matérn

The Matérn class of covariance functions is given by

$$k(\boldsymbol{t}, \boldsymbol{t}') = \frac{1}{\Gamma(\nu) 2^{\nu-1}} \Big(\sqrt{2\nu}\omega ||\boldsymbol{t} - \boldsymbol{t}'||\Big)^\nu \mathcal{K}_\nu \Big(\sqrt{2\nu}\omega ||\boldsymbol{t} - \boldsymbol{t}'||\Big), \quad \omega \geq 0, \tag{3.11}$$

where $\mathcal{K}_\nu$ is the modified Bessel function of order ν.

This class is very general and can accommodate several particular cases. For example, as $\nu \to \infty$, we obtain a squared exponential covariance function. In machine learning, we often encounter applications with using $\nu = 3/2$ and $\nu = 5/2$. This is due to the fact that if $\nu = p + 1/2$, where p is a non-negative integer, the resulting covariance function is a product of a polynomial of order p and an exponential (Rasmussen & Williams, 2006). If $\nu = 1/2$, we obtain an equivalent expression to the exponential covariance function.

Relaxing the anisotropy assumption

We can easily extend the last three families by including a different decay parameter ω_q for each coordinate direction. This is done by replacing the term $\omega||\boldsymbol{t} - \boldsymbol{t}'||^\gamma$ by $\sum_{q=1}^{Q} \omega_q||\boldsymbol{t}_q - \boldsymbol{t}'_q||^\gamma$ in eq. (3.9), and similarly in (3.10) and (3.11). This brings more flexibility as it measures how quickly the surface varies along each coordinate direction. In addition, as discussed in Shi & Choi (2011), a very small value of ω_q can also indicate that the input $\boldsymbol{t}_q$ can be excluded from the model.

Moreover, we can use an even more general norm of the form $(\boldsymbol{t} - \boldsymbol{t}')^\top \boldsymbol{B}(\boldsymbol{t} - \boldsymbol{t}')$ (as seen in (3.7)) to extend the covariance function to the anisotropic case, provided that $\boldsymbol{B}$ is positive definite. Ecker & Gelfand (1999) and Banerjee *et al.* (2015) explain how to model and conduct inference incorporating this extension.

The diagonal elements of $\boldsymbol{B}$, b_q, usually called decay parameters, control how quickly the function f varies on each coordinate direction. The larger the value, the quicker is the variation of f towards the related direction. The off-diagonal elements of $\boldsymbol{B}$, $b_{pq}, p \neq q$, may be non-zero. If they are, we say that there exists interaction between the coordinate directions $\boldsymbol{t}_p$ and $\boldsymbol{t}_q$ and covariance functions of the form (3.7) become nonseparable.

3.4 Asymptotic theory

In this section, we provide asymptotic theory for the decomposition and Bayesian prediction based on a Gaussian process prior with a general covariance structure.

In eq. (2.4), ξ_j are independent random variables and $\phi_j(\cdot)$ are the eigenfunctions of the kernel function $k(\cdot, \cdot)$. Therefore, the eigenfunctions are orthonormal satisfying

$$\int k(\boldsymbol{t}, \boldsymbol{t}')\phi_j(\boldsymbol{t}')d\boldsymbol{t}' = \lambda_j\phi_j(\boldsymbol{t}), \quad \int \phi_i(\boldsymbol{t})\phi_j(\boldsymbol{t})d\boldsymbol{t} = \delta_{ij}, \tag{3.12}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ are the eigenvalues of $k(\cdot, \cdot)$ and δ_{ij} is the Kronecker delta.

Let $X^c(\boldsymbol{t}) = X(\boldsymbol{t}) - \mu(\boldsymbol{t})$ and $\Xi(f) = \int_{\mathcal{T}} k(\boldsymbol{t}, \cdot)f(\boldsymbol{t})d\boldsymbol{t}$ be an operator for $f \in L^2(\mathcal{T})$.

In fact,

$$\xi_j = \langle X^c(\cdot), \phi_j(\cdot)\rangle = \int X^c(\boldsymbol{t})\phi_j(\boldsymbol{t})d\boldsymbol{t}$$

has mean 0 and variance λ_j.

Theorem 1. *For $J \geq 1$, for which $\lambda_J > 0$, the functions $\{\phi_j,\ j = 1, \ldots, J\}$ provide the best finite dimensional approximation to $X^c(\boldsymbol{t})$ with respect to minimising criterion*

$$\underset{g_1,\ldots,g_J \in L^2(\mathcal{T})}{\operatorname{argmin}} \mathrm{E}\Big[||X^c(\boldsymbol{t}) - \sum_{j=1}^{J} g_j(\boldsymbol{t})\xi_j^*||^2\Big], \tag{3.13}$$

where $g_1, \ldots, g_J \in L^2(\mathcal{T})$ are orthonormal and $\xi_j^* = \langle X^c(\cdot), g_j(\cdot)\rangle = \int X^c(\boldsymbol{t})g_j(\boldsymbol{t})d\boldsymbol{t}$. The minimised value is $\sum_{j=J+1}^{\infty}\lambda_j$.

The proof is given in Section 3.9. This theorem is similar to Theorem 1 in Chen *et al.* (2017); but the latter provides the best finite approximation under separability assumption. The above theorem is true for a very general covariance structure even if it is nonstationary or nonseparable (see detailed discussion in 4).

We are also interested in the convergence rates of $k_{\hat{\theta}}(\cdot,\cdot)$, where $\hat{\theta}$ is the parameter vector that maximises the marginal log-likelihood (3.4). The following theorem provides the convergence rates of $k_{\hat{\theta}}(\cdot,\cdot)$ (and the related terms of its decomposition) also under a general covariance structure.

Theorem 2. *Suppose conditions C1 - C3 in Section 3.9 hold, and $\hat{\mu}(\boldsymbol{t})$ satisfies $\sup_{\boldsymbol{t}} |\hat{\mu}(\boldsymbol{t}) - \mu(\boldsymbol{t})| = O_p\big(\{\log(n)/n\}^{1/2}\big)$ (see e.g. Chen & Müller (2012)), we have, for $1 \leq j \leq J$,*

$$\begin{aligned}
&||k_{\hat{\theta}}(\cdot,\cdot) - k_\theta(\cdot,\cdot)|| = O_p(\{\log(n)/n\}^{1/2}),\\
&||\hat{\lambda}_j - \lambda_j|| = O_p(\{\log(n)/n\}^{1/2}),\\
&||\hat{\phi}_j(\cdot) - \phi_j(\cdot)|| = O_p(\{\log(n)/n\}^{1/2}),\\
&||\hat{\xi}_j - \xi_j|| = O_p(\{\log(n)/n\}^{1/2}).
\end{aligned}$$

The proof is given in Section 3.9.

We now look at the relationship between the Bayesian prediction and the decomposition based on Karhunen-Loéve expansion. Using models (3.1) and (3.2), where $f \sim GP(0, k)$ with $k = k_\theta$ and $\varepsilon(\boldsymbol{t})$ is a Gaussian error process $GP(0, k_\varepsilon)$ with $k_\varepsilon(\boldsymbol{t}, \boldsymbol{t}') = \sigma_\varepsilon^2 I(\boldsymbol{t} = \boldsymbol{t}')$. Hence, $X \sim GP(0, \tilde{k}_\theta)$ with $\tilde{k}_\theta = k_\theta + k_\varepsilon$. Given $\boldsymbol{f} = \big(f(\boldsymbol{t}_1), \ldots, f(\boldsymbol{t}_n)\big)^\top$, we use $\mathrm{E}\left[f(\boldsymbol{t})|\boldsymbol{f}\right] = \boldsymbol{k}_n^\top(\boldsymbol{t})\boldsymbol{K}_n^{-1}\boldsymbol{f}$ to estimate $f(\boldsymbol{t})$. Given the observed data $\mathcal{D}$, we use (3.5) to estimate $f(\boldsymbol{t})$.

In addition, from Karhunen-Loéve expansion we have

$$f(\boldsymbol{t}) = \sum_{j=1}^{\infty} \phi_j(\boldsymbol{t})\xi_j, \qquad X(\boldsymbol{t}) = \sum_{j=1}^{\infty} \tilde{\phi}_j(\boldsymbol{t})\tilde{\xi}_j, \tag{3.14}$$

where $\phi_j(\cdot)$ and $\tilde{\phi}_j(\cdot)$ are the eigenfunctions of k_θ and $\tilde{k}_\theta$, respectively, and their corresponding eigenvalues are $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ and $\tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq 0$, respectively. The truncated sum of (3.14) will be

$$f_n(\boldsymbol{t}) = \sum_{j=1}^{n} \phi_j(\boldsymbol{t})\xi_j, \qquad X_n(\boldsymbol{t}) = \sum_{j=1}^{n} \tilde{\phi}_j(\boldsymbol{t})\tilde{\xi}_j. \tag{3.15}$$

Nyström method

We now briefly describe the Nyström method for approximating eigenfunctions, which is used in the proof of Theorem 3. The Nyström method is presented by Williams & Seeger (2001) to speed up kernel-based machines and further discussed by Williams *et al.* (2002). Using a random sample $\{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n\}$, the first integral of (3.12) can be approximated by

$$\lambda_j \phi_j(\boldsymbol{t}') = \int k(\boldsymbol{t}, \boldsymbol{t}')\phi_j(\boldsymbol{t})d\boldsymbol{t} \approx \frac{1}{n}\sum_{h=1}^{n} k(\boldsymbol{t}_h, \boldsymbol{t}')\phi_j(\boldsymbol{t}_h). \tag{3.16}$$

If we plug in $\boldsymbol{t}' = \boldsymbol{t}_h$ for $h = 1, \ldots, n$, into (3.16), we have a matrix eigenproblem

$$\boldsymbol{K}_n \boldsymbol{V}_n = \boldsymbol{\Lambda}_n \boldsymbol{V}_n,$$

where $\boldsymbol{K}_n = (k(\boldsymbol{t}_i, \boldsymbol{t}_j))_{n\times n}$, $\boldsymbol{\Lambda}_n = \operatorname{diag}\big(\lambda_1^{(n)}, \ldots, \lambda_n^{(n)}\big)$, with $\lambda_1^{(n)} \geq \cdots \geq \lambda_n^{(n)} \geq 0$, where $\lambda_j^{(n)}$ is the j-th eigenvalue of $\boldsymbol{K}_n$ and $\boldsymbol{V}_{j,n}$ (the j-th column of $\boldsymbol{V}_n$) the corresponding normalised eigenvector. Therefore, the eigenfunctions $\phi_j,\ j = 1, \ldots, n$, are approximated by

$$\phi_j(\boldsymbol{t}_h) \approx \sqrt{n} V_{hj,n}, \quad \text{and} \quad \lambda_j \approx \frac{\lambda_j^{(n)}}{n}, \tag{3.17}$$

where $V_{hj,n}$ is the h-th element of $\boldsymbol{V}_{j,n}$.

The Nyström approximation for the eigenfunctions extends the first equation of (3.17) from the locations $\{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n\}$ to any arbitrary location $\boldsymbol{t}$ by

$$\phi_j(\boldsymbol{t}) \approx \frac{\sqrt{n}}{\lambda_j^{(n)}} \boldsymbol{k}_n^\top(\boldsymbol{t}) \boldsymbol{V}_{j,n},$$

where $\boldsymbol{k}_n(\boldsymbol{t}) = \big(k(\boldsymbol{t}_1, \boldsymbol{t}), \ldots, k(\boldsymbol{t}_n, \boldsymbol{t})\big)^\top$.

Theorem 3 *Under conditions in Theorem 2, we have* $\mathrm{E}[f(\boldsymbol{t})|\boldsymbol{f}] = f_n(\boldsymbol{t}) + o_p(1)$. *Moreover, under model* (3.1), $\mathrm{E}[f(\boldsymbol{t})|\mathcal{D}] = X_n(\boldsymbol{t}) + o_p(1)$.

The proof is given in Section 3.9. This theorem indicates that the Bayesian prediction and Karhunen-Loéve expansion provide similar results. In the proof, we could have kept the convergence rate of $O_p\big(1/\sqrt{n}\big)$ instead of $o_p(1)$. However, the former convergence rate is faster than the convergence rates of $O_p\big(\log(n)/\sqrt{n}\big)$ shown in Theorem 2, and this might cause some misunderstanding. This happens because in Theorem 3 we show the convergence rates of the decompositions for a given covariance function, and therefore convergence rates related to parameter estimation are not considered.

This theorem provides flexibility in functional data analysis. If we are mainly interested in a predictive model for Gaussian data, we may just use the Bayesian prediction. The implementation is fairly efficient if the sample size is not very large. However, if we are also interested in how the covariance is structured, we may study the eigenfunctions and the eigenvalues. This also provides a way to develop efficient approximation for big data (e.g. Nyström method (Shi & Choi, 2011, p.42)) or for non-Gaussian data.

3.5 Approximate implementation methods

The evaluation of (3.4) requires computational time $\mathcal{O}(n^3)$ in order to invert $\boldsymbol{\Psi}_n$. As the number of sampling points n increases, the computational time becomes prohibitively high and we may need some strategy to speed up computation. The log-determinant $\log|\boldsymbol{\Psi}_n|$ can easily be obtained as a by-product of the inverse and therefore is not a main concern. We describe different approximate implementation methods in the next three subsections and give a few remarks on implementation issues in the fourth subsection.

3.5.1 Nyström approximation for covariance matrices

When the sample size n is large, the Nyström method can be used to obtain an approximation to $\boldsymbol{K}_n$. We first select a subset (of size $m < n$) of the rows/columns of $\boldsymbol{K}_n$. This subset is chosen to approximate the eigenfunctions at all n points. Then it is easy to obtain the Nyström approximation for the Gram matrix as

$$\boldsymbol{K}_n \approx \boldsymbol{K}_{nm}\boldsymbol{K}_m^{-1}\boldsymbol{K}_{mn},$$

where $\boldsymbol{K}_{nm}$ is the $n \times m$ block of the matrix $\boldsymbol{K}_n$ and $\boldsymbol{K}_m := \boldsymbol{K}_{mm}$ for notation convenience.

Consequently, we obtain an approximation to $\boldsymbol{\Psi}_n^{-1}$ in (3.4):

$$\begin{aligned}\boldsymbol{\Psi}_n^{-1} &= (\boldsymbol{K}_n + \sigma_\varepsilon^2 \boldsymbol{I}_n)^{-1} \\ &\approx \sigma_\varepsilon^{-2}\big[\boldsymbol{I}_n - \boldsymbol{K}_{nm}\big(\sigma_\varepsilon^2 \boldsymbol{K}_m + \boldsymbol{K}_{mn}\boldsymbol{K}_{nm}\big)^{-1}\boldsymbol{K}_{mn}\big],\end{aligned} \tag{3.18}$$

which involves a computational time of $\mathcal{O}(m^2 n)$ in the log-likelihood function evaluation. Williams *et al.* (2002) point out that the Nyström approximation for $\boldsymbol{K}_n$ work well if its $(m+1)$-th eigenvalue is significantly smaller than σ_ε^2.

Plugging in (3.18) into (3.5) and (3.6), we obtain the mean and variance predictions of the Nyström for approximating GPR.

3.5.2 Subset of Regressors

Silverman (1985) show that the posterior mean of the GP predictor can be obtained from a regression model $X(\boldsymbol{t}) = \sum_{i=1}^n c_i k(\boldsymbol{t}, \boldsymbol{t}_i)$ with a prior $\boldsymbol{c} \sim N(\boldsymbol{0}, \boldsymbol{K}_n^{-1})$. If this sum is truncated at $m < n$, setting all the remaining coefficients to zero, this results in a GPR model with covariance function $k_{\mathrm{SR}}(\boldsymbol{t}, \boldsymbol{t}') = \boldsymbol{k}_m^\top(\boldsymbol{t}) \boldsymbol{K}_m^{-1} \boldsymbol{k}_m(\boldsymbol{t}')$, where $\boldsymbol{k}_m(\boldsymbol{t}) = \big(k(\boldsymbol{t}, \boldsymbol{t}_1), \ldots, k(\boldsymbol{t}, \boldsymbol{t}_m)\big)^\top$. The resulting predictive mean and variance are, respectively,

$$\begin{aligned}\mathrm{E}\left[f_{SR}(\boldsymbol{t})|\mathcal{D}\right] &= \boldsymbol{k}_m^\top(\boldsymbol{t})\big(\sigma_\varepsilon^2 \boldsymbol{K}_m + \boldsymbol{K}_{mn}\boldsymbol{K}_{nm}\big)^{-1}\boldsymbol{K}_{mn}\boldsymbol{x}, \\ \mathrm{Var}\left[f_{SR}(\boldsymbol{t})|\mathcal{D}\right] &= \sigma_\varepsilon^2 \boldsymbol{k}_m^\top(\boldsymbol{t})\big(\sigma_\varepsilon^2 \boldsymbol{K}_m + \boldsymbol{K}_{mn}\boldsymbol{K}_{nm}\big)^{-1}\boldsymbol{k}_m(\boldsymbol{t}).\end{aligned}$$

This method was originally proposed by (Poggio & Girosi, 1990) in a regularisation framework and was later called the subset of regressors (SR) method.

SR is recommended over the Nyström method for approximating GPR because the latter might have bad performance for small n (Williams *et al.*, 2002) and its predictive variance can sometimes be negative.

3.5.3 Subset of Data

Another strategy to reduce computational time is by using the subset of data (SD) approximation method, which literally consists in selecting a subset of size $m < n$ of the training inputs, a subset called the *active set*, and discard the remaining observations. This strategy clearly leads to a computational complexity $\mathcal{O}(m^3)$.

Methods to select which observations belongs to the active set to minimise are usually based on a loss of information measure. For example, the *information gain criterion* (Seeger *et al.*, 2003), a greedy selection method based on a Kullback-Leibler divergence, and the *differential entropy score* (Lawrence *et al.*, 2003), based on the concept of entropy

of a Gaussian distribution. In our simulation studies, however, we have observed that these methods usually require considerable computational cost and do not provide significantly better results than if we simply randomly choose the subset of data points.

As we shall see in the simulation study in Section 3.6, for a fixed m, the SR approximation method usually provides better predictions than SD does. In our implementations, we have noticed that the predictions of SD method, however, can be improved if we use a subset of size m with observations closely located to the test set locations where we want to predict.

3.6 Simulation study

We assume the two-dimensional random process

$$\begin{aligned} X(\boldsymbol{t}) &= f(\boldsymbol{t}) + \varepsilon(\boldsymbol{t}), \quad \varepsilon(\boldsymbol{t}) \sim N(0, \sigma_\varepsilon^2), \\ f(\boldsymbol{t}) &\sim GP\big(\mathbf{0}, k(\boldsymbol{t}, \boldsymbol{t}')\big), \end{aligned} \tag{3.19}$$

where $k(\boldsymbol{t}, \boldsymbol{t}') = \nu \exp\left\{-0.5(\boldsymbol{t} - \boldsymbol{t}')^\top \boldsymbol{A}(\boldsymbol{t} - \boldsymbol{t}')\right\}$, with $\boldsymbol{A} = \operatorname{diag}(a_1, a_2)$. We set $\nu = 2$ and $\sigma_\varepsilon^2 = 0.1^2$. In addition, $a_1 = a_2 = 0.1$ are used to generate *smooth data* and $a_1 = a_2 = 1$ to generate *rough data*. We simulate $N = 100$ replicated surfaces on a two-dimensional grid of size $n = n_1 \times n_2 = 50 \times 40 = 2000$ and analyse different levels of sparsity by considering that the observed data are a subset of size $n_{\text{obs}} < n$. For what we call 'high sparsity', we only have $n_{\text{obs}} = 200$ observations. For the 'medium sparsity', $n_{\text{obs}} = 1000$, and for 'low sparsity', $n_{\text{obs}} = 1800$. The remaining $n_{\text{test}} = n - n_{\text{obs}}$ observations represent the test set which we use to assess predictions.

Analysing the decomposition of GPR model

We are first interested in analysing the decomposition of the GPR model (assuming that the parameters of the covariance function are known) in terms of CFVE and prediction results. To access predictions of test set values $\boldsymbol{x}^* = \big(x(t_1^*), \ldots, x(t_{n_{\text{test}}}^*)\big)^\top$ of each replicated curve, we use the standardised mean squared error (SMSE), defined as

$$\text{SMSE} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (\boldsymbol{x}_i^* - \hat{\boldsymbol{x}}_i^*)^2 / \text{Var}\left[\boldsymbol{x}^*\right].$$

By normalising the MSE by the variance of the test set values, the GPR predictions can be compared to the predictions made by the trivial sample mean estimator $\bar{\boldsymbol{x}}^*$, which will have SMSE close to 1.

The average SMSE prediction errors and the CFVE for smooth curves are shown in Table 3.1, and for rough curves in Table 3.2. For each level of sparsity, the predictions of Full GPR model can be obtained by either using (3.5) directly or using the truncated sum $\tilde{f}_{n_{\text{obs}}}(\boldsymbol{t}^*) = \sum_{j=1}^{n_{\text{obs}}} \tilde{\phi}_j(\boldsymbol{t}^*)\tilde{\xi}_j$ (see (3.15)), that is, using all the eigenfunctions available.

For a **fixed sample size** n_{obs}, we can clearly see that when the dataset contains rough curves, we need many more eigenfunctions to obtain small SMSE and high CFVE. Indeed, for rough curves, the sequence of eigenvalues has a much slower decay, which means that later eigenfunctions still has important contribution to explaining the variation in the data.

For a **fixed number of eigenfunctions**, J, SMSE becomes slightly smaller as we increase the sample size n_{obs} for both degrees of smoothness. However, as n_{obs} becomes larger, although CFVE is not significantly reduced in the case of smooth curves, it is reduced considerably when the curves are rough. We can also look at CFVE results in Figure 3.1 and SMSE results in Figure 3.2. For each fixed J, we vary n_{obs}. In this way, we look at connected points from the right to the left for a given J to see what happens as we increase n_{obs}. In the rough data scenario, for a fixed J, the increase of n_{obs} implies a strong decay of CFVE (see Figure 3.1). This effect does not seem to be significant for smooth data. Figure 3.2 shows that, for a fixed J, the increase of n_{obs} basically do not change the quality of the predictions.

In conclusion, the proportion J/n_{obs} has to be larger for rougher data to explain most of the variability in the data. On the other hand, for both smooth and rough data, as we increase n_{obs} for a fixed J, the predictions become only slightly better. In practice, for a given sample size n_{obs}, we usually choose J by looking at the CFVE for different values of J.

	High Sparsity		Medium Sparsity		Low Sparsity	
	SMSE	CFVE	SMSE	CFVE	SMSE	CFVE
Full GPR	0.065	-	0.007	-	0.006	-
J=10	0.796	0.312	0.778	0.253	0.763	0.261
J=30	0.498	0.666	0.466	0.583	0.453	0.587
J=50	0.322	0.836	0.274	0.766	0.268	0.764
J=100	0.134	0.978	0.083	0.940	0.076	0.937
J=150	0.077	0.997	0.028	0.982	0.025	0.981
J=200	0.065	1.000	0.013	0.993	0.011	0.992
J=300	-	-	0.008	0.996	0.006	0.996
J=500	-	-	0.007	0.998	0.006	0.997
J=750	-	-	0.007	0.999	0.006	0.997
J=1000	-	-	0.007	1.000	0.006	0.998
J=1350	-	-	-	-	0.006	0.999
J=1800	-	-	-	-	0.006	1.000

Table 3.1: Average SMSE and CFVE for the case of smooth data

	High Sparsity		Medium Sparsity		Low Sparsity	
	SMSE	CFVE	SMSE	CFVE	SMSE	CFVE
Full GPR	0.809	-	0.338	-	0.108	-
J=10	0.990	0.119	0.990	0.044	0.993	0.034
J=30	0.964	0.277	0.966	0.121	0.969	0.093
J=50	0.936	0.414	0.940	0.192	0.945	0.147
J=100	0.874	0.670	0.869	0.333	0.877	0.274
J=150	0.836	0.888	0.813	0.447	0.808	0.381
J=200	0.809	1.000	0.755	0.543	0.751	0.469
J=300	-	-	0.656	0.697	0.637	0.610
J=500	-	-	0.506	0.877	0.448	0.792
J=750	-	-	0.390	0.971	0.307	0.905
J=1000	-	-	0.338	1.000	0.224	0.958
J=1350	-	-	-	-	0.163	0.989
J=1800	-	-	-	-	0.108	1.000

Table 3.2: Average SMSE and CFVE for the case of rough data

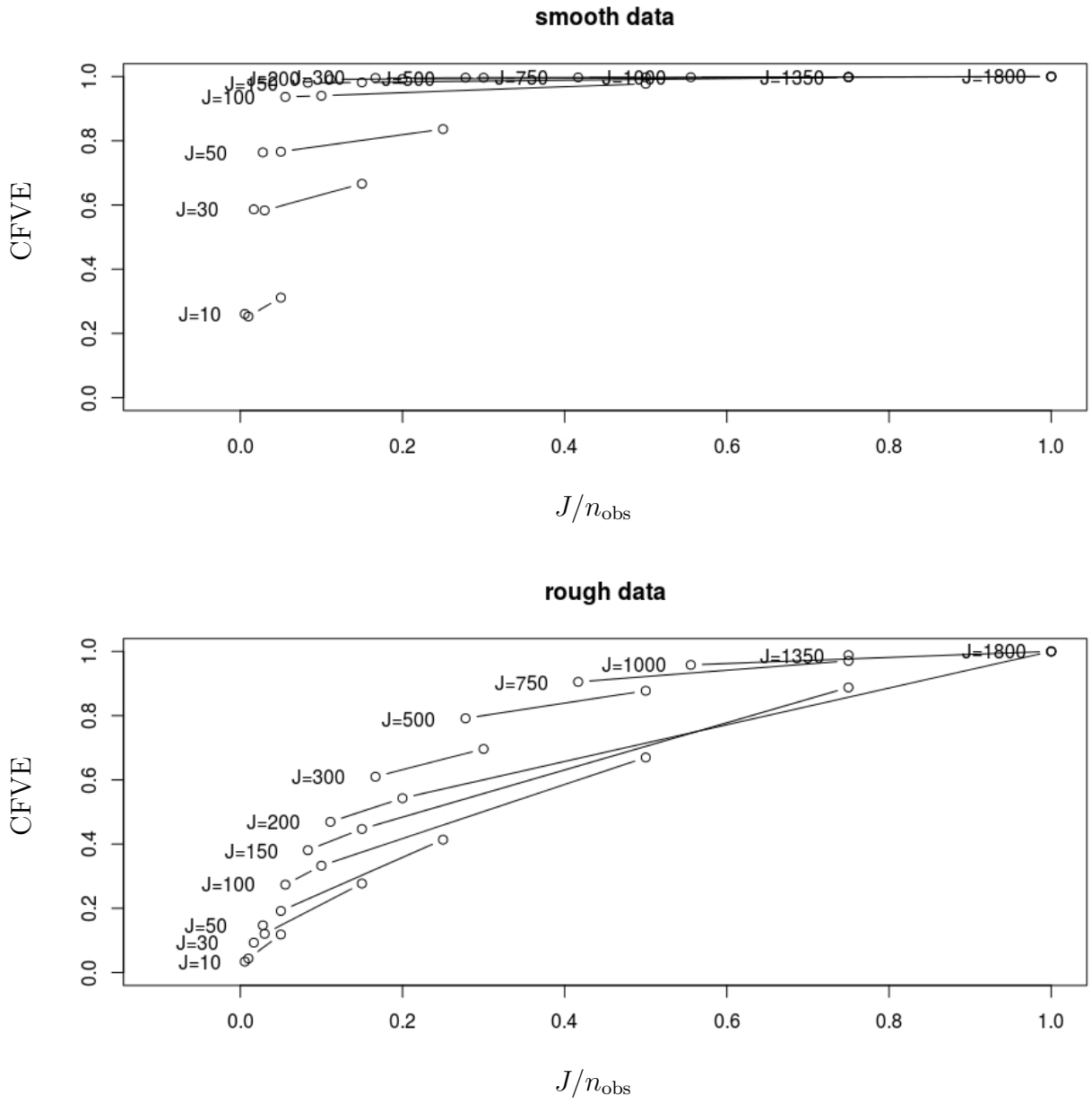

Figure 3.1: CFVEs for smooth curves (top) and rough curves (bottom) plotted for each ratio J/n_{obs}, with fixed J and varying n_{obs}.

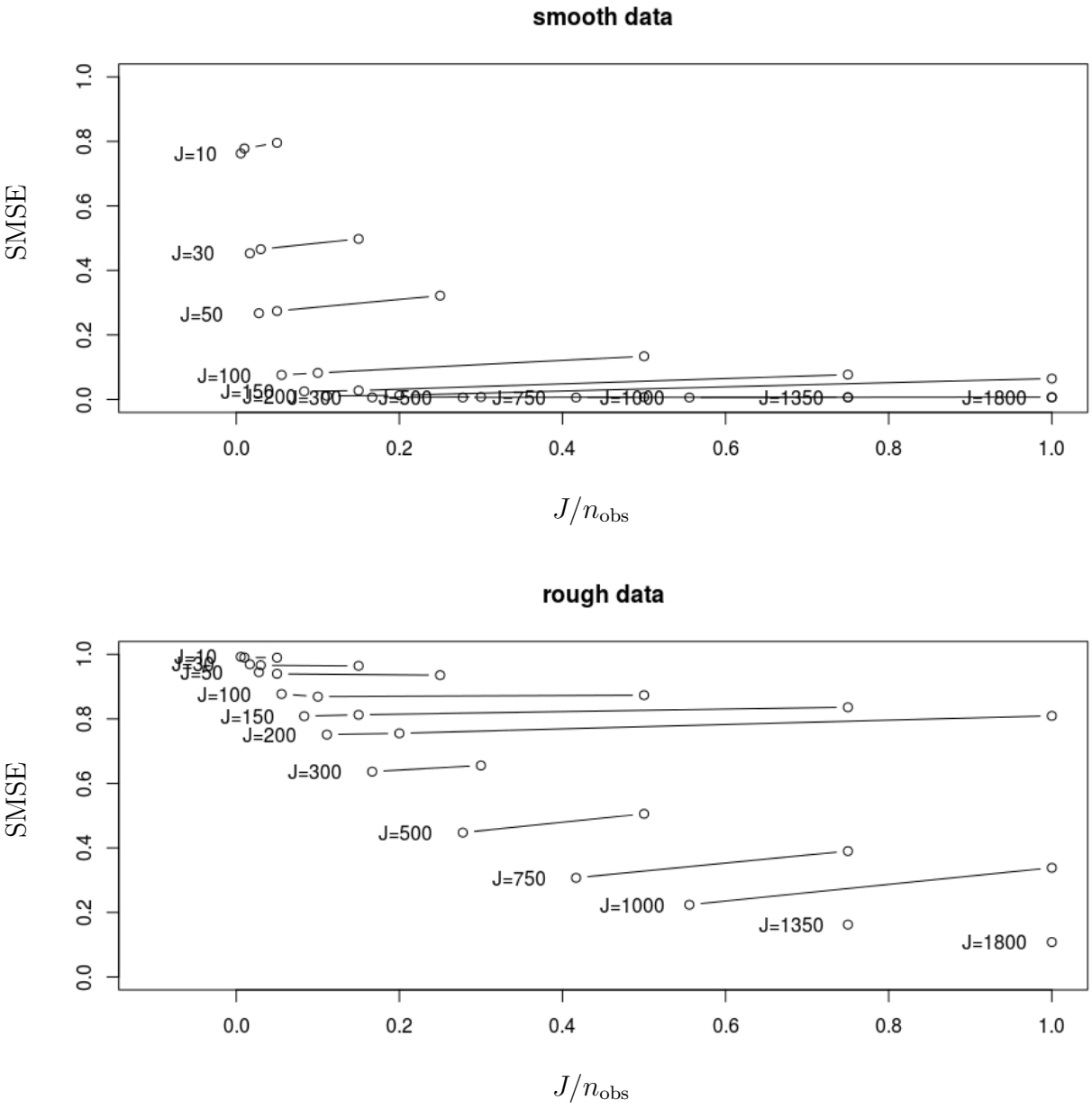

Figure 3.2: SMSEs for smooth curves (top) and rough curves (bottom) plotted for each ratio J/n_{obs}, with fixed J and varying n_{obs}.

Assessing predictions made by approximate implementation methods

We are now interested in comparing predictions made by SD and SR approximation methods with the ones made by the Full GPR. For the approximation methods, we will use different subsets of size m.

In order to consider the predictive distribution (and not only the predictive mean as the SMSE does) to assess predictions, we use the mean standardised log loss (MSLL), defined by

$$\text{MSLL} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \left[\left(\frac{1}{2} \log\left(2\pi\sigma_*^2\right) + \frac{(\boldsymbol{x}_i^* - \hat{\boldsymbol{x}}_i^*)^2}{2\sigma_*^2} \right) - \left(\frac{1}{2} \log\left(2\pi s_*^2\right) + \frac{(\boldsymbol{x}_i^* - \bar{\boldsymbol{x}}^*)^2}{2s_*^2} \right) \right],$$

where $\sigma_*^2 = \text{Var}\left[f(\boldsymbol{t})|\mathcal{D}\right]$ for the Full GPR in (3.6) or the corresponding predicting variance for SR and SD methods as we have discussed above. Observe we subtract a quantity we would have obtained by using a trivial Gaussian distribution model which uses the sample mean $\bar{\boldsymbol{x}}^*$ and sample variance s_*^2 of the training data to predict the test set values. A negative MSLL means we obtain better results than the trivial model.

The data generating process (3.19) is used to simulate 50 datasets where each one has $N = 100$ replicated surfaces. For each dataset we calculate the SMSE and MSLL results, giving rise to the corresponding mean $+/-$ one standard deviation plotted in Figure 3.3. The SR and SD approximation methods use different subsets of size m and the Full GPR model uses all the $n_{\text{obs}} = 1800$ observations available in the training set.

Let us analyse the SMSE results first. For smooth data, we can clearly see that we can use a m much smaller than n_{obs} (for example, SD with $m = 400$ or SR with $m = 200$) to have prediction performance very similar to the one of the Full GPR model. For rough curves, however, we clearly need a larger m. Finally, for both types of data and for a fixed m, the SR approximation outperforms the SD one. The MSLL results show that SD and SR have similar performance for rough curves and that SR is only worse than SD for very small m.

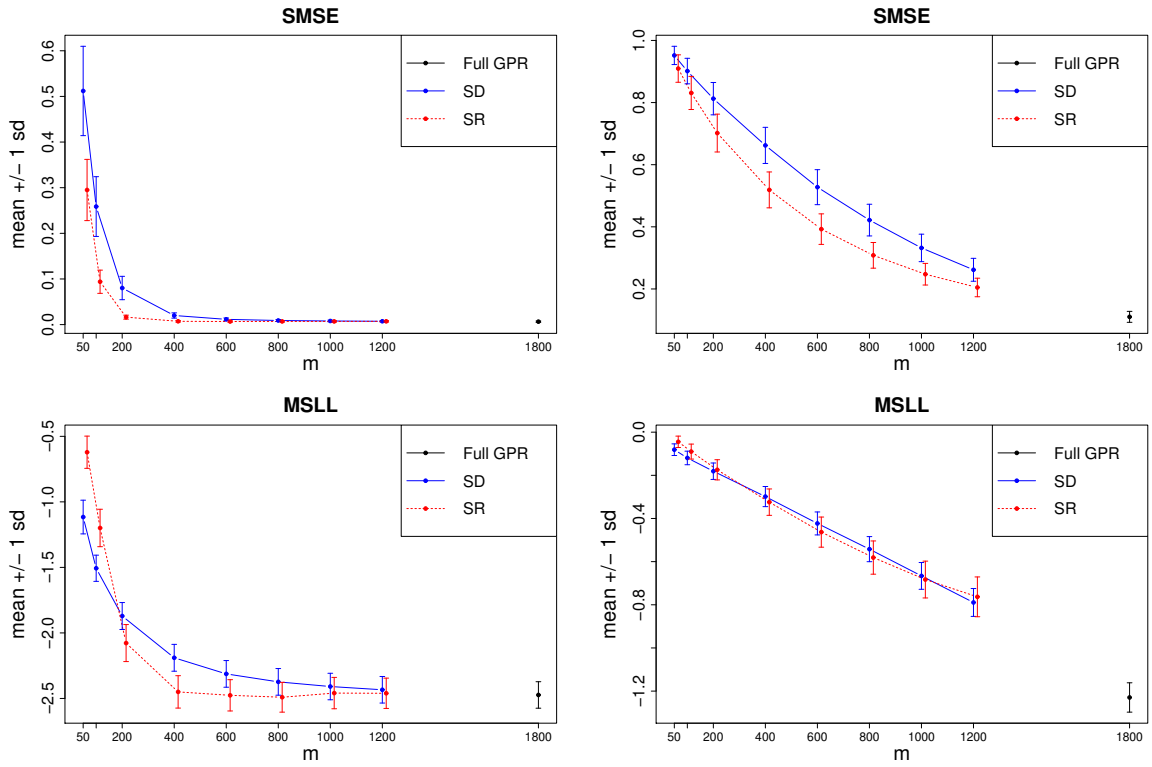

Figure 3.3: Plot of SMSE and MSLL against m for smooth data (left) and for rough data (right). The SR results are horizontally jittered for better visualisation.

It is important to highlight that SD involves a considerably smaller computational time (and storage) than SR does for a fixed m as we can see in Figure 3.4. Therefore, whereas the choice of a suitable m has to take into account the level of smoothness in the data, the choice of the approximation method should consider the fact that SR method requires a quite small ratio m/n_{obs} to reduce the computational time significantly.

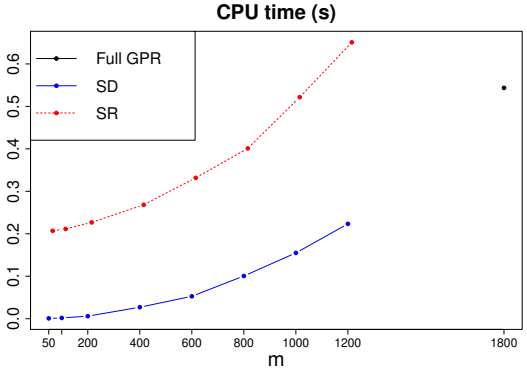

Figure 3.4: Plot of mean CPU time (in seconds) used to evaluate the corresponding loglikelihood function for each m.

3.7 Implementation

The GPR methodology has been implemented in the R language and environment (R Core Team, 2018), where we have written inlined C++ code. As C++ is a statically-typed language, it helps us to reduce the likelihood of bugs. Moreover, the code is optimised based on the memory storage and CPU features.

We can write C++ code in a .cpp file and use the R package Rcpp to embed and compile the C++ code directly in an R script file. Alternatively, we can write a chunk of C++ code in an R script file and use the function cxxfunction() the R package inline (Sklyar *et al.*, 2018) for compilation. We have implemented the latter procedure to make our C++ code available in R.

Our programs use Armadillo C++ linear algebra library (Sanderson, 2010), which provides efficient implementation of elementary operations such as matrix multiplication and matrix factorisations. It also gives us a variety of trigonometric and statistics functions. This library can be used through the R package RcppArmadillo (Eddelbuettel & Sanderson, 2014).

As the computational complexity of evaluating (3.4) is of the order $O(n^3)$, rapid programs are required. Besides writing the code in C++ and use Armadillo C++ library, we make below a few additional remarks.

Due to the symmetry of the $n \times n$ covariance matrix $\boldsymbol{\Psi}_n$, we do not need to evaluate the covariance function at all pairs of points. We only need to calculate $n(n+1)/2$ elements (which correspond to the upper-triangular matrix including the main diagonal) to populate $\boldsymbol{\Psi}_n$.

In addition, we have used the Cholesky decomposition of $\boldsymbol{\Psi}_n$ for the following reasons. Firstly, if we use this decomposition, the inverse $\boldsymbol{\Psi}_n^{-1}$ can be obtained by using the R function chol2inv. This is much faster than applying the commonly used R function solve to calculate the inverse. Secondly, the Cholesky decomposition provides a numerically more stable solution than directly inverting the matrix (Rasmussen & Williams, 2006). Thirdly, the log determinant of $\boldsymbol{\Psi}_n$ can easily be obtained as a by-product of the calculation of the inverse: once the decomposition $\boldsymbol{\Psi} = \boldsymbol{L}^\top \boldsymbol{L}$ is obtained, where $\boldsymbol{L}$ is an upper-triangular matrix, we can quickly calculate $\log|\boldsymbol{\Psi}|$ since $\log|\boldsymbol{\Psi}| = 2\sum_{i=1}^{n} \log \boldsymbol{L}_{ii}$.

In practice, it is often necessary to add a small multiple of the identity matrix, $\zeta \boldsymbol{I}$, to $\boldsymbol{\Psi}$ to improve the numerical conditioning of $\boldsymbol{\Psi}$ (Rasmussen & Williams, 2006). As a result, this has the same effect as including an additional independent noise of variance ζ, which is not a problem as we usually need a very small ζ. In our implementations, we set $\zeta = 10^{-8}$.

In order to save computational time, we can also suppose sparsity for the covariance

matrix as follows:

$$\boldsymbol{\Psi}(\boldsymbol{t}, \boldsymbol{t}') \leftarrow \boldsymbol{\Psi}(\boldsymbol{t}, \boldsymbol{t}') \circ \mathcal{M}(\boldsymbol{t}, \boldsymbol{t}'),$$

where $\circ$ denotes the Schur product and the matrix-valued function $\mathcal{M}(\boldsymbol{t}, \boldsymbol{t}')$ is a modulating function (e.g., equal to 0 whenever $||\boldsymbol{t} - \boldsymbol{t}'||_1 > \gamma$ and 1 otherwise, also known as *taper function*). Figure 3.5 illustrates a covariance matrix of a given GPR model where input t is unidimensional for illustration purposes. In practice, it is often acceptable to set to zero the elements whose distance from the diagonal is large, with that distance being determined by the tuning parameter γ. The choice of γ is discussed in (Pourahmadi, 2013, Chapter 6) and references therein. In our implementations, we only used tapering of covariance matrices in simulation studies of stationary data where we knew that most of the true elements were smaller than 10^{-8}, elements which were set to zero.

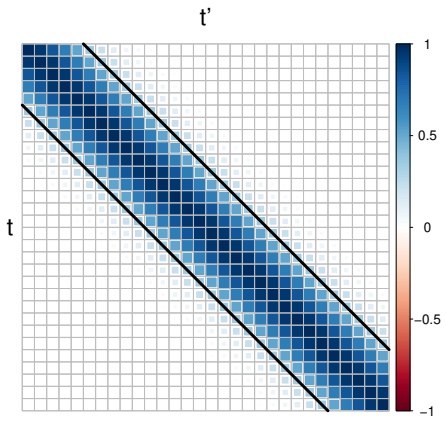

Figure 3.5: Sparsity assumed for a covariance matrix $\boldsymbol{\Psi}(t, t')$. Values which are more distant to the matrix diagonal than the black diagonal lines are set to zero.

3.8 Conclusion

As we have seen, the Nyström-based method for approximating a covariance matrix $\boldsymbol{\Psi}_n$ is a reduced-rank method which uses only $m < n$ eigenvectors and provide very good approximation for a quite small m especially when the spectrum of $\boldsymbol{\Psi}_n$ decays fast, i.e. when all except the very first few eigenvalues are very close to zero. Our results indicate that SR approximation, in general, provides better predictions than SD does for the same m. However, if the number of observations in each curve is too large, in such a way that Nyström's computational complexity of $\mathcal{O}(m^2 n)$ is still prohibitive, SD approximation method might be a suitable alternative.

The asymptotic theory shows important results. Theorems 1 and 2 provide the best

finite dimensional approximation and convergence rates under a very general covariance structure of the function-valued process. Theorem 2 indicates that Karhunen-Loéve expansion may be used to analyse the eigenfunctions and eigenvalues as we do in FDA. As our framework allows a general covariance structure, the eigenfunctions can provide us new insights into the data as we do not need to assume covariance separability, a topic which will be discussed in Chapter 4.

3.9 Appendix

Proof of Theorem 1

We show that

$$
\begin{aligned}
&\mathrm{E}\Big[||X^c(\boldsymbol{t}) - \sum_{j=1}^{J} g_j(\boldsymbol{t})\xi_j^*||^2\Big] \\
&= \mathrm{E}\langle X^c(\cdot) - \sum_{j=1}^{J} g_j(\cdot)\xi_j^*, X^c(\cdot) - \sum_{j=1}^{J} g_j(\cdot)\xi_j^*\rangle \\
&= \mathrm{E}\Big[\langle X^c(\cdot), X^c(\cdot)\rangle - \sum_{j=1}^{J}\langle X^c(\cdot), g_j(\cdot)\rangle^2\Big] \\
&= \mathrm{E}||X^c||^2 - \sum_{j=1}^{J}\mathrm{E}\Big[\int_{\mathcal{T}}\int_{\mathcal{T}} X^c(\boldsymbol{t})X^c(\boldsymbol{t}')g_j(\boldsymbol{t})g_j(\boldsymbol{t}')d\boldsymbol{t}d\boldsymbol{t}'\Big] \\
&= \mathrm{E}||X^c||^2 - \sum_{j=1}^{J}\int_{\mathcal{T}}\int_{\mathcal{T}} k(\boldsymbol{t},\boldsymbol{t}')g_j(\boldsymbol{t})g_j(\boldsymbol{t}')d\boldsymbol{t}d\boldsymbol{t}' \\
&= \mathrm{E}||X^c||^2 - \sum_{j=1}^{J}\langle \Xi(g_j), g_j\rangle.
\end{aligned}
$$

Hence, the minimising problem (3.13) becomes to maximise $\sum_{j=1}^{J}\langle \Xi(g_j), g_j\rangle$ with respect to $g_1, \ldots, g_J \in L^2(\mathcal{T})$. Since the operator Ξ is symmetric, positive definite Hilbert-Schmidt, following Theorem 3.2 in Horváth & Kokoszka (2012) the proof is completed.

Proof of Theorem 2

Without loss of generality, we consider $Q = 2$. Then $\boldsymbol{t} = (s, \tau)^\top$ with $s, \tau \in \mathbb{R}$. Let $\{\hat{\lambda}_j, j = 1, 2, \ldots\}$ and $\{\hat{\phi}_j(\cdot), j = 1, 2, \ldots\}$ be the eigenvalues and eigenfunctions of the

covariance function $\hat{k}(\boldsymbol{t},\boldsymbol{t}') = k_{\hat{\theta}}(\boldsymbol{t},\boldsymbol{t}')$, where $\boldsymbol{t} = (s,\tau)^\top,\ \boldsymbol{t}' = (s',\tau')^\top$, and

$$\hat{\xi}_j = \langle X^c(\cdot), \hat{\phi}_j(\cdot)\rangle = \int X^c(\boldsymbol{t})\hat{\phi}_j(\boldsymbol{t})d\boldsymbol{t}.$$

Let $p(\boldsymbol{x}_l^c;\boldsymbol{\theta}) = p(x_1^c,\ldots,x_l^c;\boldsymbol{\theta})$ be the density function of $\boldsymbol{x}_l^c$. Let $\boldsymbol{\theta}_0$ be the true value of $\boldsymbol{\theta}$ and $p_l(\boldsymbol{\theta})$ be the conditional density of $\boldsymbol{x}_l^c$ for given $\boldsymbol{x}_{l-1}^c$. Actually, for every $l \geq 1$,

$$p_l(\boldsymbol{\theta}) = p(\boldsymbol{x}_l^c;\boldsymbol{\theta})/p(\boldsymbol{x}_{l-1}^c;\boldsymbol{\theta}).$$

It shows that $p_l(\boldsymbol{\theta}) = N(\mu_{l|l-1}, \sigma^2_{l|l-1})$ with

$$\mu_{l|l-1} = \boldsymbol{k}_l\boldsymbol{K}_{l-1}^{-1}\boldsymbol{x}_{l-1}^c,\quad \sigma^2_{l|l-1} = k(\boldsymbol{t}_l,\boldsymbol{t}_l) - \boldsymbol{k}_l\boldsymbol{K}_{l-1}^{-1}\boldsymbol{k}_l^\top,$$

where $\boldsymbol{k}_l = \big(k(\boldsymbol{t}_1,\boldsymbol{t}_l),\ldots,k(\boldsymbol{t}_{l-1},\boldsymbol{t}_l)\big)^\top$. Assume that $p_l(\boldsymbol{\theta})$ is twice differentiable with respect to $\boldsymbol{\theta}$. Let $\phi_l(\boldsymbol{\theta}) = \log p_l(\boldsymbol{\theta})$, $\boldsymbol{U}_l(\boldsymbol{\theta}) = \dot{\phi}_l(\boldsymbol{\theta})$ and $\boldsymbol{V}_l(\boldsymbol{\theta}) = \ddot{\phi}_l(\boldsymbol{\theta})$, where $\dot{g}$ and $\ddot{g}$ are the first and second derivatives of function $g(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, respectively. Without loss of generality, we consider the parameter with one dimension. Then $\boldsymbol{U}_l(\boldsymbol{\theta})$ and $\boldsymbol{V}_l(\boldsymbol{\theta})$ are scalars $U_l(\theta)$ and $V_l(\theta)$, and denoted by $U_l = U_l(\theta_0)$ and $V_l = V_l(\theta_0)$. For the proof of Theorem 2, we need the following conditions:
(C1) $\sup_{s,\tau}|\mu(s,\tau)| < \infty$.
(C2) The covariance function $k_\theta(\boldsymbol{t},\boldsymbol{t}')$ has thrice continuous derivative with respect to θ, and is continuous, differentiable and square-integrable on $\boldsymbol{t},\boldsymbol{t}'$. For eigenvalues and eigenvectors of k_θ, assume $\delta_j > 0$ and $\phi_j(s,\tau)$ is square-integrable, where $\delta_j = \min\{\lambda_1 - \lambda_2, \lambda_{j-1} - \lambda_j, \lambda_j - \lambda_{j+1}\}$.

Define $i_k(\theta_0) = \mathrm{Var}[U_k|\mathscr{F}_{k-1}] = \mathrm{E}[U_k^2|\mathscr{F}_{k-1}]$, where $\mathscr{F}_{k-1} = \sigma(x_1^c,\ldots,x_{l-1}^c)$. Let $I_n(\theta_0) = \sum_{k=1}^n i_k(\theta_0)$, $S_n = \sum_{k=1}^n U_k$ and $S_n^* = \sum_{k=1}^n V_k + I_n(\theta_0)$. It shows that S_n and S_n^* are zero-mean martingales with respect to σ-filtration $\mathscr{F}_n$. The third condition is
(C3) Assume

1. $n^{-1}|\sum_{k=1}^n V_k| \xrightarrow{p} i(\theta_0)$, and $n^{-1/2}S_n \xrightarrow{L} N(0, i(\theta_0))$ for some non-random function $i(\theta_0) > 0$,

2. For all $\varepsilon > 0$ and $\eta > 0$, there exists $\delta > 0$ and $n_0 > 0$ such that for all $n > n_0$, $P\{n^{-1}|\sum_{k=1}^n (V(\theta) - V_k)| > \eta, |\theta - \theta_0| < \delta\} < \varepsilon$,

3. $n^{-1}\sum_{k=1}^n \mathrm{E}|W_k(\theta)| < M < \infty$ for all θ and n, where $W_k(\theta)$ is the third derivative of $\phi_k(\theta)$ with respect to θ.

Under conditions (C2) and (C3), it easily shows that the conditions of Theorem 2.2 in Chapter 7 of Basawa & Prakasa Rao (1980) holds. Hence, $\hat{\theta}$ is a consistent estimator

of θ_0 and has asymptotically normality,

$$n^{-1/2}(\hat{\theta} - \theta_0) \xrightarrow{L} N(0, i(\theta_0)^{-1}),$$

which indicates that

$$||\hat{\theta} - \theta_0|| = O_p(\{\log(n)/n\}^{1/2}).$$

Since covariance function k_θ is thrice continuously differentiate on θ, we have

$$||k_{\hat{\theta}}(\cdot,\cdot) - k_\theta(\cdot,\cdot)|| = O_p(\{\log(n)/n\}^{1/2}).$$

From Lemma 4.2 in Bosq (2000), it follows that for all j,

$$||\hat{\lambda}_j - \lambda_j|| \leq ||k_{\hat{\theta}}(\cdot,\cdot) - k_\theta(\cdot,\cdot)||, \tag{3.20}$$

and similar to Lemma 4.3 in Bosq (2000), we have for fixed j,

$$||\hat{\phi}_j(\cdot) - \phi_j(\cdot)|| \leq 2\sqrt{2}\delta_j^{-1}||k_{\hat{\theta}}(\cdot,\cdot) - k_\theta(\cdot,\cdot)||, \tag{3.21}$$

where $\delta_j = \min\{\lambda_1 - \lambda_2, \lambda_{j-1} - \lambda_j, \lambda_j - \lambda_{j+1}\}$. Then (3.20) and (3.21) give that

$$\begin{aligned}
&||\hat{\lambda}_j - \lambda_j|| = O_p(\{\log(n)/n\}^{1/2}),\\
&||\hat{\phi}_j(\cdot) - \phi_j(\cdot)|| = O_p(\{\log(n)/n\}^{1/2}).
\end{aligned}$$

For ξ_j, we show that

$$\begin{aligned}
&|\hat{\xi}_j - \xi| = \left|\int (Z(s,\tau) - \hat{\mu}(s,\tau))\hat{\phi}_j(s,\tau)dsd\tau - \int (Z(s,\tau) - \mu(s,\tau))\phi_j(s,\tau)dsd\tau\right|\\
\leq &\left|\int (Z(s,\tau) - \mu(s,\tau))(\hat{\phi}_j(s,\tau) - \phi_j(s,\tau))dsd\tau\right|\\
&+ \left|\int (\hat{\mu}(s,\tau) - \mu(s,\tau))(\hat{\phi}_j(s,\tau) - \phi_j(s,\tau))dsd\tau\right|\\
&+ \left|\int (\hat{\mu}(s,\tau) - \mu(s,\tau))\phi_j(s,\tau)dsd\tau\right|.
\end{aligned}$$

Hence, from condition C2, (3.20) and $\sup_{s,\tau}|\hat{\mu}(s,\tau) - \mu(s,\tau)| = O_p[\{\log(n)/n\}^{1/2}]$, it shows that

$$||\hat{\xi}_j - \xi_j|| = O_p(\{\log(n)/n\}^{1/2}).$$

Proof of Theorem 3

Let $\boldsymbol{K}_n = \big(k(\boldsymbol{t}_i, \boldsymbol{t}_j)\big)_{n\times n}$ be a Gram matrix, and $\lambda_1^{(n)} \geq \lambda_2^{(n)} \geq \cdots \geq \lambda_n^{(n)} \geq 0$ be the eigenvalues of $\boldsymbol{K}_n$, and $\boldsymbol{V}_{j,n},\ j = 1, \ldots, n$ be the eigenvectors of $\boldsymbol{K}_n$. Then from the Nyström approximation method, we show that

$$\begin{aligned} &\sqrt{n}V_{hj,n} = \phi_j(\boldsymbol{t}_h) + O_p\Big(\frac{1}{\sqrt{n}}\Big), \qquad \frac{\lambda_j^{(n)}}{n} = \lambda_j + O_p\Big(\frac{1}{\sqrt{n}}\Big), \\ &\frac{\sqrt{n}}{\lambda_j^{(n)}}\boldsymbol{k}_n^\top(\boldsymbol{t})\boldsymbol{V}_{j,n} = \phi_j(\boldsymbol{t}) + O_p\Big(\frac{1}{\sqrt{n}}\Big), \end{aligned} \tag{3.22}$$

where $V_{hj,n}$ is the h-th element of $\boldsymbol{V}_{j,n}$, and $\boldsymbol{k}_n(\boldsymbol{t}) = \big(k(\boldsymbol{t}_1, \boldsymbol{t}), \ldots, k(\boldsymbol{t}_n, \boldsymbol{t})\big)^\top$. Due to $\mathrm{E}[\xi_j] = 0$ and $\mathrm{Var}[\xi_j] = \lambda_j \to 0$ as $j \to \infty$, it follows from (3.22) that

$$f_n(\boldsymbol{t}) = \sum_{j=1}^{n} \phi_j(\boldsymbol{t})\xi_j = \sum_{j=1}^{n} \frac{\sqrt{n}}{\lambda_j^{(n)}}\boldsymbol{k}_n^\top(\boldsymbol{t})\boldsymbol{V}_{j,n}\xi_j + o_p(1).$$

In addition, we show that

$$\begin{aligned} \xi_j =& \lambda_j\langle f, \phi_j\rangle \\ =& \lambda_j\langle f, \frac{\sqrt{n}}{\lambda_j^{(n)}}\boldsymbol{k}_n^\top(\cdot)\boldsymbol{V}_{j,n}\rangle + O_p\Big(\frac{1}{\sqrt{n}}\Big) \\ =& \frac{\sqrt{n}\lambda_j}{\lambda_j^{(n)}}\boldsymbol{V}_{j,n}^\top\langle f, \boldsymbol{k}_n(\cdot)\rangle + O_p\Big(\frac{1}{\sqrt{n}}\Big) \\ =& \frac{\sqrt{n}\lambda_j}{\lambda_j^{(n)}}\boldsymbol{V}_{j,n}^\top\boldsymbol{f} + O_p\Big(\frac{1}{\sqrt{n}}\Big). \end{aligned}$$

Hence, we have

$$\begin{aligned} f_n(\boldsymbol{t}) =& \sum_{j=1}^{n} \frac{\sqrt{n}}{\lambda_j^{(n)}}\boldsymbol{k}_n^\top(\boldsymbol{t})\boldsymbol{V}_{j,n}\frac{\sqrt{n}\lambda_j}{\lambda_j^{(n)}}\boldsymbol{V}_{j,n}^\top\boldsymbol{f} + o_p(1) \\ =& \boldsymbol{k}_n^\top(\boldsymbol{t})\sum_{j=1}^{n}\frac{1}{\lambda_j^{(n)}}\boldsymbol{V}_{j,n}\boldsymbol{V}_{j,n}^\top\boldsymbol{f} + o_p(1) \\ =& \boldsymbol{k}_n^\top(\boldsymbol{t})\boldsymbol{K}_n^{-1}\boldsymbol{f} + o_p(1), \end{aligned}$$

which indicates that $\mathrm{E}[f(\boldsymbol{t})|\boldsymbol{f}] = \boldsymbol{k}_n^\top(\boldsymbol{t})\boldsymbol{K}_n^{-1}\boldsymbol{f} = f_n(\boldsymbol{t}) + o_p(1)$.

The Nyström approximation is also applied to $X(\boldsymbol{t})$ and $\varepsilon(\boldsymbol{t})$, respectively, and we

have

$$X_n(\boldsymbol{t}) = \tilde{\boldsymbol{k}}_n^\top(\boldsymbol{t})\tilde{\boldsymbol{K}}_n^{-1}\boldsymbol{x} + o_p(1),$$
$$\varepsilon_n(\boldsymbol{t}) = \boldsymbol{k}_{\varepsilon n}^\top(\boldsymbol{t})\boldsymbol{K}_{\varepsilon n}^{-1}\boldsymbol{\varepsilon} + o_p(1),$$

where $\boldsymbol{x} = \big(x(\boldsymbol{t}_1), \ldots, x(\boldsymbol{t}_n)\big)^\top$, $\boldsymbol{\varepsilon} = \big(\varepsilon(\boldsymbol{t}_1), \ldots, \varepsilon(\boldsymbol{t}_n)\big)^\top$, $\tilde{k} = k + k_\varepsilon$, $\big[\tilde{\boldsymbol{K}}_n\big]_{ij} = \tilde{k}(\boldsymbol{t}_i, \boldsymbol{t}_j)$, $\tilde{\boldsymbol{k}}_n(\boldsymbol{t}) = \big(\tilde{k}(\boldsymbol{t}_1, \boldsymbol{t}), \ldots, \tilde{k}(\boldsymbol{t}_n, \boldsymbol{t})\big)^\top$, $\big[\boldsymbol{K}_{\varepsilon n}\big]_{ij} = k_\varepsilon(\boldsymbol{t}_i, \boldsymbol{t}_j)$, and $\boldsymbol{k}_{\varepsilon n}(\boldsymbol{t}) = \big(k_\varepsilon(\boldsymbol{t}_1, \boldsymbol{t}), \ldots, k_\varepsilon(\boldsymbol{t}_n, \boldsymbol{t})\big)^\top$. From the definition of k_ε, we know that $\boldsymbol{k}_{\varepsilon n}(\boldsymbol{t}) = \boldsymbol{0}$ and $\tilde{\boldsymbol{K}}_n = \boldsymbol{K}_n + \sigma_\varepsilon^2 \boldsymbol{I}_n$. Hence, it follows that

$$f_n(\boldsymbol{t}) = X_n(\boldsymbol{t}) - \varepsilon_n(\boldsymbol{t}) = \boldsymbol{k}_n^\top(\boldsymbol{t})(\boldsymbol{K}_n + \sigma_\varepsilon^2 \boldsymbol{I}_n)^{-1}\boldsymbol{x} + o_p(1),$$

which suggests that

$$\mathrm{E}\left[f(\boldsymbol{t})|\mathcal{D}\right] = \boldsymbol{k}_n^\top(\boldsymbol{t})(\boldsymbol{K}_n + \sigma_\varepsilon^2 \boldsymbol{I}_n)^{-1}\boldsymbol{x} = X_n(\boldsymbol{t}) + o_p(1).$$

Chapter 4

Modelling function-valued processes with nonstationary, nonseparable covariance structure

When $\boldsymbol{t}$ is multi-dimensional, a general nonparametric covariance cannot usually be used due to the curse of dimensionality. One way to address the problem is to assume a separable covariance function

$$k(\boldsymbol{t}, \boldsymbol{t}') = k_1(t_1, t_1') \cdots k_Q(t_Q, t_Q'), \tag{4.1}$$

that is, if it can be factorised into the product between covariance functions, each one corresponding to one dimension, then it can be modelled nonparametrically (see e.g. Chen *et al.*, 2017; Rougier, 2017).

In this chapter, we propose a semiparametric approach for the estimation of a flexible covariance function in such a way we can relax the assumptions of stationarity and separability. The nonstationarity over $\boldsymbol{t}$ is defined by a convolution-based approach via a varying kernel, whose parameters are modelled nonparametrically. In particular, we propose to use a suitable parametrisation for the varying anisotropy matrix, allowing for unconstrained estimation.

Section 4.1 discusses the covariance separability assumption made in FPCA models. In Section 4.2, we define a nonstationary covariance structure and propose to use a spherical parametrisation for the varying anisotropy matrix. Simulation studies are presented in Section 4.3 and an application to Canadian temperature data in Section 4.4.

4.1 Function-valued processes with separable covariance structure

The accurate estimate of the covariance function, which is important for FPCA and other inference methods of functional data analysis (Ramsay & Silverman, 2005), is a challenging task. When the dimension of the input space is $Q = 2$, the covariance function depends on four arguments and, in the case of sparse designs, nonparametric estimation may suffer from the curse of dimensionality and slow computing. These difficulties are rapidly aggravated as Q becomes larger.

In order to address these issues, many models for two-way functional data (e.g. Chen & Müller (2012); Allen *et al.* (2014); Chen *et al.* (2017)) and spatiotemporal data (Banerjee *et al.* (2015) and references therein) assume that the covariance function $k(\boldsymbol{t}, \boldsymbol{t}')$ is separable. In other words, they assume that $k(\boldsymbol{t}, \boldsymbol{t}')$ can be factorised into the product between Q covariance functions, each one corresponding to one coordinate direction.

Let us discuss the covariance separability assumption for the case $\boldsymbol{t} = (s, \tau)^\top \in \mathbb{R}^2$ to simplify the exposition. According to Mercer's theorem, the marginal covariance functions can be decomposed as

$$k_1(s, s') = \lim_{J\to\infty} \sum_{j=1}^{J} \lambda_{1j}\phi_{1j}(s)\phi_{1j}(s')$$

and

$$k_2(\tau, \tau') = \lim_{J\to\infty} \sum_{l=1}^{J} \lambda_{2l}\phi_{2l}(\tau)\phi_{2l}(\tau'),$$

analogously as the decomposition of the full covariance function $k(\boldsymbol{t}, \boldsymbol{t}')$ (see eq. (2.2)), determined by the eigenvalues λ_j and eigenfunctions $\phi_j(\boldsymbol{t})$.

If the covariance function $k(s, \tau; s', \tau')$ is separable, then

$$\begin{aligned} k(s, \tau; s', \tau') &= k_1(s, s')k_2(\tau, \tau') \\ &= \lim_{J\to\infty} \sum_{j=1}^{J}\sum_{l=1}^{J} \lambda_{1j}\lambda_{2l}\phi_{1j}(s)\phi_{2l}(\tau)\phi_{1j}(s')\phi_{2l}(\tau'). \end{aligned}$$

(Rougier, 2017). In this case, for every j and l, $\lambda_{1j}\lambda_{2l}$ is an eigenvalue of $k(\boldsymbol{t}, \boldsymbol{t}')$ and $\phi_{1j}(s)\phi_{2l}(\tau)$ the corresponding eigenfunction. Note that covariance separability implies separability of eigenfunctions. If we observe nonseparable eigenfunctions, then the covariance function must not be separable, and this is another reason to visualise the eigenfunctions. Note that multidimensional eigenfunctions can also be called eigensurfaces and thus we can use these terms interchangeably.

The covariance separability assumption allows efficient computation of the eigenfunctions and functional principal components because we only need to compute the eigenfunctions of the marginal kernels. Moreover, it reduces the *curse of dimensionality* problem that is present when estimating nonparametrically the $(2Q)$-dimensional covariance function. Based on these facts, Chen *et al.* (2017) suggest using tensor product representations. In the model that they called Product FPCA, the two-dimensional function-valued process X is represented as

$$X(s,\tau) = \mu(s,\tau) + \sum_{j=1}^{\infty}\sum_{l=1}^{\infty} \xi_{jl}\phi_{1j}(s)\phi_{2l}(\tau), \tag{4.2}$$

where ϕ_1 and ϕ_2 are eigenfunctions of the marginal covariance functions. This model implicitly assumes that the covariance structure is separable.

Chen *et al.* (2017) show that the Product FPCA model has nearly optimal solution in terms of maximising (2.6) under appropriate assumptions. In the application to human fertility data (Human Fertility Database, 2017), they also show that the model is useful as it can be used to analyse the effects of the two inputs separately.

For a three-dimensional process X (see simulation study in subsection 4.3.2), we will consider a natural extension of the Product FPCA model (4.2) and represent X as

$$X(\tau,s_1,s_2) = \mu(\tau,s_1,s_2) + \sum_{j=1}^{\infty}\sum_{l=1}^{\infty}\sum_{m=1}^{\infty} \xi_{jlm}\phi_{1j}(\tau)\phi_{2l}(s_1)\phi_{3m}(s_2).$$

Besides reducing computational costs and offering attractive interpretation, the covariance separability assumption is also useful because it makes it easier to guarantee positive definiteness of the covariance function. However, it does not allow any interaction between the coordinate directions in the covariance, an assumption that may be too strong for applications to data with complex covariance structure. This has motivated recent development of hypothesis tests for separability (Aston *et al.*, 2017; Constantinou *et al.*, 2017; Chen & Lynch, 2017; Cappello *et al.*, 2018).

We can instead use GPR models with parametric covariance function which allows for nonseparability. Although a parametric covariance function may not fit the covariance structure very well, this can be overcome by selecting it from a variety of families of flexible covariance functions.

4.2 Function-valued processes with nonseparable and/or nonstationary covariance structure

Although general classes of nonseparable covariance functions were proposed almost two decades ago (Cressie & Huang, 1999; Gneiting, 2002; De Iaco *et al.*, 2002; Stein, 2005), they are usually restricted to the scope of stationarity, another assumption we wish to avoid. We would like to implement an approach which not only offers attractive interpretation, but which is also flexible to model complex, nonstationary covariance structure.

When using a stationary covariance function, we assume a constant variance for the entire input domain. Suppose that there are two regions of the domain, each one having data with different degree of variability. If the data show larger variability in the first region, the variance estimate for the second region tends to be inflated, resulting in predictions with too large variance for the second region.

In order to handle this, we need to employ nonstationary covariance functions which desirably accommodate a wide class of nonstationary stochastic processes. In addition, the choice of the covariance function k must be valid.

The linear covariance function (Shi & Choi, 2011) is an example of nonstationary covariance function. Its simplicity, though, is of limited use for modelling complex covariance structures and it is often used together with other covariance functions (e.g. Wang & Shi (2014)). A popular strategy to deal with nonstationarity in spatial statistics literature is through *deformation* (Sampson & Guttorp, 1992). The idea is to transform the geographical space into another space where stationarity holds. The choice of a suitable transformation is a challenging task. In addition, this approach requires independent replications of the spatial process (Banerjee *et al.*, 2015). Another method called *treed* GP (Gramacy & Lee, 2008) splits the input space into subregions. This requires the choice of the number of subregions and the split locations, which might not be trivial. Ba & Joseph (2012) propose to model X as a sum of a global GP and a local GP, where the nonstationarity is introduced through scaling of the local stationary process. We may wish to consider the case where the nonstationarity is defined by a varying anisotropy matrix, as we will see in the next subsection.

4.2.1 Convolution-based covariance functions

Higdon *et al.* (1999) propose a constructive, convolution-based approach to account for nonstationarity in the covariance function. They represent a spatial process as a moving

average of Gaussian white noise processes $z(\cdot)$ as

$$f(\boldsymbol{t}) = \int_{\mathbb{R}^2} k_{\boldsymbol{t}}(\boldsymbol{u}) z(\boldsymbol{u}) d\boldsymbol{u}, \tag{4.3}$$

where the nonstationarity is achieved by considering a spatially-varying kernel $k_{\boldsymbol{t}}$. The covariance function of (4.3) takes the form

$$\mathrm{Cov}\big[f(\boldsymbol{t}), f(\boldsymbol{t}')\big] = \int_{\mathbb{R}^2} k_{\boldsymbol{t}}(\boldsymbol{u}) k_{\boldsymbol{t}'}(\boldsymbol{u}) d\boldsymbol{u} \tag{4.4}$$

and is positive definite provided that $\sup \int_{\mathbb{R}^2} k_{\boldsymbol{t}}(\boldsymbol{u})^2 d\boldsymbol{u} \leq \infty$.

The convolution-based approach has become popular mainly because specifying a kernel which satisfies the above condition is much easier than specifying a covariance function directly. Higdon (2002) suggests different process convolution specifications to build flexible space and space-time models.

Paciorek & Schervish (2006) show that the covariance function (4.4) is valid in every Euclidean space $\mathbb{R}^Q$, $Q = 1, 2, \ldots$. They also note that if we assume a Gaussian kernel $k_{\boldsymbol{t}}(\boldsymbol{u}) = (2\pi)^{-Q/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\big\{-(1/2)(\boldsymbol{t}-\boldsymbol{u})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{t}-\boldsymbol{u})\big\}$, the covariance function of $f(\cdot)$ will be

$$\mathrm{Cov}\big[f(\boldsymbol{t}), f(\boldsymbol{t}')\big] = \sigma^2 |\boldsymbol{\Sigma}(\boldsymbol{t})|^{1/4} |\boldsymbol{\Sigma}(\boldsymbol{t}')|^{1/4} \left| \frac{\boldsymbol{\Sigma}(\boldsymbol{t}) + \boldsymbol{\Sigma}(\boldsymbol{t}')}{2} \right|^{-1/2} \exp\{-Q_{\boldsymbol{t}\boldsymbol{t}'}\}, \tag{4.5}$$

where

$$Q_{\boldsymbol{t}\boldsymbol{t}'} = (\boldsymbol{t} - \boldsymbol{t}')^\top \left(\frac{\boldsymbol{\Sigma}(\boldsymbol{t}) + \boldsymbol{\Sigma}(\boldsymbol{t}')}{2} \right)^{-1} (\boldsymbol{t} - \boldsymbol{t}').$$

A more general class for nonstationary covariance functions given by

$$\mathrm{Cov}\big[f(\boldsymbol{t}), f(\boldsymbol{t}')\big] = \sigma(\boldsymbol{t})\sigma(\boldsymbol{t}') |\boldsymbol{\Sigma}(\boldsymbol{t})|^{1/4} |\boldsymbol{\Sigma}(\boldsymbol{t}')|^{1/4} \left| \frac{\boldsymbol{\Sigma}(\boldsymbol{t}) + \boldsymbol{\Sigma}(\boldsymbol{t}')}{2} \right|^{-1/2} g\Big(\sqrt{Q_{\boldsymbol{t}\boldsymbol{t}'}}\Big), \tag{4.6}$$

where $g(\cdot)$ is a valid isotropic correlation function.

Even if the anisotropy matrix is assumed to be constant ($\boldsymbol{\Sigma}(\boldsymbol{t}) = \boldsymbol{\Sigma}$), the covariance function (4.6) is nonstationary. In this special case, the nonstationarity is introduced through scaling of a stationary process (Banerjee *et al.*, 2015, Section 3.2). In other words, if a stationary process $V(\boldsymbol{t})$ has mean 0, variance 1 and correlation function ρ, then $Z(\boldsymbol{t}) = \sigma(\boldsymbol{t})V(\boldsymbol{t})$ is a nonstationary process with covariance function $\mathrm{Cov}\big[Z(\boldsymbol{t}), Z(\boldsymbol{t}')\big] = \sigma(\boldsymbol{t})\sigma(\boldsymbol{t}')\rho(\boldsymbol{t} - \boldsymbol{t}')$. The composite Gaussian process model (Ba & Joseph, 2012) also uses this idea to allow for varying volatility in a local GP process.

The anisotropy matrix $\boldsymbol{\Sigma}(\boldsymbol{t})$ measures how quickly varying is the fluctuation of the

random processes over $\boldsymbol{t}$ and one may want to allow it to vary over $\boldsymbol{t}$. Both $\sigma(\cdot)$ and $\boldsymbol{\Sigma}(\cdot)$ can also vary over $\boldsymbol{\tau} \in \mathcal{T}^* \subset \mathbb{R}^{Q^*}$, where $Q^* \leq Q$. This $\boldsymbol{\tau}$ can represent, for example, time or spatial coordinates, accounting for time-varying or spatially-varying parameters, or both. This provides a flexible way to model nonstationary and nonseparable covariance structure. We will use the observed data to estimate the covariance structure nonparametrically. The details will be discussed in the next subsection.

If g is, for example, a (squared) exponential function, it is easy to see that if and only if we can factorise $\sigma(\boldsymbol{t}) = \sigma(t_1) \cdots \sigma(t_Q)$ and have zero off-diagonal elements in $\boldsymbol{\Sigma}(\boldsymbol{t})$, then a separable covariance function (4.1) is obtained.

Parametrisation of the varying anisotropy matrix

We must ensure positive definiteness of the anisotropy matrix $\boldsymbol{\Sigma}(\boldsymbol{t})$ in (4.6). This can be done by using different parametrisations. For example, Higdon (1998), Higdon *et al.* (1999), and Risser & Calder (2017) use geometrically-based parametrisations which capture local anisotropy by rotating and stretching coordinate directions. Paciorek & Schervish (2006) suggest using a spectral decomposition. However, these methods are either designed for some special cases or are difficult to form interpretation about its elements.

Pinheiro & Bates (1996) present five parametrisations for a covariance matrix, one of which is the spherical parametrisation, a particularly interesting strategy because it provides direct interpretation of parameters in terms of variances and correlations. We propose to use the spherical parametrisation for $\boldsymbol{\Sigma}(\boldsymbol{t})$ and interpret the parameters in terms of length-scale (to assess how rapidly varying is the function f of eq. (3.1) in each coordinate direction) and direction of dependence (to see potentially interaction between the coordinate directions).

As discussed above, the off-diagonal elements of $\boldsymbol{\Sigma}(\boldsymbol{t})$ have to be zero to produce a separable covariance function. Therefore, a value which is not zero indicates nonseparable covariance structure due to the interaction between the coordinate directions of $\boldsymbol{t}$ in the way the process fluctuates over $\boldsymbol{t}$.

We will consider the Cholesky decomposition

$$\boldsymbol{\Sigma}(\boldsymbol{t}) = \boldsymbol{\Sigma}(\boldsymbol{\tau}) = \boldsymbol{L}(\boldsymbol{\tau})^\top \boldsymbol{L}(\boldsymbol{\tau}),$$

where $\boldsymbol{L}$ is an $Q \times Q$ upper triangular matrix (including main diagonal). Positiveness of the main diagonal entries of $\boldsymbol{L}$ ensures that $\boldsymbol{\Sigma}$ is positive definite.

We will follow closely the exposition of Pinheiro & Bates (1996) to explain the spherical parametrisation. Let $\boldsymbol{L}_q$ denote the q-th column of $\boldsymbol{L}$ and ℓ_q denote the spherical

coordinates of the first q elements of $\boldsymbol{L}_q$. Therefore, we have

$$\begin{aligned}
[L_q]_1 &= [\ell_q]_1 \cos([\ell_q]_2), \\
[L_q]_2 &= [\ell_q]_1 \sin([\ell_q]_2) \cos([\ell_q]_3), \\
&\dots, \\
[L_q]_{q-1} &= [\ell_q]_1 \sin([\ell_q]_2) \cdots \cos([\ell_q]_q), \\
[L_q]_q &= [\ell_q]_1 \sin([\ell_q]_2) \cdots \sin([\ell_q]_q).
\end{aligned}$$

Let us define a diagonal matrix $\boldsymbol{C}$ whose diagonal entries are $[\boldsymbol{C}_{qq}] = [\boldsymbol{\Sigma}_{qq}]^{1/2}$. Then we can write $\boldsymbol{\Sigma}$ in terms of a matrix $\boldsymbol{R}$:

$$\boldsymbol{\Sigma} = \boldsymbol{C}^{1/2} \boldsymbol{R} \boldsymbol{C}^{1/2},$$

where $\boldsymbol{R}_{qq} = 1,\ q = 1, \dots, Q$, and $\boldsymbol{R}_{pq} = \rho_{pq}, p \neq q$. The parameter $\rho_{pq} \in (-1, 1)$ measures the direction of linear dependence between the coordinates p and q. If $\rho_{pq} \neq 0$ for some pair (p, q), then the covariance function (4.6) is nonseparable. The value of $\boldsymbol{\Sigma}_{qq}$ can be interpreted as the length-scale parameter and therefore measures how rapidly varying is the function f in (3.1) towards the coordinate q.

We can show that $\boldsymbol{\Sigma}_{qq} = [\ell_q]_1^2$ and that $\rho_{1q} = \cos([\ell_q]_2),\ q = 2, \dots, Q$, with $-1 < \rho_{1q} < 1$. This means that we can interpret the values of $\boldsymbol{L}$ in terms of the length-scale parameters and directions of dependence of $\boldsymbol{\Sigma}$.

The spherical parametrisation is unique if

$$\begin{aligned}
&[\ell_q]_1 > 0, \quad q = 1, \dots, Q, \\
&[\ell_q]_p \in (0, \pi), \quad q = 2, \dots, Q, \ \ p = 2, \dots, q.
\end{aligned}$$

We can then easily proceed with an unconstrained estimation by defining a new vector of parameters $\boldsymbol{\alpha}$ which includes $\log([\ell_q]_1),\ q = 1, \dots, Q$, and $\log\big([\ell_q]_p / (\pi - [\ell_q]_p)\big),\ q = 2, \dots, Q,\ \ p = 2, \dots, q$. Each element $\boldsymbol{\alpha}_j = \boldsymbol{\alpha}_j(\boldsymbol{\tau})$, for $j = 1, \dots, Q(Q+1)/2$, depends on $\boldsymbol{\tau}$ if the covariance structure is nonstationary.

The unconstrained estimation of each element allows it to be modelled as a nonparametric function of $\boldsymbol{\tau}$. The spherical parametrisation has some other advantages over other parametrisations in that: (i) it is uniquely defined and can be readily extended for any $Q > 2$, which is difficult when implementing geometrically-based parametrisations; (ii) it has about the same computational efficiency as the Cholesky parametrisation applied directly; (iii) we can make interpretation of the values of $\boldsymbol{L}$ in terms of the length-scale parameters and directions of dependence of $\boldsymbol{\Sigma}$; and (iv) we can account for uncertainty

on $\boldsymbol{\alpha}$, and consequently conduct inference on length-scale and direction of dependence.

A geometrical interpretation of the spherical parametrisation can be seen in Rapisarda *et al.* (2007). Other parametrisations based on Cholesky decomposition has been widely discussed. Zhang *et al.* (2015) mention that unconstrained nature of the parametrisation of the Cholesky factor allows to represent angles of the spherical parametrisation via regression as functions of some covariates, an idea also used by Pourahmadi (1999) and Leng *et al.* (2010) when parametrising covariance matrices using a modified Cholesky decomposition.

If the covariance structure depends along one coordinate direction $\tau \subset \mathbb{R}$ (i.e. $Q^* = 1$, e.g. time-varying parameters), many nonparametric methods can be used, e.g.

$$\alpha_j(\tau) = \sum_{l=1}^{L} \theta_{jl} B_{jl}(\tau), \tag{4.7}$$

where B_ℓ form B-spline basis functions (de Boor, 2001). This representation ensures that the resulting function is smooth and still very flexible as we can change the degree of the piecewise polynomials and the number and location of knots. The locations of the knots are usually the quantiles of τ, but they can be chosen differently; we can also allow for discontinuities in derivatives by repeating knots at the same location. The gain of adding more knots comes with the cost of increasing the number of parameters to be estimated. Typically, the number of knots is chosen by cross-validation.

For multidimensional $\boldsymbol{\tau} \in \mathbb{R}^{Q^*}$ (useful to model spatially-varying parameters), we can construct multivariate B-splines basis functions by taking the product of the Q^* univariate basis.

An alternative method is to use a Gaussian process to model each $\boldsymbol{\alpha}_j(\boldsymbol{\tau})$ using a parametric covariance function. Let $\boldsymbol{\alpha}_{ji} = \boldsymbol{\alpha}_j(\boldsymbol{\tau}_i),\ i = 1, \ldots, n$. Then we define

$$(\boldsymbol{\alpha}_{j1}, \ldots, \boldsymbol{\alpha}_{jn}) \sim N\big(\mathbf{0}, \boldsymbol{K}_j(\boldsymbol{\theta}_j)\big), \tag{4.8}$$

where $\boldsymbol{K}_j$ is an $n \times n$ covariance matrix where its (i, i')-th element is calculated by the covariance function $k_j(\boldsymbol{\tau}_i, \boldsymbol{\tau}_{i'}; \boldsymbol{\theta}_j)$, depending on unknown parameter $\boldsymbol{\theta}_j$. In practice, we may use the same covariance function for $j = 1, \ldots, Q$, and for $j = Q+1, \ldots, Q(Q+1)/2$. This method can cope with the large dimensional cases, i.e. $Q^* > 1$.

We now denote the covariance function constructed by (4.6) and the above parametrisation methods by $k(\boldsymbol{t}, \boldsymbol{t}'; \boldsymbol{\theta})$ for any $\boldsymbol{t}, \boldsymbol{t}' \in \mathbb{R}^Q$, where $\boldsymbol{\theta}$ includes all the unknown parameters in (4.7) if B-splines are used or the unknown parameters in (4.8) if GPRs are used; in addition, $\boldsymbol{\theta}$ includes $\log(\sigma^2)$ if we use (4.5). We will use an empirical Bayesian approach to estimate the unknown parameters and thus the nonstationary covariance structure.

Local empirical Bayes estimation of the covariance function

To reduce the computational costs when calculating the determinant and the inverse of $\boldsymbol{\Psi}_n$ in (3.4), we can instead using the local likelihood estimation (LLE) (Tibshirani & Hastie, 1987). In the LLE, instead of maximising (3.4) directly, we maximise

$$\mathcal{L}_k(\boldsymbol{\theta}_k|\mathcal{D}_k) = -\frac{1}{2}\log|\boldsymbol{\Psi}(\boldsymbol{\theta}_k)| - \frac{1}{2}\boldsymbol{x}_k'\boldsymbol{\Psi}(\boldsymbol{\theta}_k)^{-1}\boldsymbol{x}_k - \frac{n_k}{2}\log 2\pi$$

locally, where k is the index of location $\boldsymbol{t}_k$. Estimates of $\boldsymbol{\theta}_k$ are obtained by considering only the data in the neighbourhood of $\boldsymbol{t}_k$, that is, $\mathcal{D}_k = \big\{(\boldsymbol{x}_i, \boldsymbol{t}_i) : \{||\boldsymbol{t}_i - \boldsymbol{t}_k|| < r\}\big\}$, where r is a predefined radius. Using the available observations in the neighbourhood of $\boldsymbol{t}_k$ is important as the behaviour of the covariance function near the origin determines properties of the process (Stein, 1999). Risser & Calder (2017) suggest a mixture component approach in which they estimate the spatially varying parameters $\boldsymbol{\theta}_k,\ k = 1, \ldots, k_{max}$, locally and then, for any arbitrary location $\boldsymbol{t}$, $\boldsymbol{\theta}_k(\boldsymbol{t})$ is obtained by averaging, respectively, $\boldsymbol{\theta}_k,\ k = 1, \ldots, k_{max}$, with a weight function depending on the distance between $\boldsymbol{t}_k$ and $\boldsymbol{t}$.

A special case is when the nonstationarity depends on one coordinate direction as discussed around equation (4.7). We can use B-spline basis functions and then estimate the corresponding coefficients θ_{jl}. In practice, we may simply estimate $\boldsymbol{\alpha}_j$ locally for some locations via LLE (i.e. assuming $\boldsymbol{\alpha}_j$ is a constant in a neighbourhood of the locations) and then regress these estimates to obtain smooth functions $\boldsymbol{\alpha}_j(\boldsymbol{\tau})$ for any $\boldsymbol{\tau} \in \mathbb{R}^{Q^*}$, using a nonparametric approach, e.g. B-splines.

When we use all the data (not using local likelihood estimation (LLE)), this of course requires a potentially high computational cost as we consider all the data rather than only the data within a neighbourhood. However, if computational costs are not prohibitive, this approach should be preferable to the local likelihood approach, whose performance heavily depends on the neighbourhood size r. In the LLE, if a small neighbourhood is used (e.g. in order to model very local features), one might obtain unstable local estimates. On the other hand, if a large neighbourhood is used (something necessary when data are sparse), then the local stationarity assumption is no longer appropriate and the local estimates might be very biased.

4.3 Simulation studies

In this section, we show three examples of data with nonseparable, nonstationary covariance structure. We discuss how directions of linear dependence between coordinates can be visualised and show that nonseparable models can explain more variation in the data

using less components than separable models do.

4.3.1 Simulation study 1

We simulate 30 realisations from a zero mean, two-dimensional function-valued process $X(\boldsymbol{t})$, $\boldsymbol{t} = (\tau, s)^\top$, observed at $n_1 \times n_2 = 25 \times 25 = 625$ equally spaced points on $[0,1]^2$. We assume a measurement error with variance $\sigma^2_\varepsilon = 0.1$ and the covariance function (4.6), with varying overall variance $\sigma^2(\tau) = \exp(\tau)$ and varying anisotropy matrix $\boldsymbol{\Sigma}(\tau) = \boldsymbol{C}^{1/2}\boldsymbol{R}(\tau)\boldsymbol{C}^{1/2}$, where $\boldsymbol{C} = \text{diag}(0.1, 0.1)$, $\boldsymbol{R}_{11}(\tau) = \boldsymbol{R}_{22}(\tau) = 1$ and different specifications for $\boldsymbol{R}_{12}(\tau) = \boldsymbol{R}_{21}(\tau)$: we set $\boldsymbol{R}_{12}(\tau) = 0$ to produce a separable covariance function and $\boldsymbol{R}_{12}(\tau) = 0.95\tau$ and $\boldsymbol{R}_{12}(\tau) = 0.8$ to produce nonseparable ones. Four different specifications for $g(\cdot)$ were used for generating the data: Matérn with $\nu = 1/2$, $\nu = 3/2$, $\nu = 5/2$, and $\nu = 5$.

Then we estimate the covariance structure by the NSGP model with two specifications for $g(\cdot)$ (squared exponential and exponential) and by the Product FPCA model. In the NSGP model, the τ-varying parameters are estimated via B-splines with 3 knots.

Figure 4.1 and 4.2 show the CFVEs obtained when the data generating process is a GP and a T-process (with 6 degrees of freedom), respectively. The conclusions made for both figures are very similar, showing that the proposed NSGP approach is robust to heavy-tailed data.

The top panel of the Figures 4.1 and 4.2 show the separable case; the middle panel shows the case where the nonseparable feature is only strong when τ is large; and the bottom panel show the case where the nonseparable feature is strong for any τ. Note that the value of $\boldsymbol{R}_{12}(\tau)$ measures how distant is the covariance structure from the separability assumption. The figures indicate very similar performance of the three methods when the covariance is separable. However, when the nonseparable feature is strong, NSGP models obtain better (or at least very similar) CFVEs than Product FPCA does. In other words, when the covariance structure becomes more distant from the separability assumption, then Product FPCA model seems to not capture important information about the variation in the data which are explained by nonseparable eigensurfaces. Observe, however, that the first eigensurface estimated by each method have very similar contribution, even in the strong nonseparability cases. This, of course, is because nonseparable features are caused by interaction between the coordinate directions, and therefore we do expect to see a clearer advantage of nonseparable models only in later eigensurfaces, not in the first one.

Whereas Matérn with $\nu = 1/2$ is equivalent to the exponential kernel, Matérn with $\nu = \infty$ converges to the squared exponential kernel. This explains why the exponential

correlation kernel is preferable for data generated with small ν and the squared exponential kernel provides better results for large ν.

The larger the value of ν the smoother are the random functions. Therefore, as we can see in the Figures 4.1 and 4.2, the larger the value of ν, the smaller is the number of components necessary to achieve a high CFVE. Each row of figures were not plotted using the same scale in order to visualise better the difference between the methods.

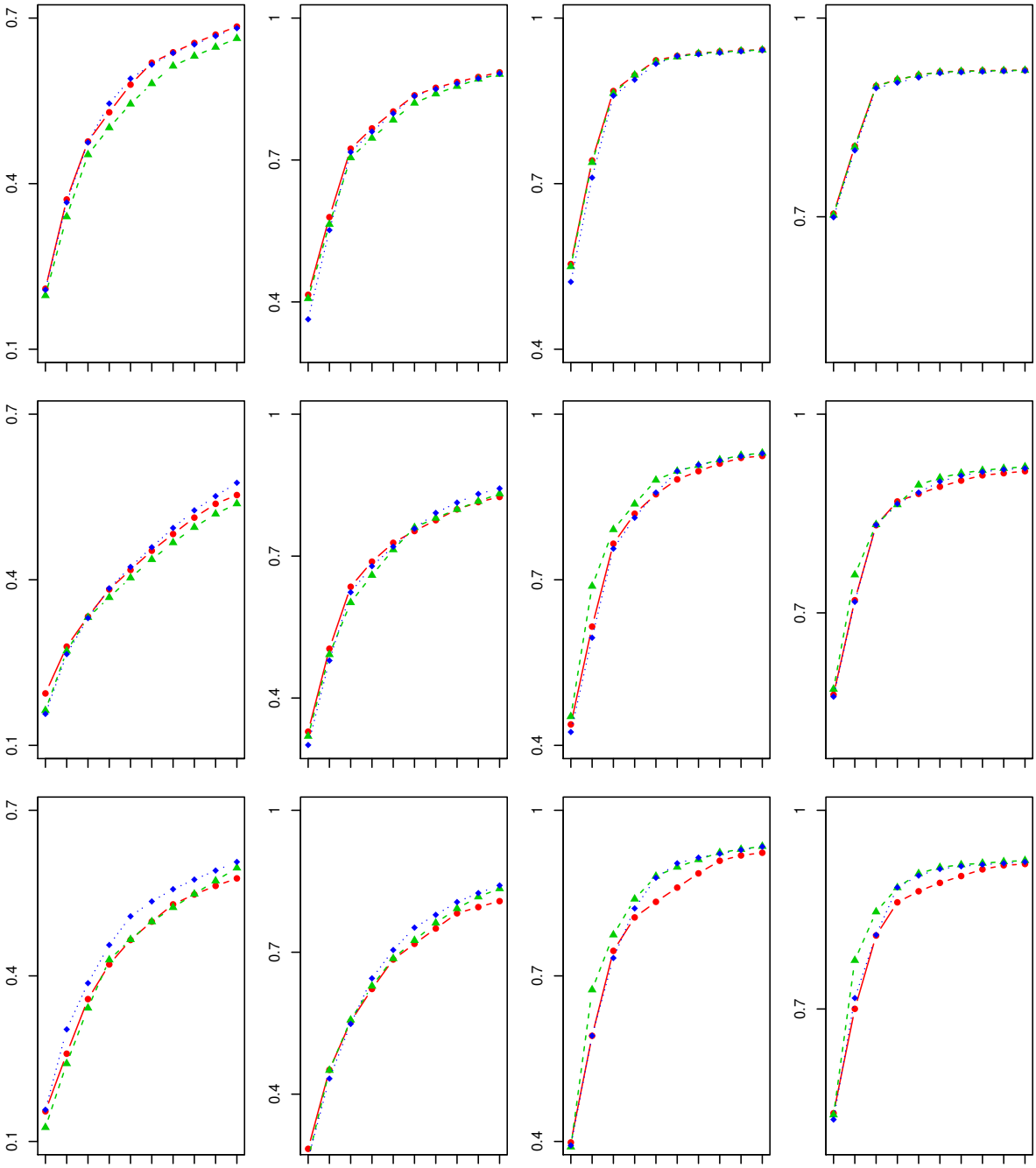

Figure 4.1: CFVEs of GP data for $J = 1, \ldots, 10$, obtained by Product FPCA (red), NSGP with squared exponential $g(\cdot)$ (green), and NSGP with exponential $g(\cdot)$ (blue). In each column, from the top to the bottom, $\boldsymbol{R}_{12}(\tau) = 0$, $\boldsymbol{R}_{12}(\tau) = 0.95\tau$, $\boldsymbol{R}_{12}(\tau) = 0.8$. In each row, from the left to the right, the data generating process follows a GP where $g(\cdot)$ is Matérn with $\nu = 1/2$, $\nu = 3/2$, $\nu = 5/2$, and $\nu = 5$.

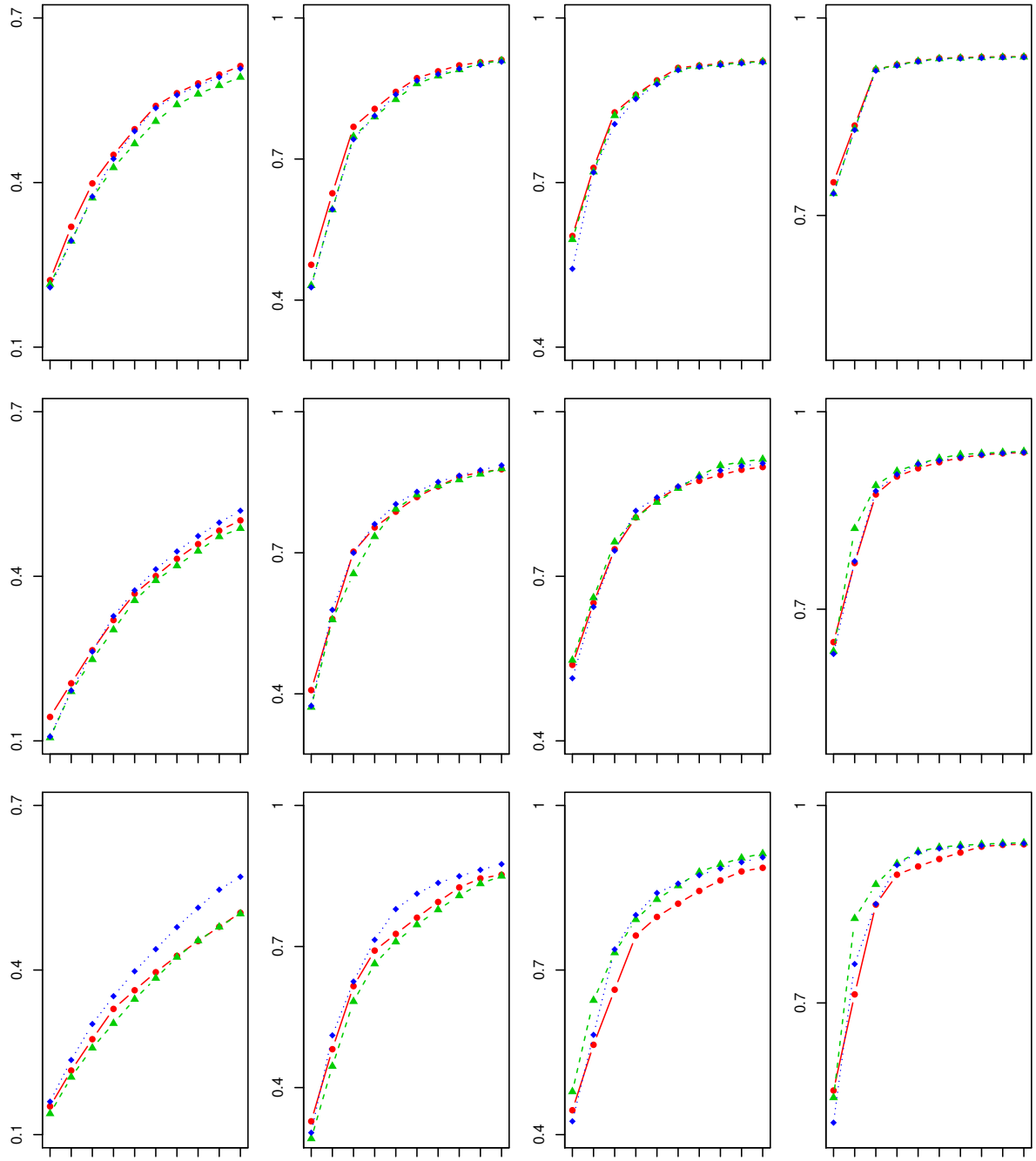

Figure 4.2: CFVEs of T-process data for $J = 1, \ldots, 10$, obtained by Product FPCA (red), NSGP with squared exponential $g(\cdot)$ (green), and NSGP with exponential $g(\cdot)$ (blue). In each column, from the top to the bottom, $\boldsymbol{R}_{12}(\tau) = 0$, $\boldsymbol{R}_{12}(\tau) = 0.95\tau$, $\boldsymbol{R}_{12}(\tau) = 0.8$. In each row, from the left to the right, the data generating process follows a T-process where $g(\cdot)$ is Matérn with $\nu = 1/2$, $\nu = 3/2$, $\nu = 5/2$, and $\nu = 5$.

4.3.2 Simulation study 2

In this study, we simulate a three-dimensional function-valued process $X(\tau, s_1, s_2)$, where $\tau, s_1, s_2 \in [0, 1]$, from (3.1), where f is zero-mean T-process with covariance function (4.6) and squared exponential correlation kernel $g(\cdot)$. We also set $\sigma_\varepsilon^2 = 0.1$. We assume that the parameters σ^2 and $\mathbf{\Sigma}$ in (4.6) depend only on τ. This example is comparable to spatiotemporal models which have time and spatial coordinates as input variables and time-varying coefficients (e.g. dynamic linear models (Banerjee *et al.*, 2015)).

We set $\sigma^2(\tau) = \exp(\tau)$ and

$$\mathbf{\Sigma}(\tau) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \rho_{23}(\tau) \\ 0 & \rho_{23}(\tau) & 1 \end{bmatrix},$$

with $\rho_{23}(\tau) = 0.95\tau$. That is, the process f has overall variance $\sigma^2(\cdot)$ and the interaction between coordinate directions s_1 and s_2, given by $\rho_{23}(\cdot)$, increases over τ. This produces a nonstationary, nonseparable covariance function.

As τ increases, ρ_{23} becomes more distant from zero, and thus models which assume separable covariance structure might be not suitable. Figs 4.3 and 4.4, show, for $\tau = 0.25$ and $\tau = 1$, respectively, the true leading eigensurfaces $\phi(\tau, s_1, s_2)$ and the corresponding estimates obtained by nonstationary GP (NSGP) and Product FPCA.

The first eigensurface represents the direction of the largest variation in the data relative to the mean function. The second eigensurface corresponds to the direction of the largest variation which is orthogonal to the first eigensurface, and so forth. Covariance separability implies separability of eigensurfaces. Therefore, if we observe nonseparable eigensurfaces, this means that the separability assumption is not satisfied.

Although the data were simulated from a T-process, the NSGP model obtains estimated eigensurfaces quite similar to the true ones. The model clearly identifies that the interaction between s_1 and s_2 becomes stronger for higher values of τ – see how the diagonal orientation is clearer as τ increases from Fig. 4.3 to Fig. 4.4. This fact is not detected by the Product FPCA model as this assumes separability of the eigensurfaces.

In practice, one eigensurface can often be interpreted as a *size* component; if the corresponding countour plot is ellipsoidal with diagonal orientation, then a model which assumes separability will not describe it well. The visualisation of eigensurfaces can also identify, for example, a contrast between low-latitude and high-latitude; this contrast, however, might not be constant over longitude. If we allow for nonseparability, we can identify, for example, a contrast between between northwest and southeast (see e.g. the

second true eigensurface in Fig. 4.4, assuming that s_1 and s_2 represent longitude and latitude, respectively).

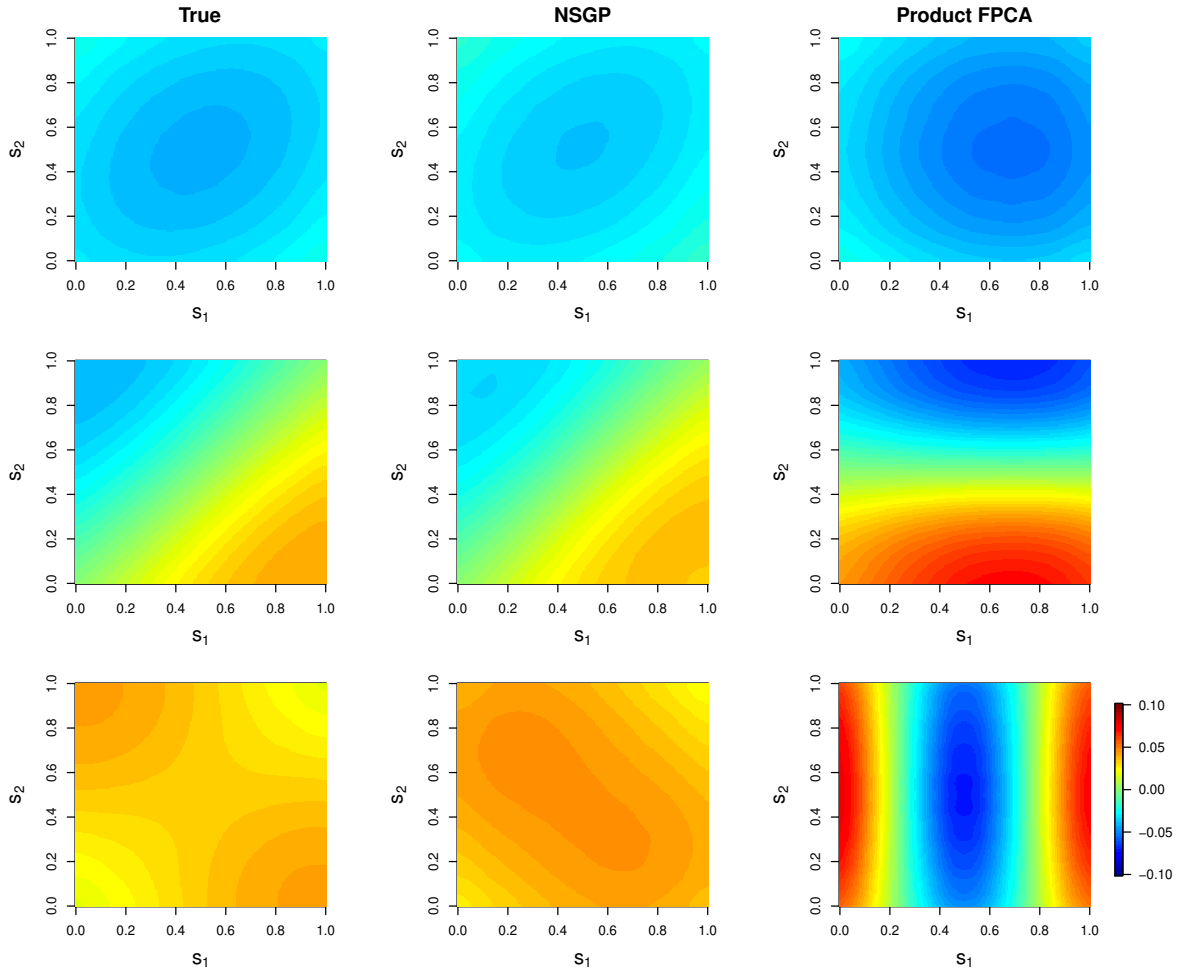

Figure 4.3: First four leading eigensurfaces $\phi(0.25, s_1, s_2)$ of the true model (left column) and the corresponding estimated eigensurfaces $\hat{\phi}(0.25, s_1, s_2)$ from the NSGP model (centre) and Product FPCA model (right).

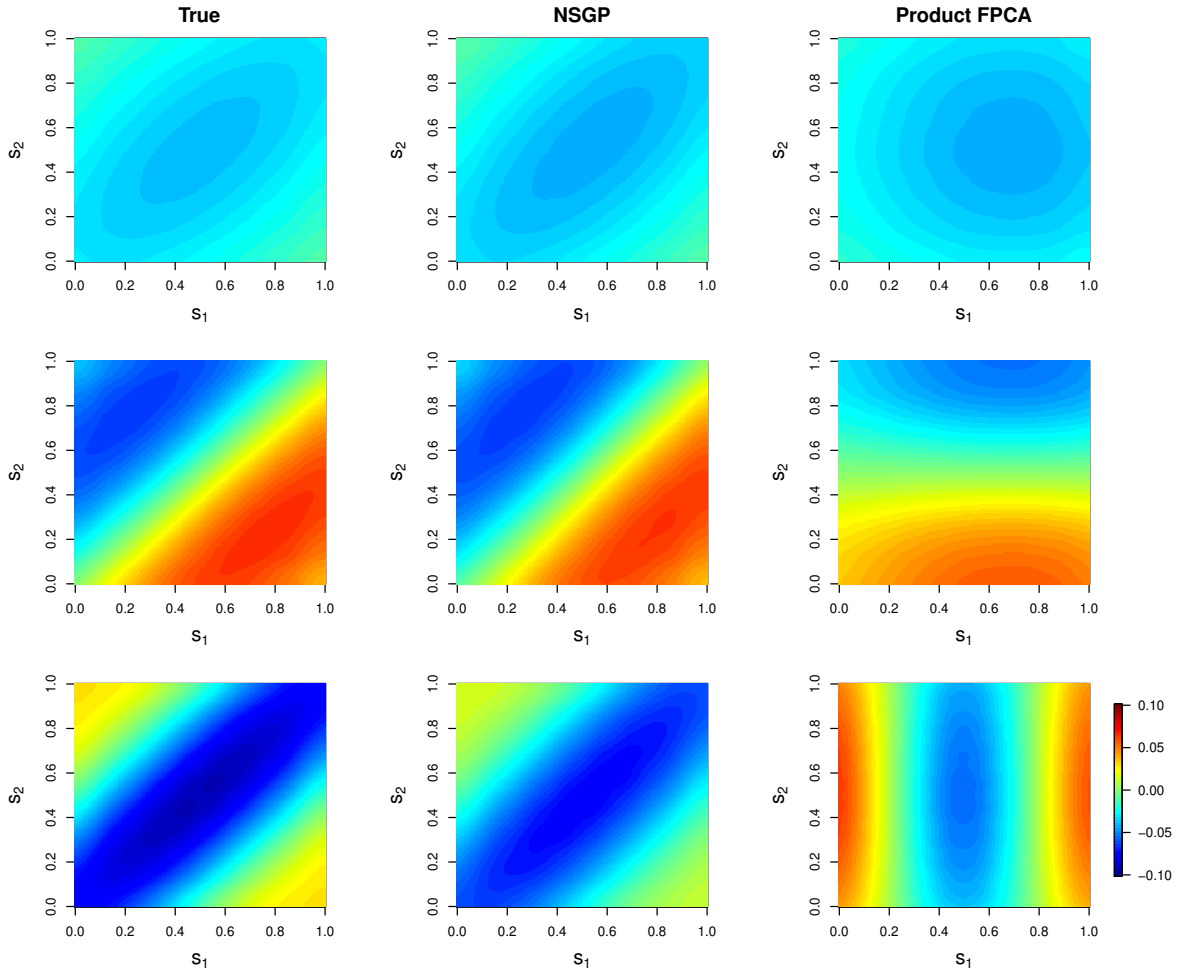

Figure 4.4: First four leading eigensurfaces $\phi(1, s_1, s_2)$ of the true model (left column) and the corresponding estimated eigensurfaces $\hat{\phi}(1, s_1, s_2)$ from the NSGP model (centre) and Product FPCA model (right).

The CFVEs of the first 16 leading three-dimensional eigensurfaces are illustrated in Fig. 4.5. As expected, the advantage of the nonseparable model in terms of CFVE is clear not in the first, but in later components. In addition, we can conclude that the NSGP model requires less components to explain the same amount of variation in this dataset.

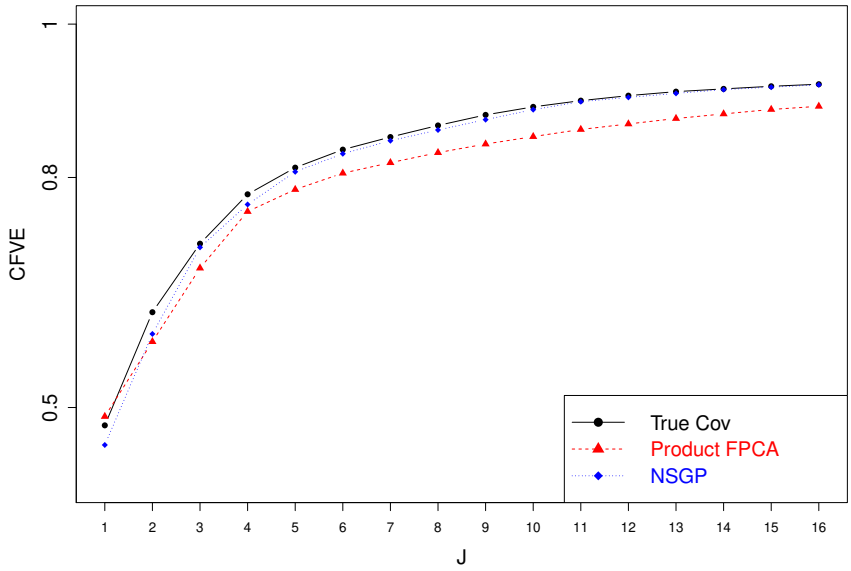

Figure 4.5: Comparison of cumulative FVEs obtained by the true, and Product FPCA, and NSGP models.

4.3.3 Simulation study 3

In this simulation study, we assume that the two-dimensional function-valued process $X(s_1, s_2)$ has zero mean and covariance function given by

$$\text{Cov}\left[X(\boldsymbol{s}), X(\boldsymbol{s}')\right] = \sum_{j=1}^{20} \alpha_j \phi_j(s_1 + s_2)\phi_j(s_1' + s_2'),$$

where $\phi_j(\cdot)$ are Chebyshev polynomials, $\alpha_j = j^{-3/2}$ and $\boldsymbol{s} \in [-1, 1]^2$. The basis functions of the form $\phi_j(s_1 + s_2)$ are clearly nonseparable and produce a nonseparable covariance structure.

We generate 100 surfaces observed at $n_1 \times n_2 = 20 \times 20 = 400$ equally spaced points and estimate the covariance structure by NSGP and Product FPCA models. Figure 4.6 illustrates that the NSGP model obtains more accurate estimates of the leading eigensurfaces. The diagonal shapes of the true leading eigenfunctions indicate strong nonseparability features in the covariance function, features which are not captured by the Product FPCA model. The eigensurfaces have such diagonal shapes because they are polynomials of the sum $s_1 + s_2$, and later eigensurfaces change faster along the input domain because they are polynomials of higher order.

Finally, Figure 4.7 shows that, when using more than two components, NSGP is the preferable model in terms of explaining better the main modes of variation in the data.

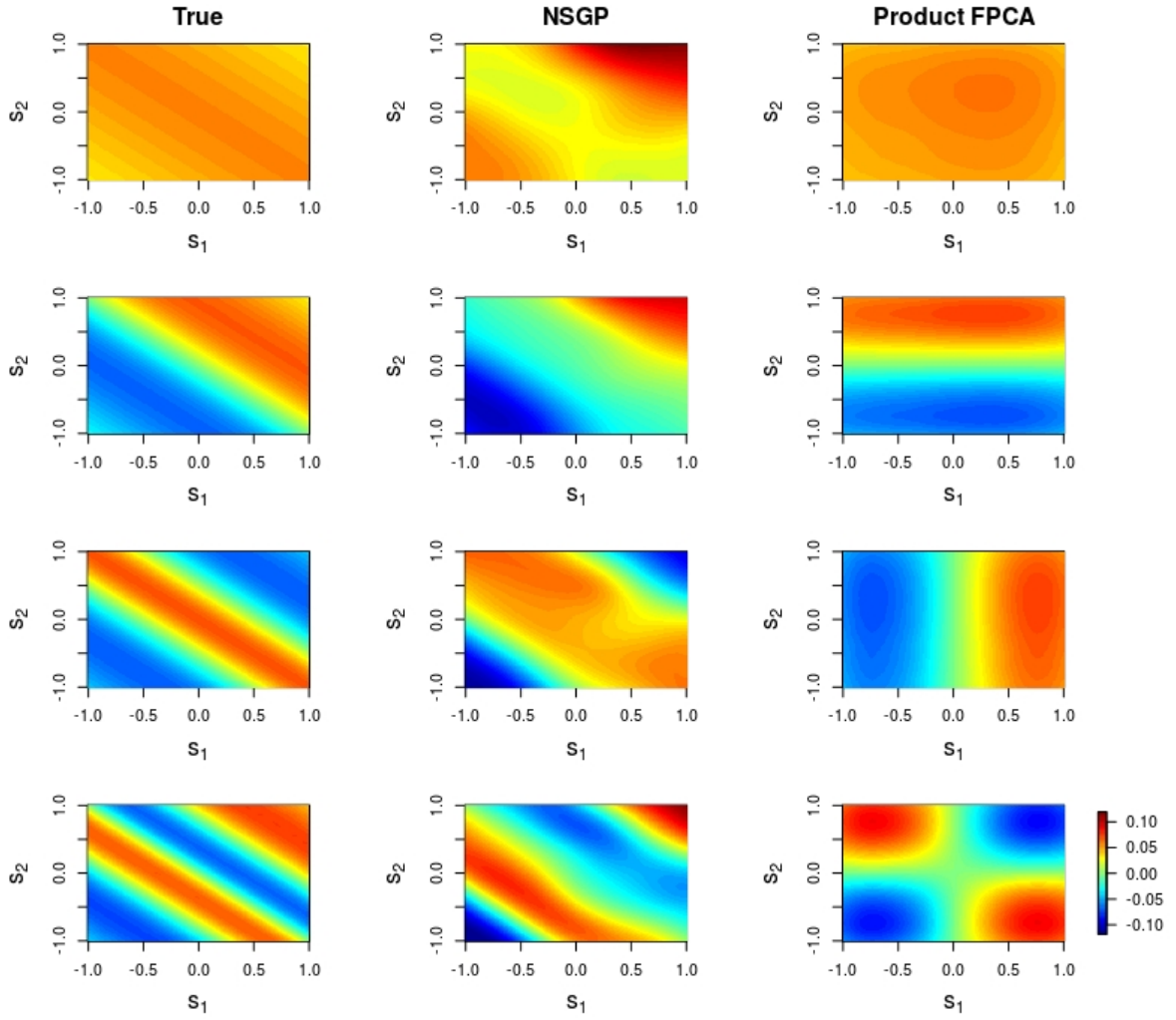

Figure 4.6: First four leading eigensurfaces $\phi(s_1, s_2)$ of the true model (left column) and the corresponding estimated eigensurfaces $\hat{\phi}(s_1, s_2)$ from the NSGP model (centre) and Product FPCA model (right). Chebyshev polynomials data.

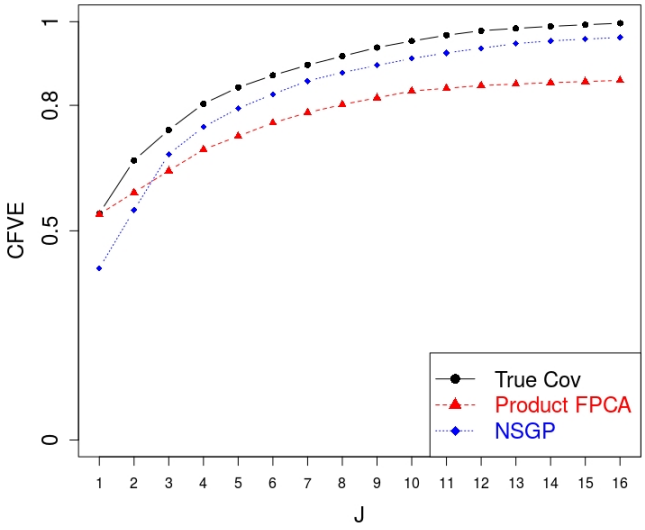

Figure 4.7: Comparison of CFVEs obtained by the true, and Product FPCA, and NSGP models. Chebyshev polynomials data.

4.4 Application to Canadian temperature data

In this application, we model the daily mean temperature of 36 stations in Canada[1] observed in all days from 01-01-1998 to 31-12-2005. The data corresponding to a year is assumed to be a realisation of a random process, so that we have eight realisations.

We use equation (3.1) to model the daily mean temperature X, where $f(\cdot)$ follows a GP with covariance function (4.6) and squared exponential correlation function $g(\cdot)$. The coordinate directions are $\boldsymbol{t} = (\tau, s_1, s_2)^\top$, corresponding to time, latitude and longitude, respectively. For each realisation (year), the mean function $\mu(\tau)$ was taken to be the sample mean across the 36 stations. The parameters $\boldsymbol{\theta}$ are assumed to be time-varying and we use a B-spline basis system with six regularly spaced knots to model them.

The estimate of the overall standard deviation $\sigma(\tau)$ can be seen in Fig. 4.10, indicating that in winter months there is a higher variation in the mean temperature data relative to the mean function across the stations. This can be observed in the mean temperature data (see Fig. 4.9).

The estimates for the elements of the varying matrix $\boldsymbol{\Sigma}(\tau)$ are shown in Fig. 4.11. The change of parameter values over time indicate that the fluctuation of the process over each direction is different for different times of the year, something that stationary

[1]The dataset can be obtained by using the R package `weathercan` (LaZerte & Albers, 2018), which was used to download the data from the Environment and Climate Change Canada (ECCC) website `<http://climate.weather.gc.ca/historical_data/search_historic_data_e.html>`

models cannot capture. Moreover, the departure from 0 of the directions of dependence $\rho_{ij}, i \neq j$, reveals the presence of interaction between the coordinate directions, showing nonseparable features in the covariance structure.

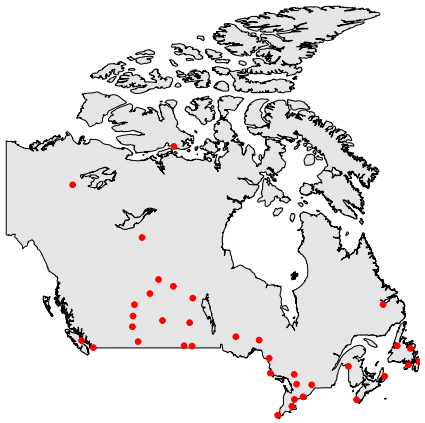

Figure 4.8: Map of Canada and the location of 36 stations where the data were observed.

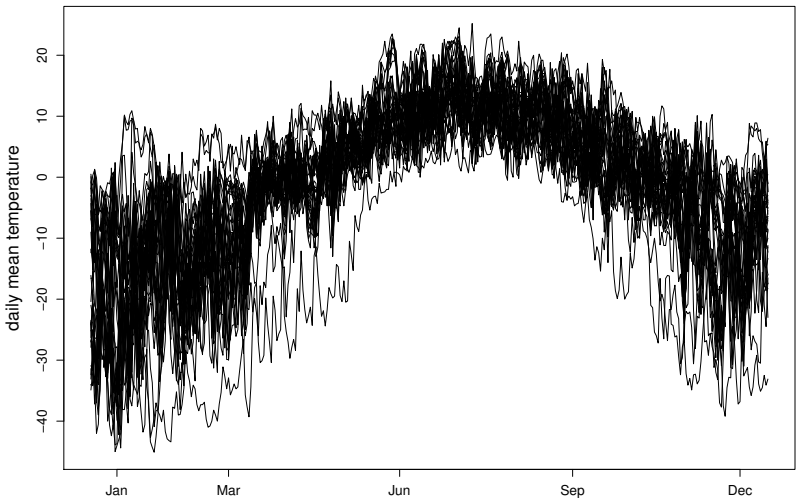

Figure 4.9: Daily mean temperature of 36 canadian stations in 2005.

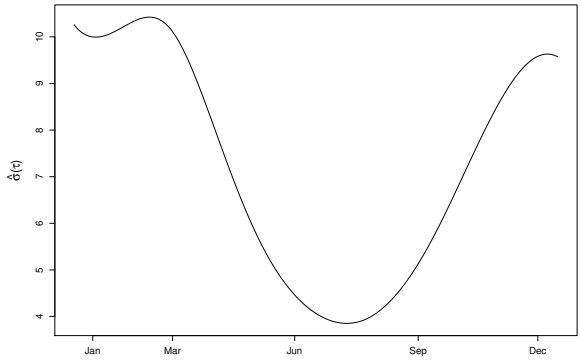

Figure 4.10: Estimate of $\sigma(\tau)$.

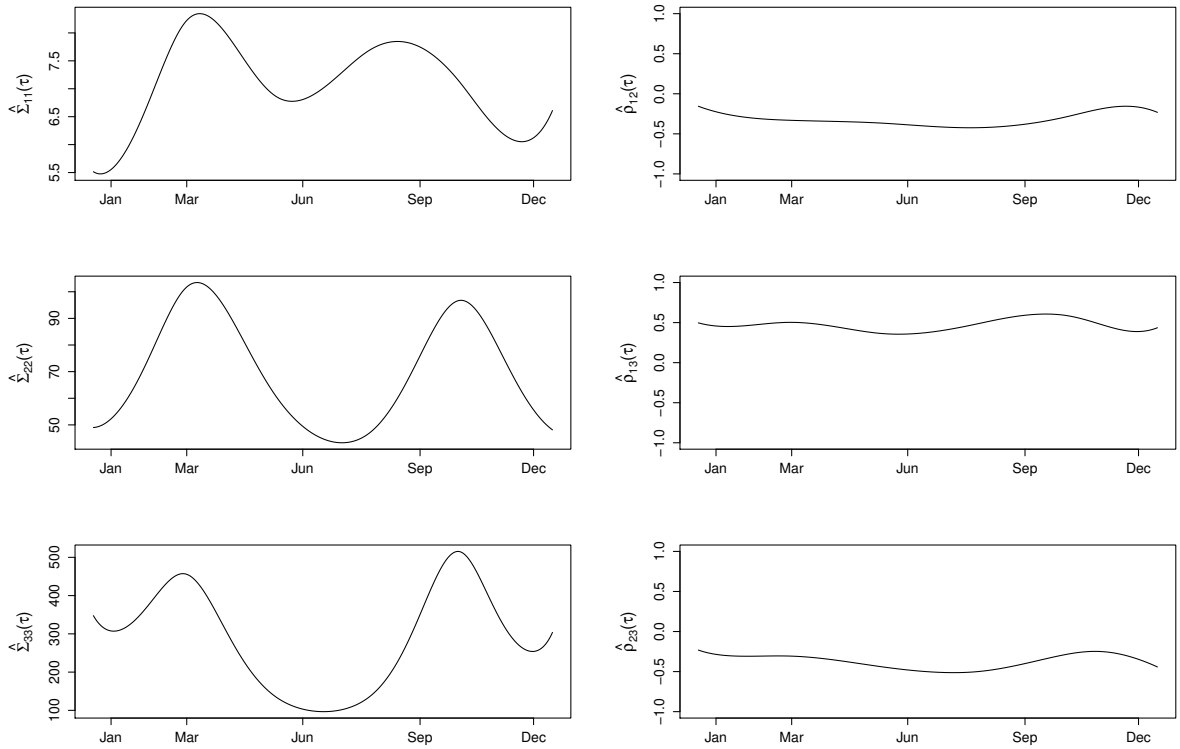

Figure 4.11: Estimates of diagonal elements of varying matrix $\mathbf{\Sigma}$ (left) and of directions of dependence $\rho_{pq} = \mathbf{\Sigma}_{pq}/\sqrt{\mathbf{\Sigma}_{pp}\mathbf{\Sigma}_{qq}}$ (right).

4.5 Implementation

In this section, we discuss a variety of implementation details to speed up computation and describe the auxiliary functions of existing R packages.

In the simulation studies, we employed the R package mvtnorm (Genz *et al.*, 2019) to simulate data from GPs and T-processes using its random number generator for the multivariate normal distribution and multivariate t distribution, respectively. In all simulation studies of this chapter, B-splines basis functions were obtained using the R package splines. In the simulation study of subsection 4.3.3, the Chebyshev polynomials were computed by using the R package orthopolynom (Novomestky, 2013).

When we implemented local likelihood estimation, we parallelised it over multiple cores, substantially reducing the computational time.

We have used the R function nlminb for unconstrained and constrained optimisation of the marginal likelihood functions. However, whenever it was possible, we converted the constrained estimation problems to the unconstrained estimation case. For example, instead of imposing the positivity constraint for σ^2, we replaced σ^2 by $\exp(\log \sigma^2)$ in the function to be optimised and took the unconstrained value $\log \sigma^2$ as the input of the function.

Notice that in our simulation studies the replicated surfaces are observed in the same locations. This means we need not calculate different covariance matrices when evaluating the likelihood function. The covariance matrix is the same for all replicated curves/surfaces and has to be calculated just once. Although replications in general are not observed in the same locations, this is the case for some types of data (e.g. fertility data).

We are often interested in the very first few eigenvalues and eigenvectors of a covariance matrix. Therefore, we can use the R package rARPACK (Qiu *et al.*, 2016) to obtain the only first few elements instead of performing the eigendecomposition which may be time consuming as it has complexity of $O(n^3)$. Consequently, CFVE results were not obtained by (2.6). Instead, we regressed the data on each eigensurface and calculated the R^2 to see the proportion of variability in the data which is explained by that eigensurface.

To plot the eigensurfaces, we have used surface approximation from bivariate scattered data using multilevel B-splines. This was implemented using the R package MBA (Finley *et al.*, 2017). The package fields (Nychka *et al.*, 2017) was used to divide up the graphics window into a matrix of plots and to produce a legend strip for the figures of eigensurfaces.

4.6 Conclusion

Our Bayesian modelling framework can handle function-valued processes defined on multi-dimensional domains without assuming covariance separability. Moreover, the covariance structure is very flexible, where the nonstationarity is achieved by allowing its parameters to vary along the input domain. Our proposed spherical parametrisation for the varying anisotropy matrix enables us to have unconstrained, interpretable parameters. This is important for many applications. For example, if random trajectories fluctuate over time more quickly in the winter than in other seasons (see, for example, the application to Canadian temperature data), then time (or a time-dependent covariate) seems a natural input for the corresponding length-scale parameter. In addition, whereas the visualisation of separable eigensurfaces can identify, for example, a contrast between low-latitude and high-latitude, the visualisation of nonseparable eigensurfaces can be useful to detect a contrast between between northwest and southeast.

Due to the unconstrained parametrisation, we can model the parameters using a non-parametric model, and this flexibility is crucial for the covariance function fitting when the data have complex covariance structure. The popular parametric families of covariance functions in spatial statistics literature do not have such flexibility as they are usually stationary or separable.

Based on the decomposition of GPs in Chapter 3, the leading eigensurfaces of the estimated covariance structure can be used to extract some of the most important information of the variation in the data. Unlike models which assume covariance separability, the simulation studies have shown that interactions between coordinate directions can be identified in the covariance structure, leading to new interpretation of the data. In addition, unlike stationary parametric models, the varying parameters of the semiparametric approach can identify how some features vary along the input domain.

Chapter 5

Multivariate Gaussian Processes

In this chapter, we consider the modelling of multivariate function-valued processes $\boldsymbol{X}(\cdot) = \big(X_1(\cdot), \ldots, X_M(\cdot)\big)^\top$. The main difficulty is to define cross-covariance functions such that the full covariance function of the multivariate functional response is positive definite. A popular approach is to assume that each of the M random functions has the same covariance function $k(\cdot, \cdot)$ and that the full covariance is given by $\boldsymbol{\Psi} = \boldsymbol{P} \otimes k(\cdot, \cdot)$, where $\boldsymbol{P}$ is a $M \times M$ matrix describing the dependence between the outputs.

This strong assumption has lead us to consider a bivariate convolved GP model (Boyle & Frean, 2004) and extend it to the M-variate case, a model that we call the multivariate Gaussian process (MGP) model. In this approach, each random function $X_l,\ l = 1, \ldots, M$, is modelled as a sum of an independent latent process and another latent process common to all M random functions.

In Section 5.1, we present our proposed extension to the multivariate case and illustrate their better prediction performance when compared to the independent GP model for each random function. The section also gives two simulation studies which show that MGP can provide similar or better results than MFPCA (Happ & Greven, 2018) does for multivariate functional data. In Section 5.2, an application to human fertility data shows new insights provided by the visualisation of the leading eigensurfaces and prediction results of MGP model.

5.1 Multivariate Gaussian process model

Boyle & Frean (2004) suggest a model which includes two dependent outputs by considering four GPs constructed via convolution. Each GP constructed in this way is called convolved GP (CGP). Inspired by that model, in this section we propose an extension to the case involving M outputs.

For illustration purposes, let us see how cross-covariances can be constructed by defining trivariate dependent GPs. Let γ_0, γ_1, γ_2, and γ_3 be independent Gaussian white noise processes and h_{10}, h_{20}, h_{30}, h_{11}, h_{21}, and h_{31} be smoothing kernels. Analogously to eq. (4.3), we can construct a convolved GP from $\xi(\boldsymbol{t}) = \int h(\boldsymbol{t}-\boldsymbol{u})\gamma(\boldsymbol{u})d\boldsymbol{u}$ and denote it as $\xi(\boldsymbol{t}) \sim CGP\big(h(\boldsymbol{t}), \gamma(\boldsymbol{t})\big)$.

Consider that

$$\begin{aligned}
&\xi_1(\boldsymbol{t}) \sim CGP\big(h_{10}(\boldsymbol{t}), \gamma_0(\boldsymbol{t})\big), \quad \eta_1(\boldsymbol{t}) \sim CGP\big(h_{11}(\boldsymbol{t}), \gamma_1(\boldsymbol{t})\big),\\
&\xi_2(\boldsymbol{t}) \sim CGP\big(h_{20}(\boldsymbol{t}), \gamma_0(\boldsymbol{t})\big), \quad \eta_2(\boldsymbol{t}) \sim CGP\big(h_{21}(\boldsymbol{t}), \gamma_2(\boldsymbol{t})\big),\\
&\xi_3(\boldsymbol{t}) \sim CGP\big(h_{30}(\boldsymbol{t}), \gamma_0(\boldsymbol{t})\big), \quad \eta_3(\boldsymbol{t}) \sim CGP\big(h_{31}(\boldsymbol{t}), \gamma_3(\boldsymbol{t})\big).
\end{aligned} \tag{5.1}$$

Then we can define trivariate dependent GPs as follows:

$$\begin{aligned}
X_1(\boldsymbol{t}) &= \eta_1(\boldsymbol{t}) + \xi_1(\boldsymbol{t}) + \text{noise}_1,\\
X_2(\boldsymbol{t}) &= \eta_2(\boldsymbol{t}) + \xi_2(\boldsymbol{t}) + \text{noise}_2,\\
X_3(\boldsymbol{t}) &= \eta_3(\boldsymbol{t}) + \xi_3(\boldsymbol{t}) + \text{noise}_3.
\end{aligned}$$

In this way, we can see that ξ's are dependent because they are affected by γ_0. However, η's are independent. Therefore, the dependence between the functions X's is defined through ξ's, whereas individual characteristics are modelled by η's. Figure 5.1 illustrates this example.

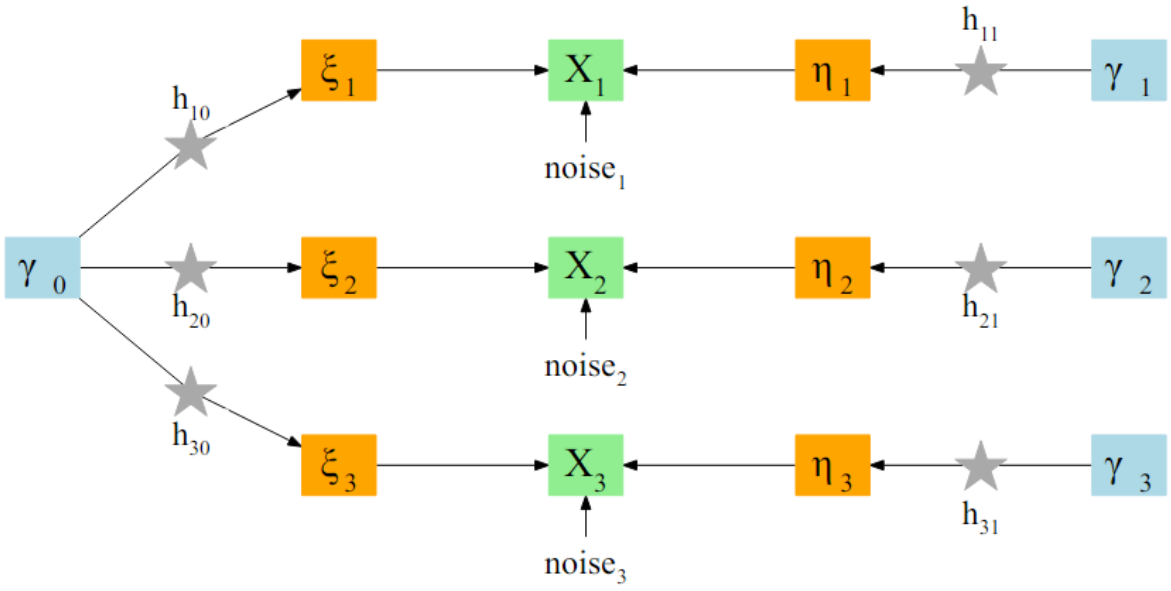

Figure 5.1: Scheme to construct MGP with three dependent outputs: X_1, X_2, and X_3.

Suppose the smoothing kernel functions in (5.1) are given by

$$
\begin{aligned}
h_{10}(\boldsymbol{t}) &= \nu_{10} \exp\Big\{ -\frac{1}{2}\boldsymbol{t}^\top A_{10}\boldsymbol{t}\Big\}, \\
h_{a0}(\boldsymbol{t}) &= \nu_{a0} \exp\Big\{ -\frac{1}{2}(\boldsymbol{t}-\boldsymbol{\mu}_a)^\top A_{a0}(\boldsymbol{t}-\boldsymbol{\mu}_a)\Big\}, \qquad a = 2,3, \\
h_{a1}(\boldsymbol{t}) &= \nu_{a1} \exp\Big\{ -\frac{1}{2}\boldsymbol{t}^\top A_{a1}\boldsymbol{t}\Big\}, \qquad a = 1,2,3.
\end{aligned}
$$

Therefore, $\big(X_1(\boldsymbol{t}), X_2(\boldsymbol{t}), X_3(\boldsymbol{t})\big)^\top$ defined in (5.1) defines a trivariate Gaussian process regression model with zero means and covariance function given by

$$
\begin{aligned}
k(\boldsymbol{t}_i, \boldsymbol{t}_j) &= \begin{bmatrix}
\mathrm{Cov}\big[X_1(\boldsymbol{t}_i), X_1(\boldsymbol{t}_j)\big] & \mathrm{Cov}\big[X_1(\boldsymbol{t}_i), X_2(\boldsymbol{t}_j)\big] & \mathrm{Cov}\big[X_1(\boldsymbol{t}_i), X_3(\boldsymbol{t}_j)\big] \\
\mathrm{Cov}\big[X_2(\boldsymbol{t}_i), X_1(\boldsymbol{t}_j)\big] & \mathrm{Cov}\big[X_2(\boldsymbol{t}_i), X_2(\boldsymbol{t}_j)\big] & \mathrm{Cov}\big[X_2(\boldsymbol{t}_i), X_3(\boldsymbol{t}_j)\big] \\
\mathrm{Cov}\big[X_3(\boldsymbol{t}_i), X_1(\boldsymbol{t}_j)\big] & \mathrm{Cov}\big[X_3(\boldsymbol{t}_i), X_2(\boldsymbol{t}_j)\big] & \mathrm{Cov}\big[X_3(\boldsymbol{t}_i), X_3(\boldsymbol{t}_j)\big]
\end{bmatrix} \\
&= \begin{bmatrix}
k_{11}(\boldsymbol{t}_i, \boldsymbol{t}_j) & k_{12}(\boldsymbol{t}_i, \boldsymbol{t}_j) & k_{13}(\boldsymbol{t}_i, \boldsymbol{t}_j) \\
k_{21}(\boldsymbol{t}_i, \boldsymbol{t}_j) & k_{22}(\boldsymbol{t}_i, \boldsymbol{t}_j) & k_{23}(\boldsymbol{t}_i, \boldsymbol{t}_j) \\
k_{31}(\boldsymbol{t}_i, \boldsymbol{t}_j) & k_{32}(\boldsymbol{t}_i, \boldsymbol{t}_j) & k_{33}(\boldsymbol{t}_i, \boldsymbol{t}_j)
\end{bmatrix},
\end{aligned}
$$

where

$$k_{aa}(\boldsymbol{t}_i, \boldsymbol{t}_j) = k^{\xi}_{aa}(\boldsymbol{t}_i - \boldsymbol{t}_j) + k^{\eta}_{aa}(\boldsymbol{t}_i - \boldsymbol{t}_j) + \delta_{ij}\sigma_a^2, \qquad a = 1, 2, 3,$$
$$k_{ab}(\boldsymbol{t}_i, \boldsymbol{t}_j) = k^{\xi}_{ab}(\boldsymbol{t}_i - \boldsymbol{t}_j), \qquad a \neq b,$$

and

$$k^{\xi}_{aa}(\boldsymbol{t}_i - \boldsymbol{t}_j) = \pi^{Q/2}\nu_{a0}^2|A_{a0}|^{-1/2} \exp\Big\{ -\frac{1}{4}(\boldsymbol{t}_i - \boldsymbol{t}_j)^\top A_{a0}(\boldsymbol{t}_i - \boldsymbol{t}_j)\Big\},\ a = 1, 2, 3,$$
$$k^{\xi}_{ab}(\boldsymbol{t}_i - \boldsymbol{t}_j) = (2\pi)^{Q/2}\nu_{a0}\nu_{b0}|A_{a0} + A_{b0}|^{-1/2} \exp\Big\{ -\frac{1}{2}(\boldsymbol{t}_i - \boldsymbol{t}_j - \boldsymbol{\mu}_{ab})^\top \Sigma_{ab}(\boldsymbol{t}_i - \boldsymbol{t}_j - \boldsymbol{\mu}_{ab})\Big\},$$
$$a \neq b,$$
$$k^{\eta}_{aa}(\boldsymbol{t}_i - \boldsymbol{t}_j) = \pi^{Q/2}\nu_{a1}^2|A_{a1}|^{-1/2} \exp\Big\{ -\frac{1}{4}(\boldsymbol{t}_i - \boldsymbol{t}_j)^\top A_{a1}(\boldsymbol{t}_i - \boldsymbol{t}_j)\Big\},\ a = 1, 2, 3,$$

where $\Sigma_{ab} = A_{a0}(A_{a0} + A_{b0})^{-1}A_{b0}$, and $\boldsymbol{\mu}_{ab} = \boldsymbol{\mu}_a - \boldsymbol{\mu}_b$.

Suppose that $X_a(\boldsymbol{t})$ has n_a observations, $a = 1, 2, 3$, and we want to construct the covariance matrix containing the autocovariances functions and cross-covariances functions between X's. Considering that $x_a(\boldsymbol{t})$ has training inputs $\boldsymbol{t}_{a,1}, \ldots, \boldsymbol{t}_{a,n_a}$, we construct the covariance matrices

$$\boldsymbol{\Psi}_{ab} = \boldsymbol{K}_{ab} + \delta_{ab}\sigma_a^2\boldsymbol{I} = \begin{bmatrix} k_{ab}(\boldsymbol{t}_{a,1}, \boldsymbol{t}_{b,1}) & \cdots & k_{ab}(\boldsymbol{t}_{a,1}, \boldsymbol{t}_{b,n_b}) \\ \vdots & \ddots & \vdots \\ k_{ab}(\boldsymbol{t}_{a,n_a}, \boldsymbol{t}_{b,1}) & \cdots & k_{ab}(\boldsymbol{t}_{a,n_a}, \boldsymbol{t}_{b,n_b}) \end{bmatrix}, \quad a, b = 1, 2, 3,$$

and take these matrices together to define a matrix $\boldsymbol{\Psi}_n$ such that

$$\boldsymbol{\Psi}_n = \begin{bmatrix} \boldsymbol{\Psi}_{11} & \boldsymbol{\Psi}_{12} & \boldsymbol{\Psi}_{13} \\ \boldsymbol{\Psi}_{21} & \boldsymbol{\Psi}_{22} & \boldsymbol{\Psi}_{23} \\ \boldsymbol{\Psi}_{31} & \boldsymbol{\Psi}_{32} & \boldsymbol{\Psi}_{33} \end{bmatrix}. \tag{5.2}$$

Let stack the sequences of observed data points of the three functions into a unique vector $\boldsymbol{x} = [x_1(\boldsymbol{t}_{1,1}), \ldots, x_1(\boldsymbol{t}_{1,n_1}), x_2(\boldsymbol{t}_{2,1}), \ldots, x_2(\boldsymbol{t}_{2,n_2}), x_3(\boldsymbol{t}_{3,1}), \ldots, x_3(\boldsymbol{t}_{3,n_3})]^\top$. Then we assume that

$$\boldsymbol{x} \sim N(\boldsymbol{0}, \boldsymbol{\Psi}_n),$$

where $n = n_1 + n_2 + n_3$, and learn the model by maximising the log-likelihood function given by

$$\mathcal{L} = -\frac{1}{2}\log|\boldsymbol{\Psi}_n| - \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{\Psi}_n^{-1}\boldsymbol{x} - \frac{n}{2}\log 2\pi,$$

where $\boldsymbol{\Psi}_n$ is a function of the data and the smoothing kernel parameters.

Although we illustrated the case of just $M = 3$ outputs, the extension to the case of $M > 3$ is straightforward. In addition, the method is flexible in the sense it can handle inputs with dimension $Q > 2$. The difficulty of having several outputs and/or higher dimension of the inputs may be the computational cost and the large number of parameters to estimate.

Comparison between Independent GPs vs MGP

We generated the data as follows:

$$X_a(t) = \mu_a(t) + f_a(t) + \varepsilon_a, \quad \varepsilon_a \sim N(0, \sigma_a^2), \quad a = 1, 2, 3,$$

where $t \in [-5, 5]$ and $t \in \mathbb{R}$, with

$$\mu_1(t) = \exp\{t/2\}, \quad \mu_2(t) = 10\sin(t), \quad \mu_3(t) = 10\cos(t).$$

The terms $f_1(t)$, $f_2(t)$ and $f_3(t)$ are Gaussian processes with zero mean and auto- and cross-covariance functions given by

$$\mathrm{Cov}\Big[f_a(t), f_b(t')\Big] = \rho_{ab} \exp\Big\{ -\frac{1}{2\gamma_{ab}^2}(t - t')^2 \Big\}, \qquad a, b = 1, 2, 3.$$

In other words, if we consider that the length-scale parameter $\gamma_{ab} = \gamma,\ \forall(a, b)$, then we are actually simulating the data from a model where each element ot the multivariate function-valued process has the same covariance kernel $k(\cdot, \cdot)$. This is achieved by assuming that the covariance matrix $\boldsymbol{K}$ is the product between a matrix $\boldsymbol{P}$, which measures the pairwise dependence between the outputs X's, and a common kernel function $k(t, t')$ for the input space:

$$\begin{aligned} \boldsymbol{K} &= \boldsymbol{P} \otimes k(t, t') \\ &= \begin{bmatrix} \rho_{11} & \rho_{12} & \rho_{13} \\ \rho_{21} & \rho_{22} & \rho_{23} \\ \rho_{31} & \rho_{32} & \rho_{33} \end{bmatrix} \otimes k(t, t'), \end{aligned}$$

where $\otimes$ is the Kronecker product between two matrices and, in our example,

$$k(t, t') = \exp\Big\{ -\frac{1}{2\gamma^2}(t - t')^2 \Big\}.$$

To generate the multivariate Gaussians, equations (5.1) are evaluated elementwise to

construct a covariance matrix $\boldsymbol{K}$ which contains the information of both autocovariance and cross-covariance functions.

When generating the data, we set $\gamma_{ab} = 0.5,\ \forall(a, b)$, and

$$\boldsymbol{P} = \begin{bmatrix} 1 & 0.5 & 0.3 \\ 0.5 & 1 & 0.8 \\ 0.3 & 0.8 & 1 \end{bmatrix}$$

in eq. (5.1), so that all the functions are positively correlated. We generated all the random functions at randomly chosen points located in $[-5, 5]$.

We have used training sets of 50 individual curves for each random function. First, we modelled each process as an independent GP using the covariance function (3.8). Next, we used the MGP model described above to model the multivariate case. We assumed a zero mean function for both methods.

In Figures 5.2 and 5.3, we drawn from the posterior after simulating 25 and 15 observations, respectively, from the prior – with the realisations of X_2 of the interval $(-1, 2)$ removed. Observe that the true behaviour of X_2 is captured better by using the MGP approach.

To assess the prediction performance of both methods, we have repeated all the above procedures 100 times and calculated the prediction RMSE for the three functions considering the 50 equally spaced points in $[-5, 5]$ – see Tables (5.1) and (5.2).

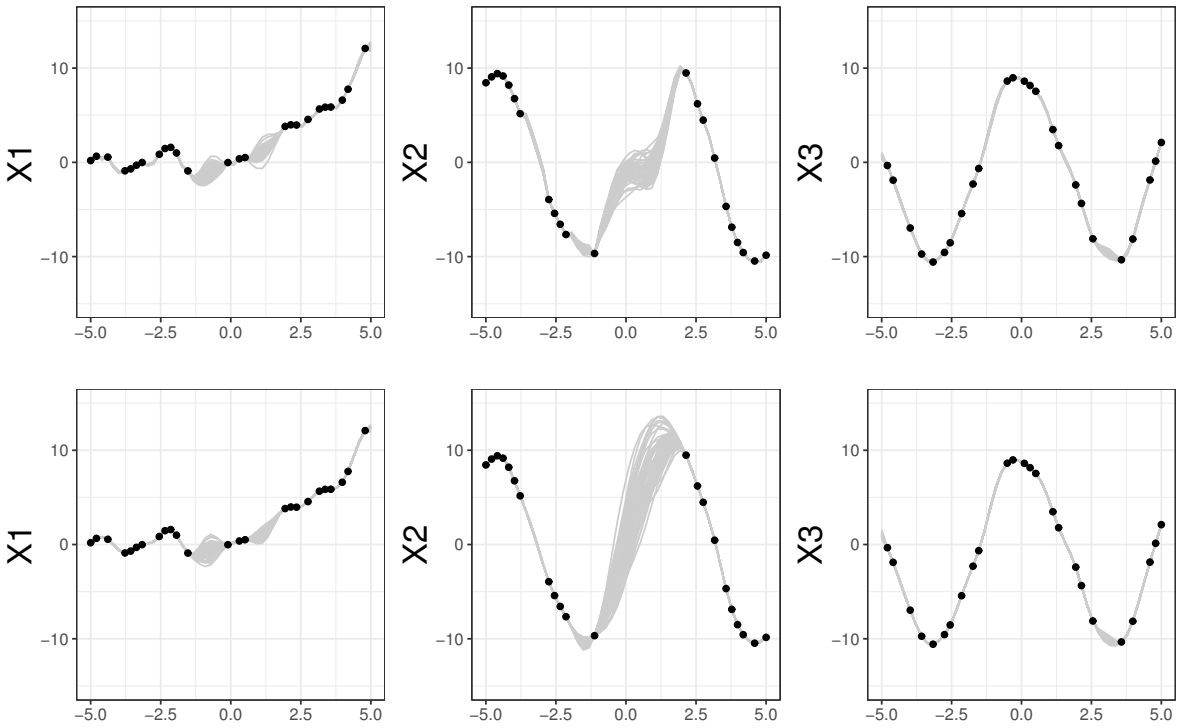

Figure 5.2: Random functions drawn from the posterior. We generated 25 observations of each random function and then we excluded the observations of X_2 belonging to the interval $(-1, 2)$.

	Independent GPs	**MGP**
X_1	1.066	1.050
X_2	3.637	1.714
X_3	1.154	1.117

Table 5.1: RMSE of the predictions made after 25 given observations

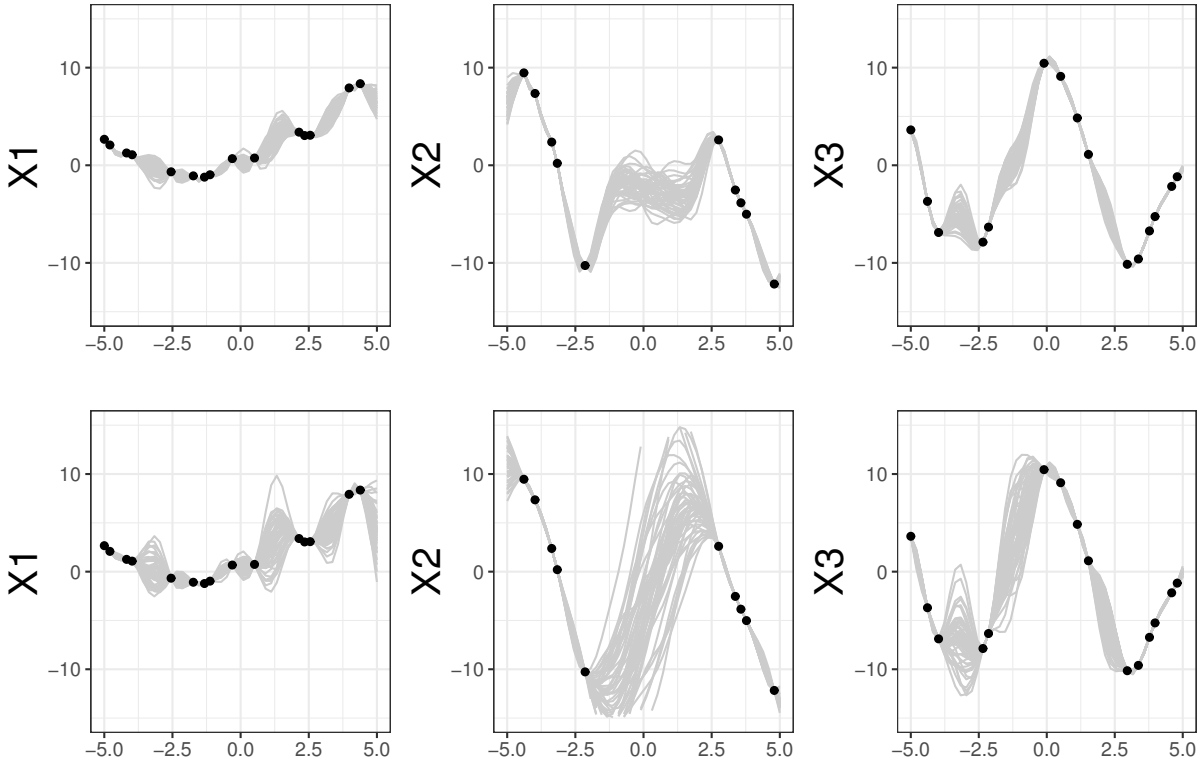

Figure 5.3: Random functions drawn from the posterior. We generated 15 observations of each random function and then we excluded the observations of X_2 belonging to the interval $(-1, 2)$.

	Independent GPs	**MGP**
X_1	1.321	1.501
X_2	4.800	2.622
X_3	1.501	2.094

Table 5.2: RMSE of the predictions made after 15 given observations

5.1.1 Simulation study 1

We simulate bivariate functional data $\boldsymbol{X}(t) = \big(X_1(t), X_2(t)\big)^\top$ based on a truncated multivariate Karhunen-Loève expansion representation

$$\boldsymbol{X}(t) = \sum_{j=1}^{J} \xi_j \boldsymbol{\phi}_j(t), \qquad t \in \mathcal{T} \subset \mathbb{R}, \tag{5.3}$$

where $\mathcal{T} = [0, 1]$. The multivariate basis functions $\boldsymbol{\phi}_j = (\phi_j^{(1)}, \phi_j^{(2)})^\top \in \mathbb{R}^2$ are constructed as follows. We use orthonormal Legendre polynomials of degree $0, \ldots, J-1$, on the domain $[0, 2]$ and split this domain into $M = 2$ parts. The first part represents the basis functions $\phi_j^{(1)}$ on $[0, 1]$ and the second part is shifted to form the second basis functions $\phi_j^{(2)}$ also on $[0, 1]$. Therefore, all the basis functions are defined on $\mathcal{T} = [0, 1]$. The scores ξ_j are simulated independently from a Gaussian distribution with zero mean and decreasing variance (in our example, the variance decrease linearly towards 0). The R package *funData* (Happ, 2018*a*) was used to simulate these data and the R package *MFPCA* (Happ, 2018*b*) was used to implement MFPCA.

Figure 5.4 show noise-free, dense data generate via (5.3) for $J = 7$ components. Then we remove observations (randomly selected) to consider different levels of sparsity: low sparsity (10% missing), medium (50%), high (80%) and very high (90%) sparsity. The case of medium sparsity can be seen in Figure 5.4. We are interested in evaluating predictions of the missing values by different models estimated by using only the observed data.

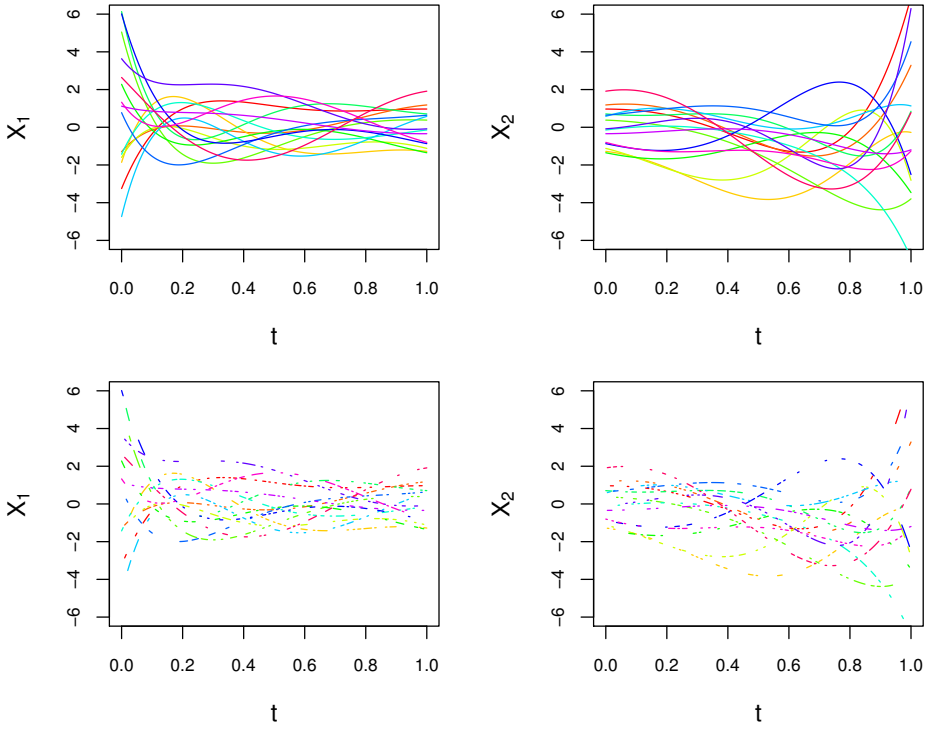

Figure 5.4: Noise-free bivariate functional data generated by using $J = 7$ components (orthonormal Legendre polynomials of maximum degree 6). Dense data are shown on the top. Medium sparsity case is shown on the bottom.

We use root standardized mean squared error (RSMSE) to evaluate the quality of the predictions and analyse scenarios with different number of replicated curves ($N = 15, 30,$ and 50) and different noise variance for the measurement error ($\sigma_\varepsilon^2 = 0.1^2$ and 0.5^2). The simulation study for each scenario is based on 50 datasets. For each scenario, therefore, we have 50 RSMSE values which we use for producing the boxplots in Figure 5.5. We can clearly observe that, mainly in settings of high sparsity, MGP predictions can be more accurate than those made by MFPCA using the same number of components.

Settings of less ($J = 4$) and more ($J = 10$) components were also considered and can be seen in Section 5.5.

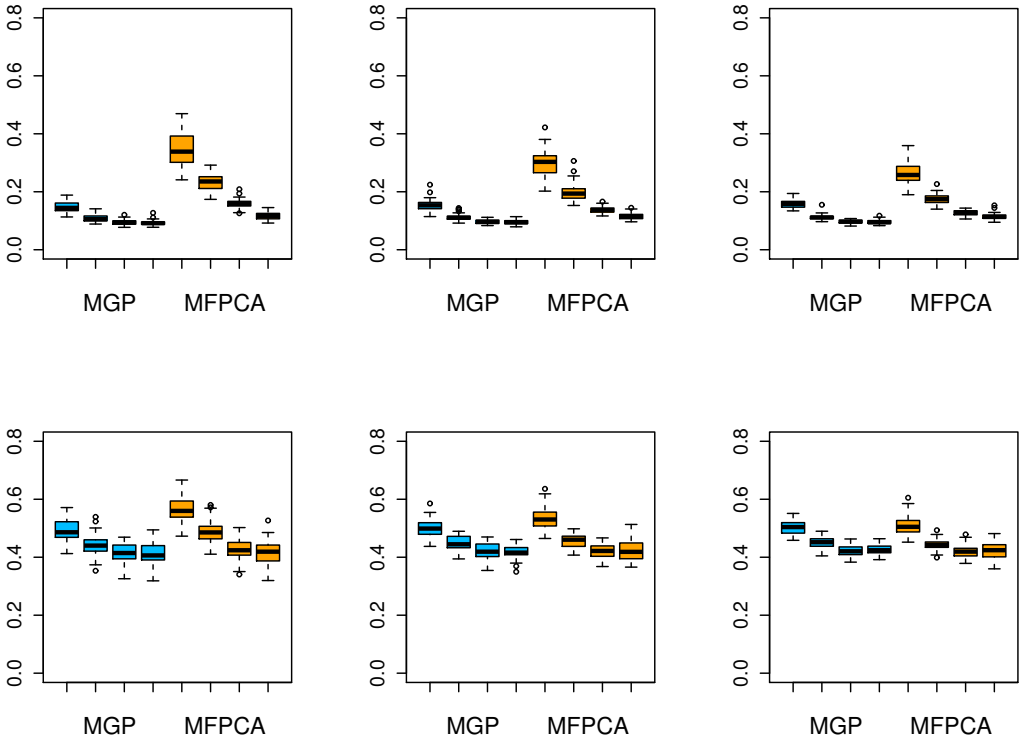

Figure 5.5: Boxplots of the prediction RSMSE using MGP (first group of four boxes) and MFPCA (last group of four boxes) calculated from datasets with different number of replicated curves: $N = 15$ (first column) $N = 30$ (second column), and $N = 50$ (third column), all generated from orthonormal Legendre polynomials of maximum degree 6 with measurement error whose variance is $\sigma_\varepsilon^2 = 0.1^2$ (first row), and $\sigma_\varepsilon^2 = 0.5^2$ (second row). From left to right, each group of four boxplots corresponds to the cases of very high, high, medium, and low sparsity.

5.1.2 Simulation study 2

We now simulate data from a function-valued process $\boldsymbol{X}(t) = \big(X_1(t), X_2(t)\big)^\top$, $t \in \mathcal{T} \subset \mathbb{R}$, defined by the sum of a global GP and a local GP, as follows:

$$\begin{aligned} \boldsymbol{X}(t) &= \boldsymbol{X}_{\text{global}}(t) + \boldsymbol{X}_{\text{local}}(t), \\ \boldsymbol{X}_{\text{global}}(t) &\sim \text{GP}\big(\mu, g_1(t,t')\big), \\ \boldsymbol{X}_{\text{local}}(t) &\sim \text{GP}\big(0, \sigma(t)\sigma(t')g_2(t,t')\big), \end{aligned} \tag{5.4}$$

where $g_1(\cdot,\cdot)$ and $g_2(\cdot,\cdot)$ are squared exponential covariance functions constructed by the MGP model, that is, they have blocks of auto- and cross-covariance functions as in eq. (5.2). The univariate model for function-valued processes defined similarly as above was proposed by Ba & Joseph (2012) and called composite GP model. Note that

the nonstationarity is defined by the varying standard deviation $\sigma(t)$. Whereas the global process is stationary, the local process can be seen as a particular case of the nonstationary covariance model (4.6) with constant anisotropy matrix.

To simulate data, we assume that $g_1(\cdot,\cdot)$ is the standardised covariance function constructed as above with smoothing kernel parameters $\nu_{10} = -1$, $\nu_{20} = 1$, $\nu_{11} = \nu_{21} = 0.1$, $A_{10} = A_{20} = 30$, $A_{11} = A_{21} = 20$. The covariance kernel $g_2(\cdot,\cdot)$ is equal to $g_1(\cdot,\cdot)$ except all the A parameters, to which we added 2000 to produce more local fluctuation in the local process. An additional measurement noise with variances $\sigma_\varepsilon^2 = 0.1^2$ and $\sigma_\varepsilon^2 = 0.5^2$ are used. Finally, we use a local standard deviation $\sigma(t) \propto \exp\{-20t\}$, so that the variance of the local bivariate random process is large near $t = 0$ and decreases exponentially as t increases (see Figure 5.6).

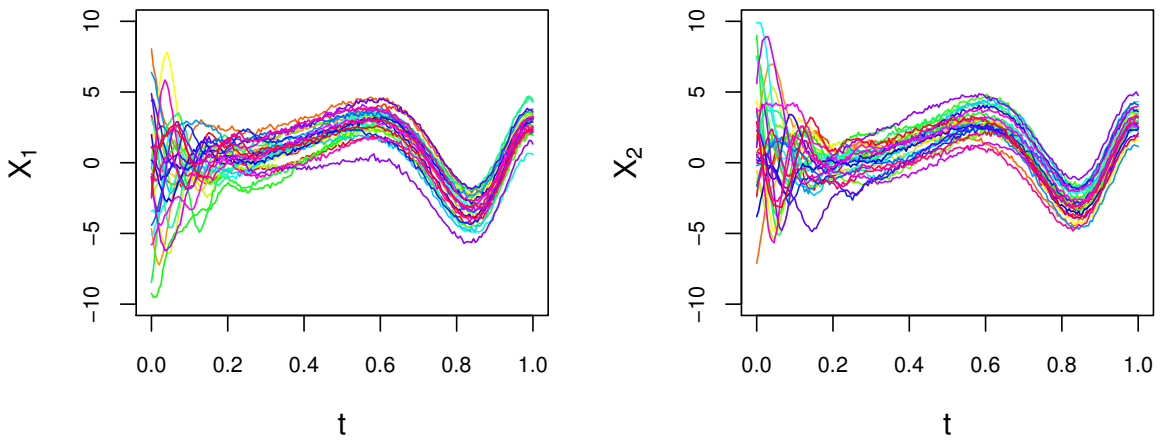

Figure 5.6: Bivariate nonstationary functional data simulated with a measurement error of variance $\sigma_\varepsilon^2 = 0.1^2$.

We estimate the covariance structure of the data generated from (5.4) via the stationary MGP model, introduced in the beginning of this chapter. This is equivalent to use only the global part in (5.4). In addition, we consider to model the sum of a global and a local process as in (5.4), and call this the nonstationary MGP (NSMGP) model, where both covariance functions $g_1(\cdot,\cdot)$ and $g_2(\cdot,\cdot)$ are estimated via MGP. Ba & Joseph (2012) estimate $\sigma(t)$ by using a Gaussian kernel regression model based on the residuals obtained for a given global trend. We estimate $\sigma(t)$ via B-splines basis functions as we have done in Chapter 4. Finally, MFPCA model was also used.

After learning the covariance structure by using both stationary MGP, NSMGP and MFPCA models, we access predictions of values of entire curves given a small subset of observed data points. Figure 5.7 illustrates the MGP and NSMGP predictions for a single bivariate curve. As the stationary model assumes constant variance of the multivariate process over t, the estimated variance of the process where $t > 0.2$ is overestimated because

of the large variability near $t = 0$. This is why the stationary model obtains predictions with large variance even when there is no much uncertainty in the true process (low variation after $t = 0.2$). Note how the NSMGP model improved predictions after learning that the process has different variance for different locations. In addition, we can also observe that the disadvantage of assuming stationarity is accentuated when the observed datapoints are sparse.

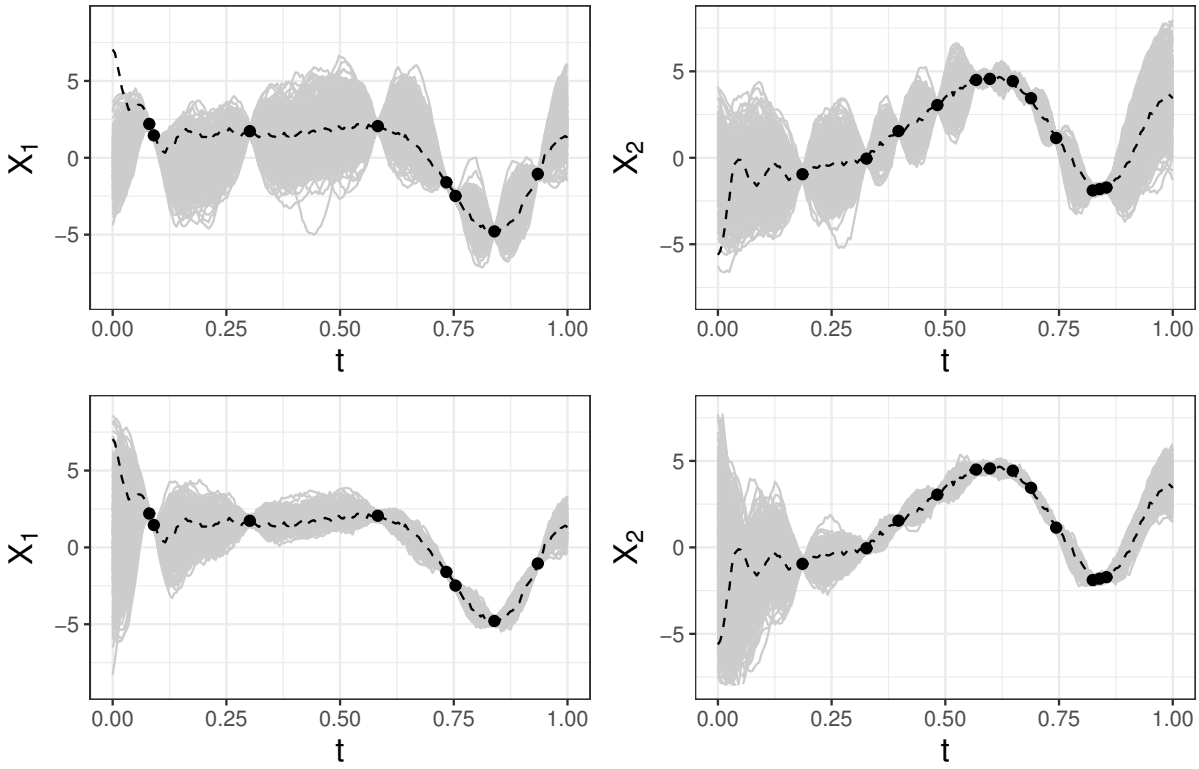

Figure 5.7: Predictions (100 grey curves) obtained by MGP (first row) and NSMGP (second row) given a few observations (black points). The true realisations are represented by the dashed lines.

We repeat the procedure 50 times. The predictions of each repetition are assessed by the prediction RSMSE and the 50 RSMSE values are used to produce the boxplots in Figure 5.8. As in the previous simulation study, MGP method provides much better predictions than MFPCA does, but the predictions are even more accurate when we consider the nonstationary version (NSMGP).

Figure 5.9 shows the CFVEs obtained by the three methods. Note that when nonstationarity is considered, one can obtain competitive results compared to the ones obtained by MFPCA.

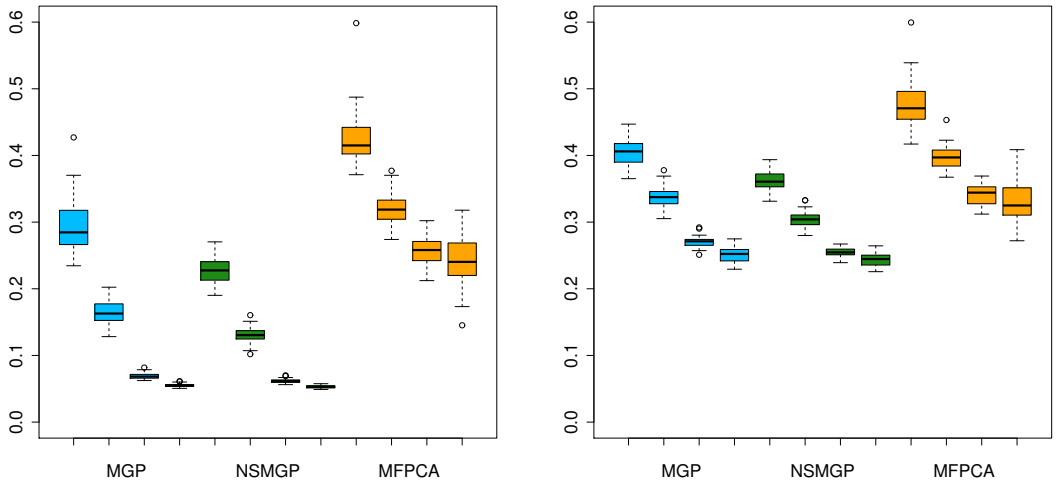

Figure 5.8: Boxplots of the prediction RSMSE using stationary MGP (first group of four boxes), NSMGP (second group of four boxes), and MFPCA (third group of four boxes) calculated from datasets with $N = 30$ replicated curves which have measurement error with variance $\sigma_\varepsilon^2 = 0.1^2$ (first column) and $\sigma_\varepsilon^2 = 0.5^2$ (second column). From left to right, each group of four boxplots corresponds to the cases of very high, high, medium, and low sparsity.

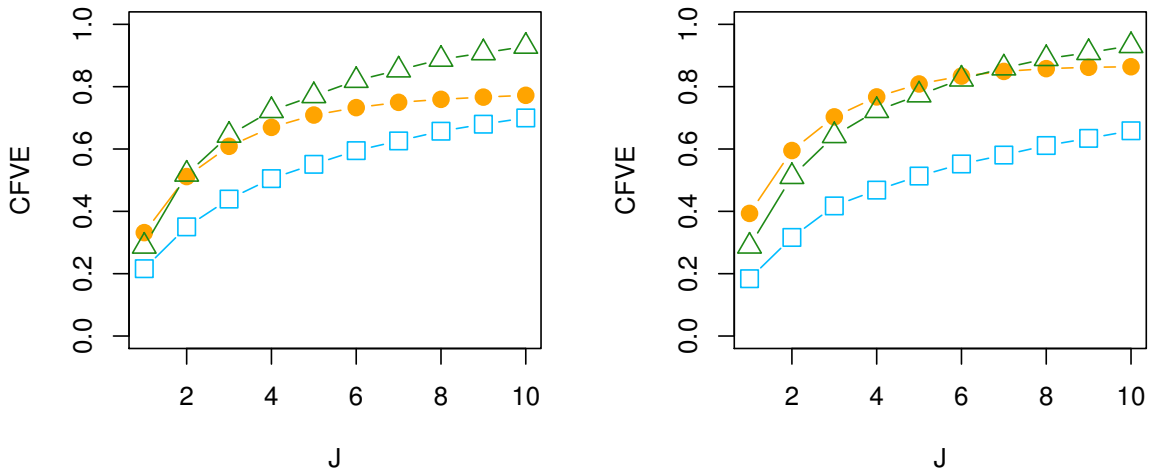

Figure 5.9: CFVEs obtained by MGP (blue squares), NSMGP (green triangles), and MFPCA (orange points) for nonstationary data with $N = 50$ replicated curves and high sparsity (left) and low sparsity (right).

5.2 Application to Human Fertility Data

In this section, we apply the Standard 2d FPCA model (following Yao *et al.* (2005)), the Marginal FPCA model (Chen *et al.*, 2017), and the NSMGP model to age-specific fertility

rates (ASFR) (Human Fertility Database, 2017). Firstly, eigensurfaces estimated by each model are analysed. Next, we will assess prediction results.

We estimate $\mu(s, \tau)$ in the same way as Chen *et al.* (2017) have done: for each age and year, we calculate the sample mean across the fertility rates of all countries. In this way, we suppose that the countries have a common mean function and then we analyse the variability across the countries.

The NSMGP model used was basically the same as the one used in the previous simulation study. We model the fertility rates as a function of two arguments (woman's age s and calendar year τ). Since $\boldsymbol{t} = (s, \tau)^\top$ is two-dimensional, we estimate the local standard deviation $\sigma(s, \tau)$ via cubic regression splines for each marginal basis and a tensor product smooth, allowing for smooth interaction between the components of s and τ.

5.2.1 Analysing eigensurfaces

We are first interested in comparing the eigensurfaces obtained by FPCA models with those obtained by the NSMGP model using the fertility rates of the four following countries: Canada, United States, Spain and Netherlands. We intentionally choose these countries because there seems to be a high correlation between the first two and between the last two, as we can see in Figure 5.10.

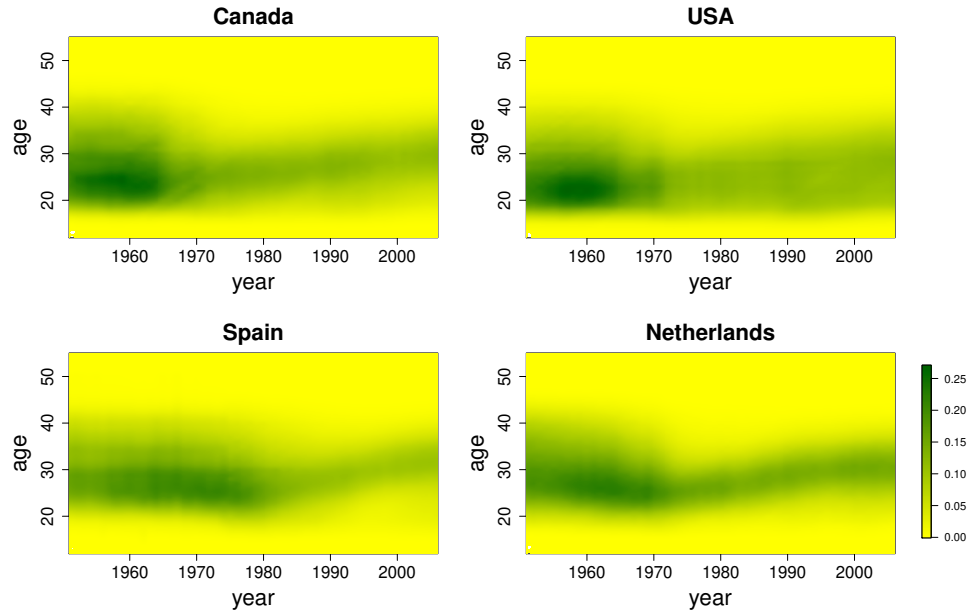

Figure 5.10: Human fertility rates of Canada, USA, Spain, Netherlands.

Standard 2d FPCA

We first perform FPCA on the ASFR data following (Yao *et al.*, 2005), which is implemented in the PACE package[1]. We call this model the 'Standard 2d FPCA'. Each country is observed over a grid of 44×56 points. We first rearrange the $M = 4$ matrices (each one with dimension 44×56) into a big matrix with dimension $(44 \cdot 56) \times M$, and then perform FPCA on this resulting matrix. In this way, each column represents one realisation (country) of the function-valued process X. To obtain the eigensurfaces, we rearrange the resulting eigenfunctions (stored as vectors of length $44 \cdot 56$) into matrices of dimension 44×56. As we are using only four countries, we cannot estimate more than two components using Standard 2d FPCA. These two components are plotted in Figure 5.11. As we can see, it is rather difficult to make interpretation about the inputs (age and year) separately.

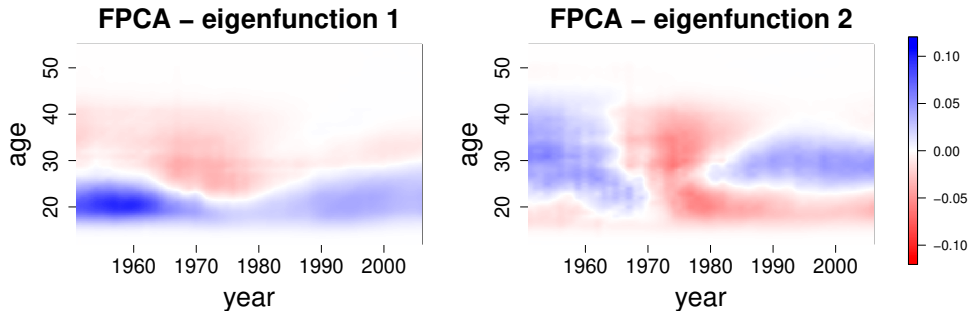

Figure 5.11: Leading eigensurfaces estimated by the Standard 2d FPCA model for ASFR of Canada, USA, Spain and Netherlands.

Marginal FPCA

We also apply the Marginal FPCA model (Chen *et al.*, 2017). In Figure 5.12, we can see the separable eigensurfaces estimated by Marginal FPCA. The first component seems to represent a contrast between fertility before and after the age of 27 years. The second component shows a contrast betweeen fertility rates in the period 1965–1985 and the other years of the sample. The third one can be interpreted as a size component, where the fertility variation between the countries is larger in the first two decades of the sample. Finally, the fourth component seems to capture the baby boom in 1960's in USA and Canada.

[1] `http://www.stat.ucdavis.edu/PACE`

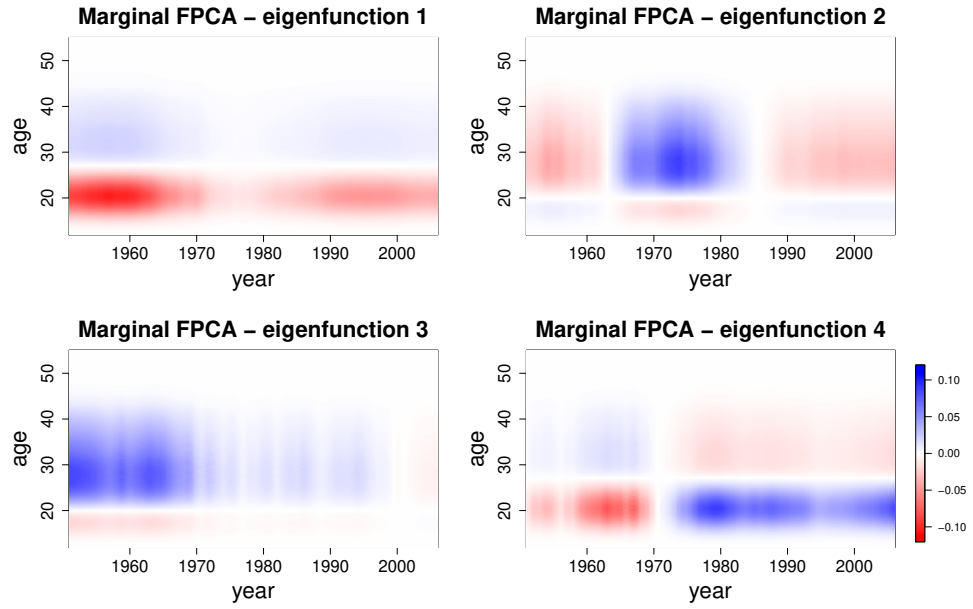

Figure 5.12: Leading eigensurfaces estimated by the Marginal FPCA model for ASFR of Canada, USA, Spain and Netherlands.

Nonstationary MGP

Finally, we apply the nonstationary version of our proposed MGP to learn the covariance structure in the ASFR data. Whereas Standard FPCA and Marginal FPCA models consider that the countries are independent realisations of the same process, our approach models each country as a different process and learns the cross-covariance function between them. For these data, we could not compare our model with MFPCA because the surface (age $\times$ year) of each country is observed just once.

The eigensurfaces estimated by the nonstationary MGP model are shown in Figures 5.13–5.16, from where we draw the following conclusions.

1st NSMGP eigensurface: highlights that there is a contrast between the first two countries and the last two, mainly for the 1960s' and 1970s' and for young women. It also shows that although USA and Canada covary positively about the mean across countries, USA have bigger contrast with Spain and Netherlands.

2nd NSMGP eigensurface: the white regions shows a contrast between between early ages and early year with late age and late year. As the white regions have diagonal shape, we can observe that this contrast is not constant over women's age or calendar year, and this feature could not be observed if one assumed covariance separability.

3rd NSMGP eigensurface: this seems to represent precisely the baby boom in 1960's

in Canada and USA, whose young women had high fertility rates. In other words, there was a young fertility concentration in those calendar years.

4th NSMGP eigensurface: this seems to highlight the mature fertility concentration in Spain and Netherlands in the first calendar years (compared to the Canada and USA).

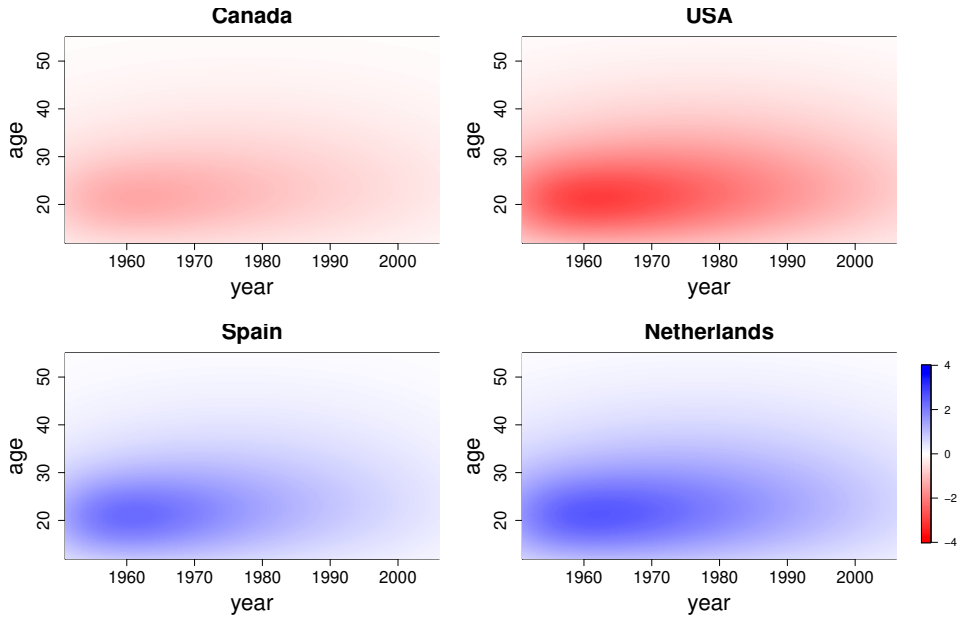

Figure 5.13: NSMGP's estimated eigensurface 1. ASFR of Canada, USA, Spain and Netherlands.

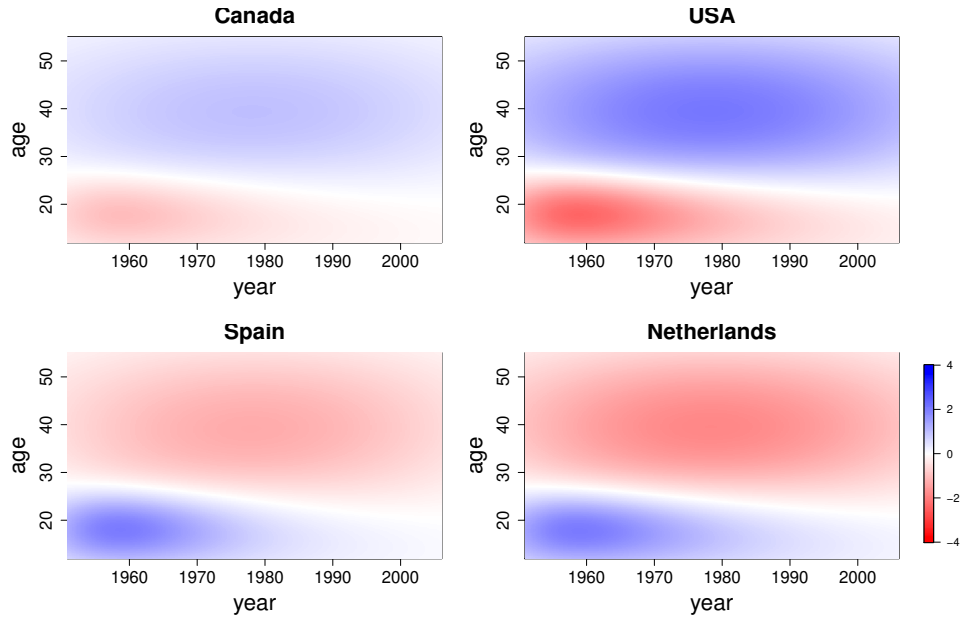

Figure 5.14: NSMGP's estimated eigensurface 2. ASFR of Canada, USA, Spain and Netherlands.

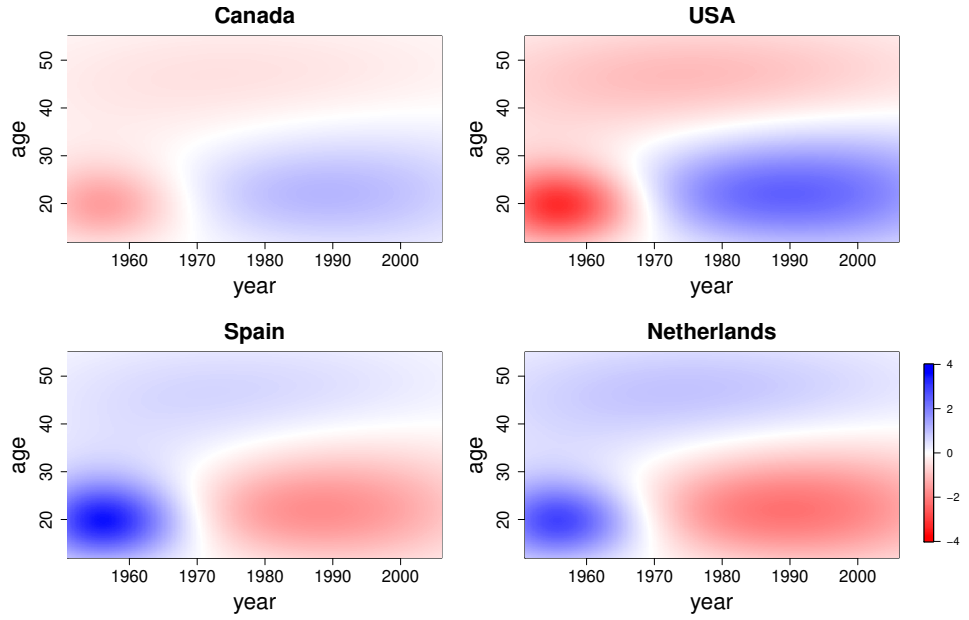

Figure 5.15: NSMGP's estimated eigensurface 3. ASFR of Canada, USA, Spain and Netherlands.

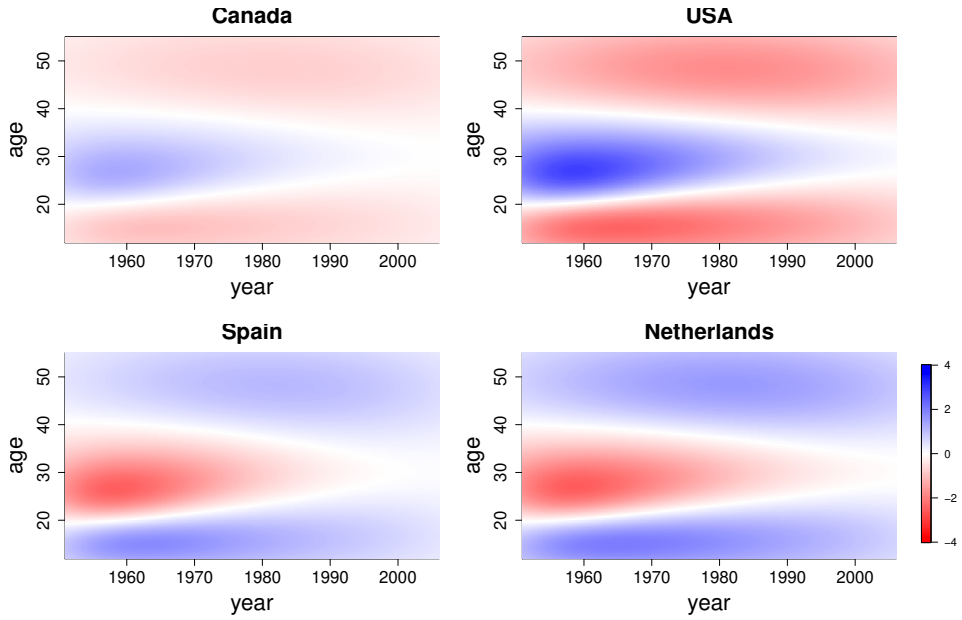

Figure 5.16: NSMGP's estimated eigensurface 4. ASFR of Canada, USA, Spain and Netherlands.

5.2.2 Assessing predictions

The ASFR dataset has several other countries with missing data. For example, the ASFR of some countries were not recorded in some calendar years. Therefore, the researcher might be interested in imputing values where there are missing data by using a model learnt from the observed data. In this situation, FPCA models would not take into account the information available of other (possibly similar) observed countries in those calendar years to predict those missing values.

To mimic the fertility rates, Chen *et al.* (2017) simulated fictitious countries by taking the Marginal FPCA model estimated from their application to ASFR data and then added an independent noise to mimic the noise level of those real data.

We used 24 sets of four fictitious countries each and evaluated the predictions of test set values made by Marginal FPCA, Product FPCA and NSMGP. For each set of countries, in each replication, the test set contains one (randomly selected) year t_i of a (randomly selected) country X_a, while the training set contains all the other available information. This procedure was repeated 100 times.

The prediction RMSE results are summarised in Table 5.3. In order to save computational time, we use different m in Nyström method in the estimation process. Next, we make predictions conditional only on the training data close to the test data we want to predict. For example, $r = 3$ means that we make predictions for $X_a(t_i)$ using only the

neighbourhood observations, i.e. at calendar years $\{t_{i-3}, \ldots, t_{i+3}\}$, of the training data.

We can see that even generating data from the Marginal FPCA model, NSMGP has better prediction performance and there is no need to use more than $m = 500$ samples in Nyström method. In addition, the cases where $r = 3$ or $r = 5$ already provide good predictions while saving computational time. Therefore, the results indicate that NSMGP method has had good prediction performance even when we used fairly small m and r to reduce computational time.

Marginal FPCA	1.1440
Product FPCA	1.0000

NSMGP	Nyström ($m = 200$)	Nyström ($m = 500$)
$r = 1$	0.9342	0.7365
$r = 2$	0.9120	0.7144
$r = 3$	0.9152	0.7132
$r = 5$	0.9262	0.7262

Table 5.3: Prediction RMSEs, relative to Product FPCA model, for simulated human fertility data.

5.3 Implementation

When modelling multivariate function-valued processes, the number of parameters to estimate may be quite large. In this case, optimisation algorithms might be sensitive to the initial values of parameters. In order to obtain faster and more stable results in the likelihood maximisation, it is ideal to start with good initial values for the parameters which are being optimised. As we may have no idea about the true values of the parameters, we suggest evaluating the likelihood function for a reasonable number of different values of the parameters and starting the optimisation with the vector which provided the highest value for the likelihood function. In general, we used 500 different vectors, but when there were many parameters (more than 10) to be optimised, then we evaluated $2,000$ different vectors before starting the optimisation process.

We should be careful when trying to interpret the parameters estimates of the smoothing kernel used to construct the convolved GPs. This is because different parameters may have similar effects in the likelihood function $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$. For example, in equations (5.1), a kernel of the form $h_{20}(\boldsymbol{t}) = \nu_{20} \exp\big\{ -\frac{1}{2}(\boldsymbol{t} - \boldsymbol{\mu}_2)^\top A_{20}(\boldsymbol{t} - \boldsymbol{\mu}_2)\big\}$ can be used to model lag dependence between the processes X_1 and X_2, where the lag distance is described by

$\boldsymbol{\mu}_2$. Suppose X_1 and X_2 are weakly correlated. There might be basically no difference between choosing a small value for ν_{20} or a very large value for $\boldsymbol{\mu}_2$, and this makes the optimisation algorithm choose arbitrarily between the combination of parameter values. In our implementations, we have assumed that there is no lag effect. This assumption makes the interpretation of the parameters simpler and is reasonable for many applications.

In addition to the computational strategies mentioned in Section 3.7, we have taken advantage of the symmetry and positive-definiteness properties of covariance matrices to speed up their inversion. This was done by using functions provided by the Armadillo C++ library.

In the nonstationary model, in order to use multivariate B-splines, we used the R package mgcv (Wood, 2018). This enabled us to implement cubic regression splines for each marginal basis and a tensor product smooth.

5.4 Conclusion

Unlike FPCA, MGP does not assume independent realisations of the same function-valued process; MGP explicitly models the cross-covariance function between the processes. MFPCA would be a more natural competing model for dealing with multivariate functional data. However, MFPCA requires multiple realisations of each function, whereas MGP needs only one. This is an important feature of the MGP method, since many applications involve a unique observation (e.g. geostatistics data, fertility and mortality data).

By modelling the cross-covariance functions between elements of the multivariate function-valued process, we show that MGP can be helpful to analyse joint variation in the data, as we have seen in the application to human fertility data. This can be further used in clustering analysis. Finally, the estimated cross-covariance functions can be useful to improve predictions of missing values of one function by borrowing information from dependent functions.

5.5 Appendix

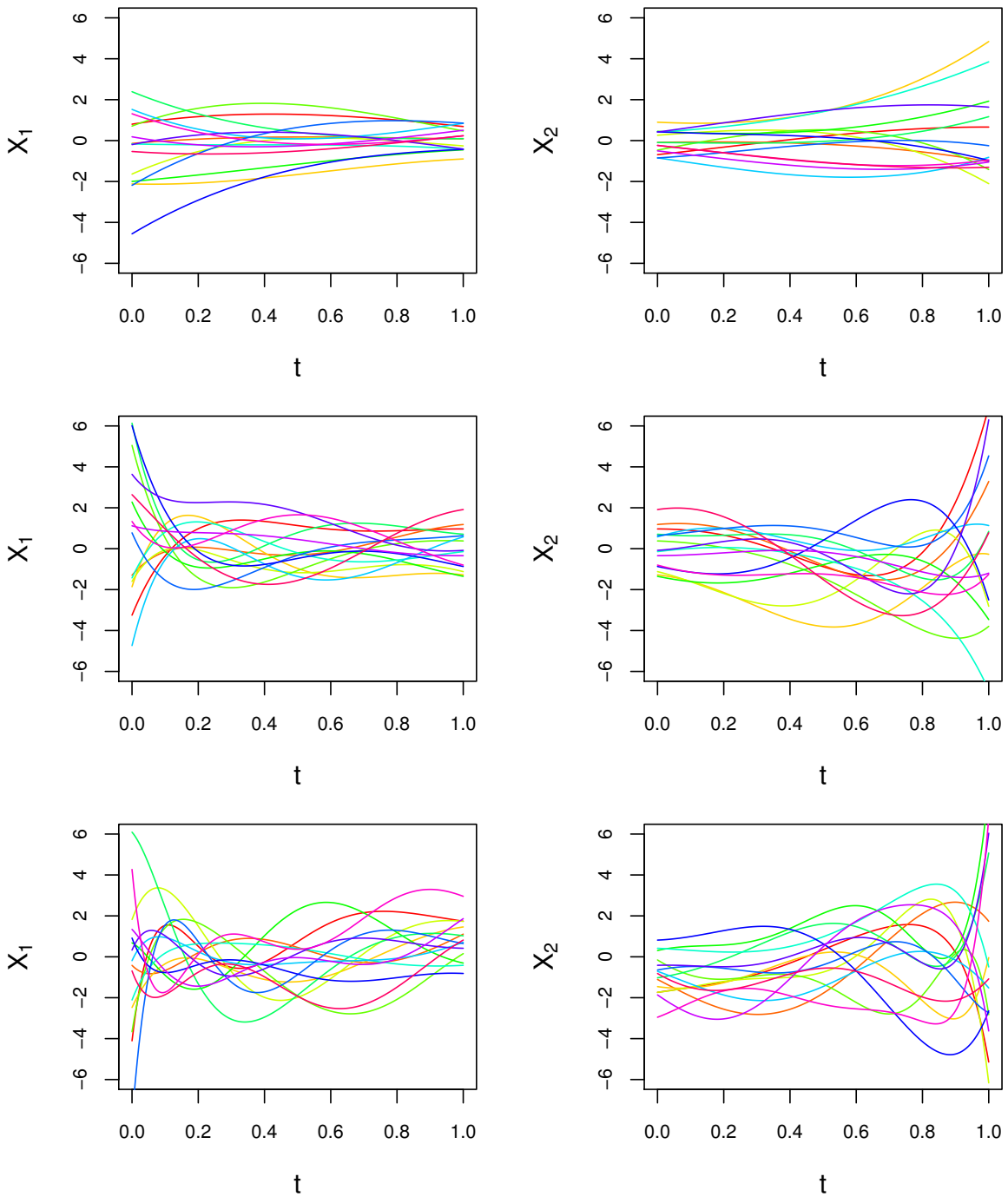

Figure 5.17: Noise-free bivariate functional data generated by using orthonormal Legendre polynomials of maximum degree 3 (first row), 6 (second row), and 9 (third row).

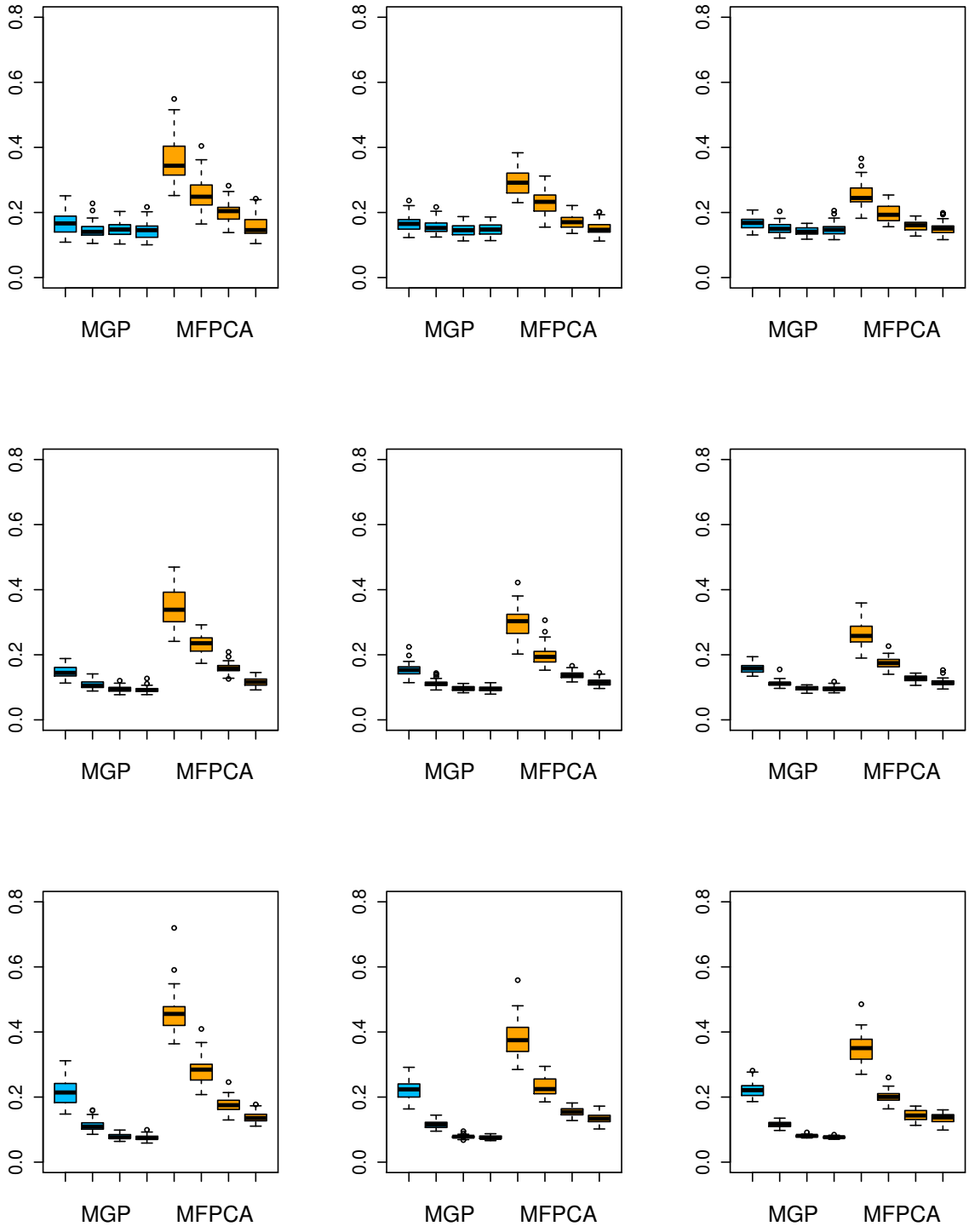

Figure 5.18: Boxplots of the prediction RSMSE using MGP (first group of 4 four boxes) and MFPCA (last group of four boxes) calculated from datasets with different number of replicated curves: $N = 15$ (first column) $N = 30$ (second column), and $N = 50$ (third column), all generated from orthonormal Legendre polynomials of maximum degree 3 (first row), 6 (second row), and 9 (third row) with measurement error with variance $\sigma_\varepsilon^2 = 0.1^2$. From left to right, each group of four boxplots corresponds to the cases of low, medium, high and very high sparsity.

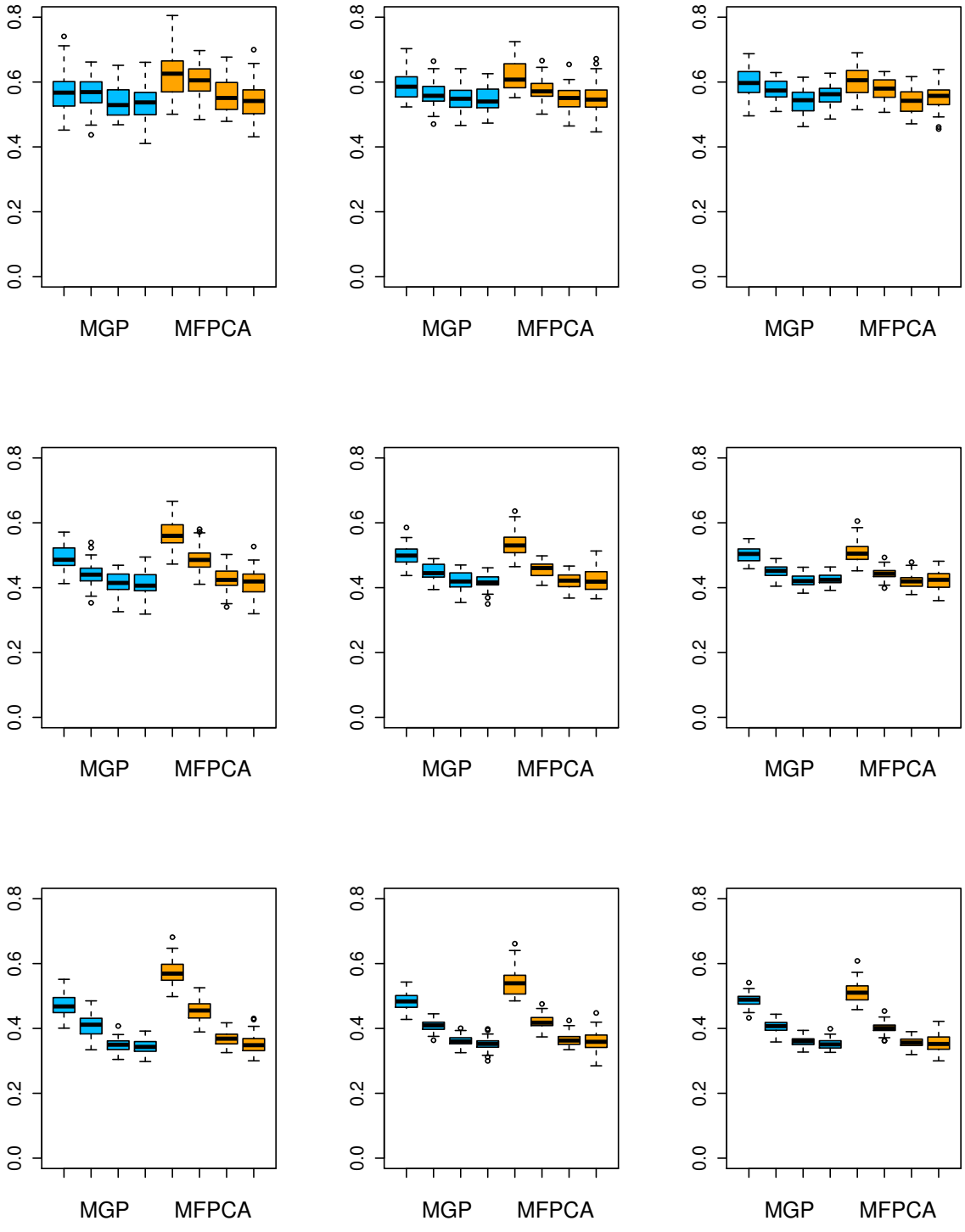

Figure 5.19: Boxplots of the prediction RSMSE using MGP (first group of 4 four boxes) and MFPCA (last group of four boxes) calculated from datasets with different number of replicated curves: $N = 15$ (first column) $N = 30$ (second column), and $N = 50$ (third column), all generated from orthonormal Legendre polynomials of maximum degree 3 (first row), 6 (second row), and 9 (third row) with measurement error with variance $\sigma_\varepsilon^2 = 0.5^2$. From left to right, each group of four boxplots corresponds to the cases of low, medium, high and very high sparsity.

Chapter 6

Multiple Functional PLS regression model

In Section 6.1, we closely follow the notation and exposition of Delaigle & Hall (2012) to introduce their explicit formulation of PLS basis for functional data and to explain their proposed algorithm. In a simulation study, under different scenarios we compare prediction results obtained by FPLSR and FPCR models using different numbers of components.

In Section 6.2, we propose an extension to the case involving multiple function-valued covariates. The estimation of the covariance structure of the multiple functional covariates is estimated as we proposed in Chapter 5. Therefore, Chapter 5 is used as a building-block for this chapter. In a simulation study in the end of the chapter, we compare MFPLSR and MFPCR in terms of prediction of a scalar response variable.

6.1 Functional Partial Least Squares regression

In order to estimate the scalar-on-function regression model, we can use FPLS basis functions. Until recently, due to the iterative nature of standard PLS algorithms, it was difficult to create intuition and to unveil properties of FPLS basis functions in an explicit way. Thus, Delaigle & Hall (2012) develop a new theory for PLS method, the 'alternative PLS' (APLS), in which the FPLS basis functions can be expressed only in terms of functions that are explicitly computable.

Let $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ be a sample of independent data pairs of the scalar random variable Y and the random function X which is defined on the compact interval

$\mathcal{T}$ satisfying $\int_{\mathcal{T}} \mathrm{E}\left[X(t)^2\right] dt < \infty$. Thus, the functional regression model is given by

$$Y = a + \int_{\mathcal{T}} b(t)X(t)dt + \varepsilon. \tag{6.1}$$

Therefore, to predict the value of Y given a known value x of the curve X, we estimate the function

$$g\big(x(t)\big) = \mathrm{E}\big[Y|X(t) = x(t)\big] = a + \int_{\mathcal{T}} b(t)x(t)dt. \tag{6.2}$$

As a is constant, from (6.1) we obtain that $a = \mathrm{E}\big[a\big] = \mathrm{E}\big[Y\big] - \int_{\mathcal{T}} b(t)\mathrm{E}\big[X(t)\big]dt$ and therefore (6.2) becomes

$$g\big(x(t)\big) = \mathrm{E}\big[Y\big] + \int_{\mathcal{T}} b(t)\mathrm{E}\big[x(t) - \mathrm{E}\left[X(t)\right]\big]dt.$$

Therefore, we need to estimate the scalar a and the function b from the data. Let us write x and b in terms of an orthonormal basis $\phi_1, \phi_2, \ldots$ defined on $\mathcal{T}$:

$$\begin{aligned} x(t) &= \sum_j \Big(\int_{\mathcal{T}} x(t)\phi_j(t)dt\Big)\phi_j(t), \\ b(t) &= \sum_j \nu_j\phi_j(t), \qquad \nu_j = \int_{\mathcal{T}} b(t)\phi_j(t)dt. \end{aligned}$$

As the basis functions are orthonormal, we can see that $\int_{\mathcal{T}} b(t)x(t)dt = \sum_j \nu_j \int_{\mathcal{T}} x(t)\phi_j(t)dt$.

We now have to choose the basis functions $\phi_1, \phi_2, \ldots$ and determine $\nu_1, \nu_2, \ldots$ given those basis functions. Note that

$$\begin{aligned} \int_{\mathcal{T}} b(t)\mathrm{E}\big[x(t) - \mathrm{E}\left[X(t)\right]\big]dt &= \int_{\mathcal{T}} \big(\sum_j \nu_j\phi_j\big)\mathrm{E}\big[x(t) - \mathrm{E}\left[X(t)\right]\big]dt \\ &= \sum_j \nu_j \int_{\mathcal{T}} \big(x(t) - \mathrm{E}\left[X(t)\right]\big)\phi_j(t)dt. \end{aligned}$$

We define $\beta_1, \ldots, \beta_J$ to be the sequence $\nu_1, \ldots, \nu_J$ that minimises

$$s_J(\nu_1, \ldots, \nu_J) = \mathrm{E}\bigg[\int_{\mathcal{T}} b(t)\big(x(t) - \mathrm{E}\left[X(t)\right]\big)dt - \sum_{j=1}^{J} \nu_j \int_{\mathcal{T}} \big(x(t) - \mathrm{E}\left[X(t)\right]\big)\phi_j(t)dt \bigg]^2.$$

Therefore, as in practice we can only calculate a finite number of terms, the functions

$$b_J(t) = \sum_{j=1}^{J} \beta_j\phi_j(t),$$

and

$$g_J\big(x(t)\big) = \mathrm{E}\left[Y\right] + \int_{\mathcal{T}} b_J(t)\mathrm{E}\big[x(t) - \mathrm{E}\left[X(t)\right]\big]dt$$
$$= \mathrm{E}\left[Y\right] + \sum_{j=1}^{J} \beta_j \int_{\mathcal{T}} \mathrm{E}\big[x(t) - \mathrm{E}\left[X(t)\right]\big]\phi_j(t)dt \tag{6.3}$$

are approximations to $b(t)$ and $g\big(x(t)\big)$, respectively.

Once the functions ϕ_j are known (or chosen), we find $\hat{\beta}_1, \ldots, \hat{\beta}_J$ by solving

$$\hat{\beta}_1, \ldots, \hat{\beta}_J = \underset{\nu_1,\ldots,\nu_J}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N}\left\{y_i - \bar{y} - \sum_{j=1}^{J}\nu_j \int_{\mathcal{T}}(x_i(t) - \bar{x}(t))\phi_j(t)dt\right\}^2$$

and use these as estimates of $\beta_1, \ldots, \beta_J$ in (6.3) to obtain the approximation to g.

They can be chosen to be independently of the data (e.g. B-splines), However, as discussed in Section 2.4, there is no guarantee that the first J elements of such bases should explain the most important variation about the regression function g. Using FPC basis functions (see Section 2.3) is a common strategy to use the information of the data, but we cannot ensure that g is well explained by first few principal components of X. For example, important terms to explain the variation of g might come from later FPCs.

The orthonormal PLS basis

We might be interested to capture some information about the interaction between the functional predictor and the response when we construct the basis. The standard PLS basis adapted to the functional context is defined iteratively by choosing ϕ_J in a sequential manner. In summary, at each step J, we maximise the covariance functional

$$f_J(\phi_J) = \mathrm{Cov}\left[Y - g_{J-1}(X), \int_{\mathcal{T}} X(t)\phi_J(t)dt\right], \tag{6.4}$$

subject to

$$\int_{\mathcal{T}}\int_{\mathcal{T}} \phi_j(s)k(s,t)\phi_J(t)dsdt = 0, \qquad \text{for} \quad 1 \leq j \leq J-1, \tag{6.5}$$

and

$$||\phi_J|| = 1, \tag{6.6}$$

where $\phi_1, \ldots, \phi_{J-1}$ have already been determined.

We are going to explain the theoretical PLS basis defined by the last three equations and describe the space generated by the first J PLS basis functions $\phi_1, \ldots, \phi_J$. These properties motivate an alternative approach of functional PLS, the APLS.

It can be shown that the function ϕ_J that maximises f in (6.4), given $\phi_1, \ldots, \phi_{J-1}$ and subject to (6.5) and (6.6), is determined by

$$\phi_J = c_0 \left[K \left\{ b(t) - \sum_{k=1}^{J-1} \Big(\int_{\mathcal{T}} b(t)\phi_j(t) \Big) \phi_j(t) dt \right\} + \sum_{k=1}^{J-1} c_k \phi_k \right],$$

where, for $1 \leq k \leq J-1$, the constants c_k are obtained by solving the linear system of $J-1$ equations

$$\int_{\mathcal{T}} \int_{\mathcal{T}} \phi_j(s) k(s,t) \phi_J(t) ds dt = 0, \quad j = 1, \ldots, J-1,$$

and where c_0 is defined uniquely, up to a sign change, by the property

$$||\phi_J|| = 1.$$

We can show that, for each $J \geq 1$, given $\phi_1, \ldots, \phi_{J-1}$, the function ϕ_J is a linear combination of J explicitly defined functions $(K^1(b), \ldots, K^p(b))$ and is unique up to a sign change. Therefore, for each J, the space generated by $\phi_1, \ldots, \phi_J$ is the same as the space generated by $K^1(b), \ldots, K^J(b)$.

In addition, if we define $\gamma_1, \ldots, \gamma_J$ to be the sequence $\omega_1, \ldots, \omega_J$ that minimises

$$\mathrm{E}\left[\int_{\mathcal{T}} \mathrm{E}\big[x(t) - \mathrm{E}\left[X(t)\right]\big] b(t) dt - \sum_{j=1}^{J} \omega_j \int_{\mathcal{T}} \mathrm{E}\big[x(t) - \mathrm{E}\left[X(t)\right]\big] K^j(b)(t) dt \right]^2,$$

then the slope function approximation b_J at (6.3) has two equivalent expressions:

$$b_J(t) = \sum_{j=1}^{J} \gamma_j K^j(b)(t) = \sum_{j=1}^{J} \beta_j \phi_j(t).$$

Stabilised algorithm for empirical APLS

In order to estimate $K^1(b)(t), \ldots, K^J(b)(t)$, we first estimate $K^1(b)(t) = \mathrm{Cov}\big[Y, X^c(t)\big]$. Next, we estimate $K^{j+1}(b)(t)$ by

$$\hat{K}^{j+1}(b)(t) = \int_{\mathcal{T}} \hat{K}^j(b)(s) \hat{k}(s,t) ds,$$

where $\hat{k}(s,t)$ is the estimator for the covariance function $\mathrm{Cov}\big[X(s), X(t)\big]$.

Finally, after obtaining $K^1(b)(t), \ldots, K^J(b)(t)$, we define $\hat{\gamma}_1, \ldots, \hat{\gamma}_J$ to be the values of

$\hat{\omega}_1, \ldots, \hat{\omega}_J$ which minimises

$$U_J(\omega_1, \ldots, \omega_J) = \frac{1}{N} \sum_{i=1}^{N} \left\{ y_i - \bar{y} - \sum_{j=1}^{J} \omega_j \int_{\mathcal{T}} (x_i(t) - \bar{x}(t)) \hat{K}^j(b)(t) dt \right\}.$$

As pointed out above, the non-orthonormal sequence $K^1(b), \ldots, K^J(b)$ generate the same space as that obtained by the orthonormal sequence $\phi_1, \ldots, \phi_J$. Therefore, we can transform our estimated sequence $\hat{K}^1(b), \ldots, \hat{K}^J(b)$ into an orthonormal sequence $\hat{\phi}_1, \ldots, \hat{\phi}_J$ using the modified Gram-Schmidt algorithm (Lange, 2010) and use it to estimate the regression coefficient function b_J.

[1] Obtain $\hat{K}^1(b)(t)$, i.e. estimate $\mathrm{Cov}\big[Y, X^c(t)\big]$.

[2] Obtain $\hat{k}(s,t)$, i.e. estimate $\mathrm{Cov}\big[X(s), X(t)\big]$.

[3] Calculate $\hat{K}^{j+1}(b)(t) = \int_{\mathcal{T}} \hat{K}^j(b)(s)\hat{k}(s,t)ds$ for $1 \leq j \leq J-1$.

[4] Transform $\hat{K}^1(b), \ldots, \hat{K}^J(b)$ into an orthonormal sequence $\hat{\phi}_1, \ldots, \hat{\phi}_J$ using the modified Gram-Schmidt algorithm.

[5] Find $\hat{\beta}_1, \ldots, \hat{\beta}_J$ by solving

$$\hat{\beta}_1, \ldots, \hat{\beta}_J = \underset{\nu_1, \ldots, \nu_J}{\mathrm{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left\{ y_i - \bar{y} - \sum_{j=1}^{J} \nu_j \int_{\mathcal{T}} \big(x_i(t) - \bar{x}(t)\big) \phi_j(t) dt \right\}^2.$$

[6] Estimate g by

$$\hat{g}_J(x) = \bar{y} + \sum_{j=1}^{J} \hat{\beta}_j \int_{\mathcal{T}} \big(x(t) - \bar{x}(t)\big) \hat{\phi}_j(t) dt.$$

When the sampling points of curves are dense and observed at the same locations for all subjects, we can estimate $K^1(b)(t)$ by

$$\hat{K}^1(b)(t) = \frac{1}{N} \sum_{i=1}^{N} \{x_i(t) - \bar{x}(t)\}\{y_i - \bar{y}\}. \tag{6.7}$$

and $k(s,t)$ by

$$\hat{k}(s,t) = \frac{1}{N} \sum_{i=1}^{N} \{x_i(s) - \bar{x}(s)\}\{x_i(t) - \bar{x}(t)\}. \tag{6.8}$$

6.1.1 Simulation study

Our first simulation exercise is to evaluate predictions provided by FPLSR and FPCR employing a functional linear regression model which includes only one functional covariate, $X(t)$, for explaining a scalar response Y:

$$Y_i = \alpha_0 + \int \beta(t) X_i(t) dt + \varepsilon_i, \quad i = 1, \ldots, N, \tag{6.9}$$

where $\varepsilon_i \sim N(0, \sigma_\varepsilon^2)$.

In addition, we have chosen σ_ε^2 in such a way that the signal-to-noise ratio (SNR) is $3, 10$ or 100.

We simulated a functional covariate which follows a Gaussian process for each individual. Basically, we generated samples from a multivariate normal distribution where the mean function was

$$\mathrm{E}\left[X(t)\right] = \sin(c\ t) + t\cos(c\ t),$$

where $c = 10$ in our example.

The covariance function we used was

$$\mathrm{Cov}\left[X(t_i), X(t_j)\right] = \exp\Big\{ -\frac{1}{\gamma}(t_i - t_j)^2 \Big\}, \quad t_i, t_j = 0.01, 0.02, \ldots, 1.00,$$

as we wish to simulate 100 points of the time domain.

We have analysed different values for γ. Figure 6.1 illustrates 40 simulated realisations of the functional predictor for $\gamma = 0.1$, $\gamma = 0.01$ and $\gamma = 0.001$.

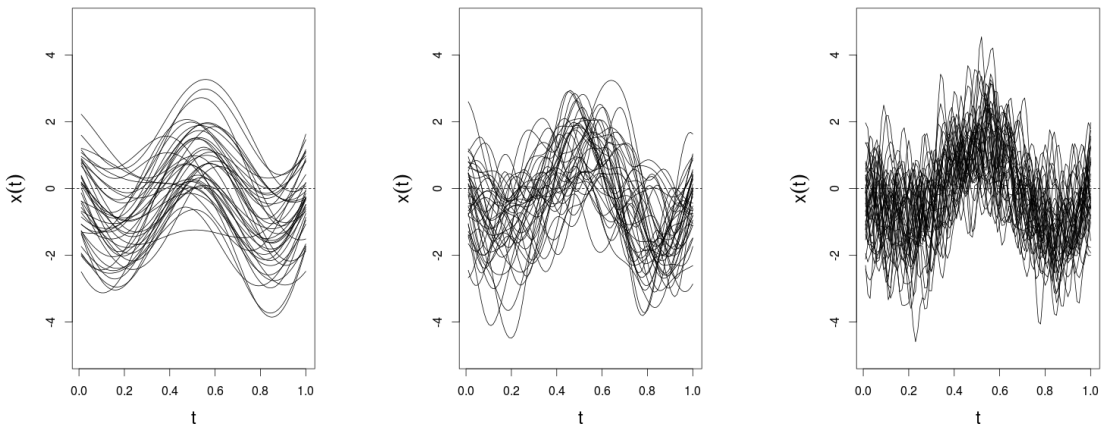

Figure 6.1: Plots of 40 realisations of the simulated functional covariate using $\gamma = 0.1$ (left), $\gamma = 0.01$ (middle) and $\gamma = 0.001$ (right).

The regression coefficient function $\beta(t)$ was simulated in the same way but always using $\gamma = 0.1$. Figure 6.2 shows one realisation of it.

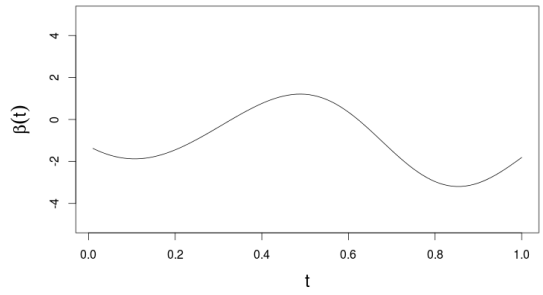

Figure 6.2: Plots of one realisation of simulated function $\beta(t)$ using $\gamma = 0.1$.

We have simulated training sets with different sample sizes ($N = 20,\ 60$, or 200) and test sets whose sample sizes are equal to 20. Next, we estimated regression equation (6.9) for each N. The estimated model was then used to predict the 20 observations of the test set. We repeated this procedure 100 times.

Let us first analyse the results for the case where the training sample size is $N = 60$. The results of this case are showed in illustrated in Figure 6.4. When $\gamma = 0.01$ and SNR = 10, FPLSR and FPCR obtain quite similar results when employing a large number of components; however, for one or two components, FPLSR performs much better than FPCR. For example, when employing two components, FPLSR obtains a prediction RMSE of 0.1696, while FPCR obtains a much worse result: 0.3676.

Also in the case of $N = 60$, observe that, for very smooth curves ($\gamma = 0.1$), FPCR has very similar prediction performance in comparison to FPLSR when using three or more components, but the latter is much worse than the former if they work with less components. For rougher predictor curves ($\gamma = 0.001$), FPCR needs too many components to try to achieve the same performance that FPLSR obtains by employing only 2–4 components.

By still analysing different values of γ, we can observe another interesting finding: the rougher are the predictor curves, the better is the prediction performance of both methods. In other words, both methods are able to extract more information from rough curves than from very smooth curves. It makes sense, since observations of very smooth curves are highly correlated and therefore most of them do not make any relevant contribution to explain the variation in the response Y. At this point, a problem appears: for real datasets, we can only observe discrete data points and since we do not know how is the true predictor curve, we do not know whether the roughness of the observed data

is explained by the curve structure itself or by the measurement error. Therefore, this discussion might lead to a potential new topic to be studied, where we could apply, for example, the measurement error model.

Still for the case of sample size $N = 60$, we can observe that, essentially for rough curves, the higher the SNR value, the bigger is the advantage of FPLSR in comparison with FPCR (in terms of providing good predictions employing as less components as possible). This reveals that FPLSR, which takes into account the interaction between the response Y and the covariate $X(t)$ when constructing components, is indeed better than FPCR when $X(t)$ helps to explain most of the variation y.

Finally, analysing different sample sizes, we can see that both methods perform slightly better as the sample size increases (see Figures 6.3 – 6.5). However, the above comments used to compare both methods for case of sample size 60 can also be used for the other cases.

In conclusion, essentially for rougher predictor curves, we can say that FPLSR seems to be preferable to FPCR as the former leads to a more parsimonious model than the latter when trying to provide the best (or at least nearly the best) prediction results.

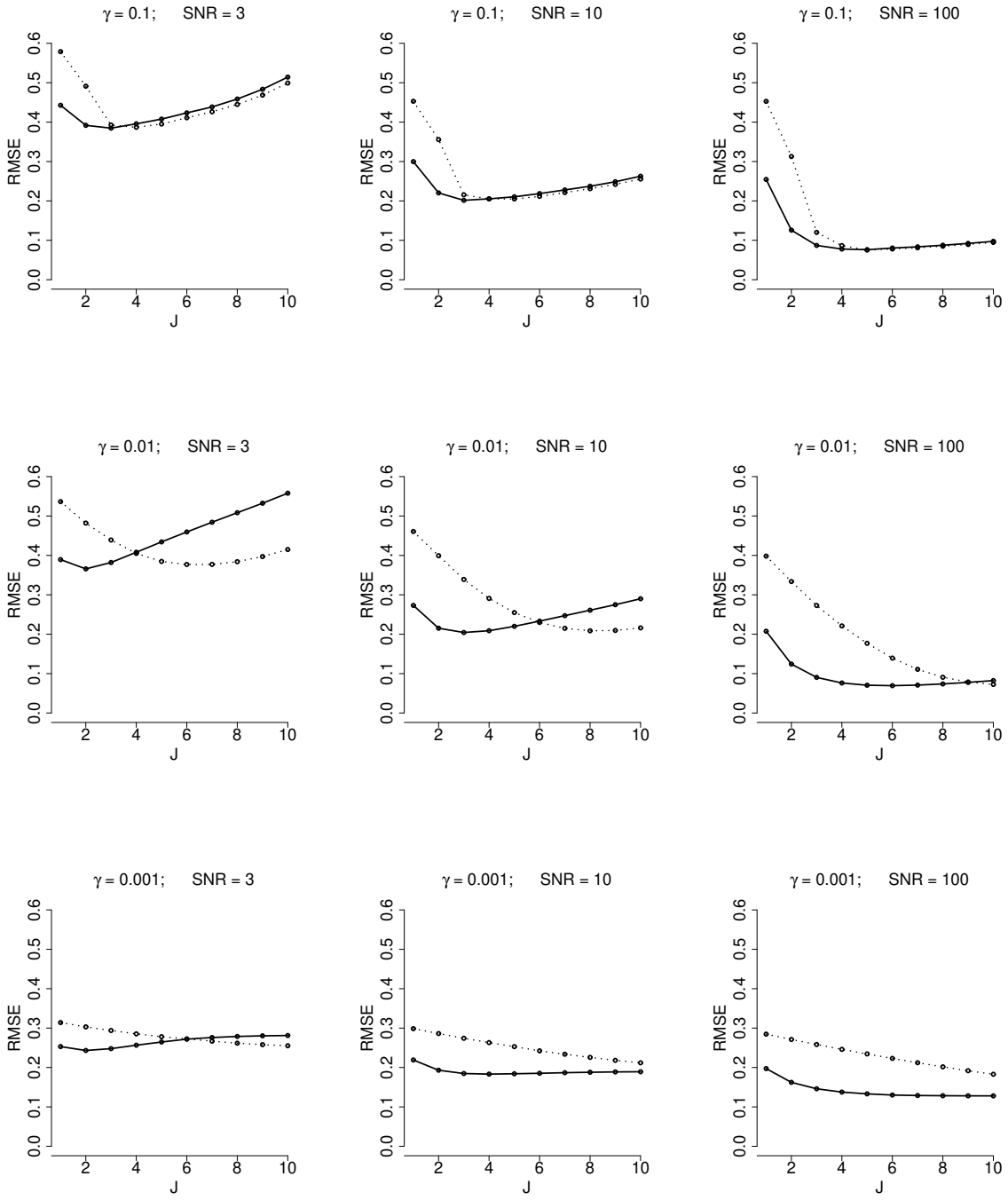

Figure 6.3: Prediction RMSE of FPLSR (solid lines) and FPCR (dotted lines) per number of components. Training sets have size $N = 20$.

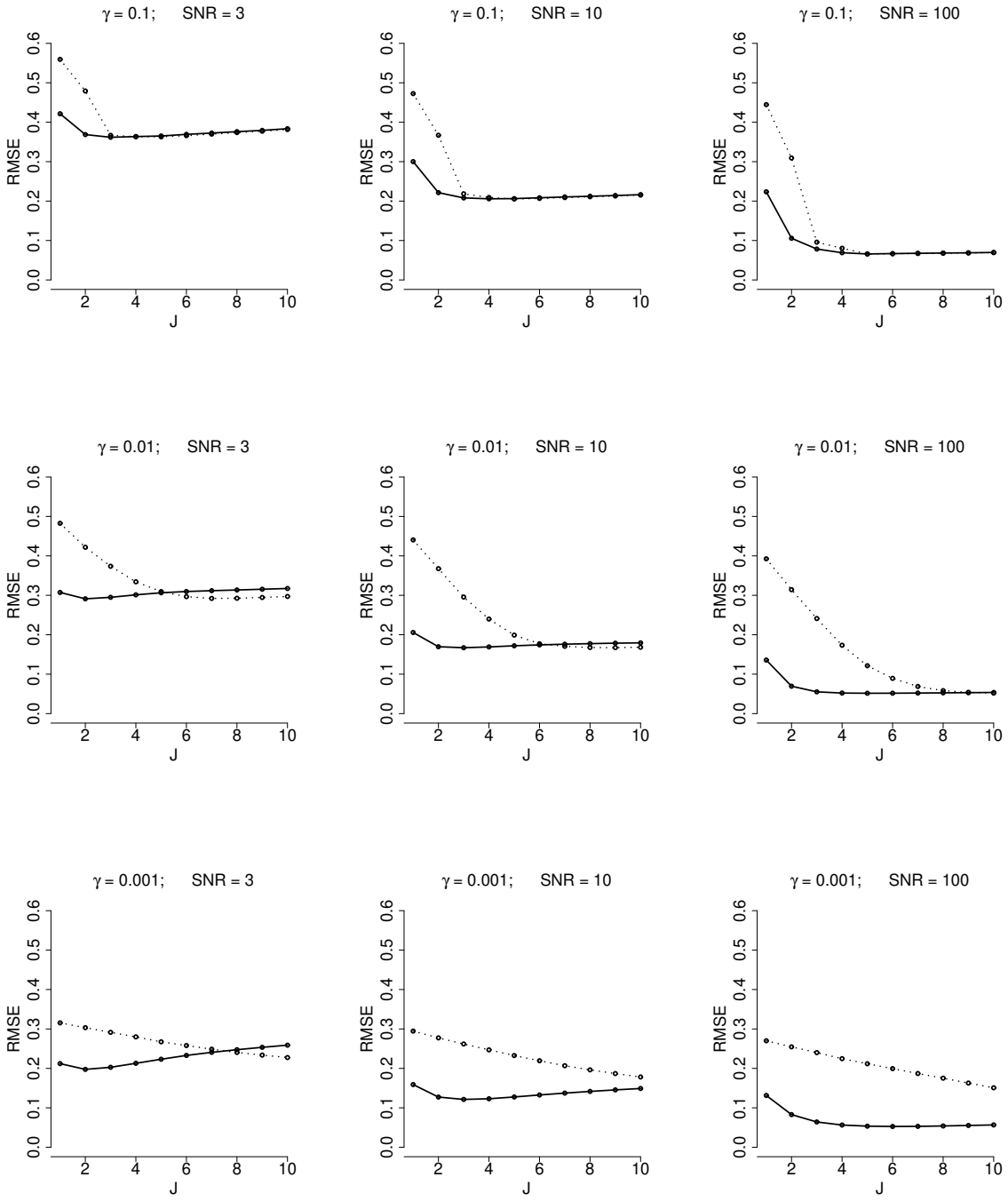

Figure 6.4: Prediction RMSE of FPLSR (solid lines) and FPCR (dotted lines) per number of components. Training sets have size $N = 60$.

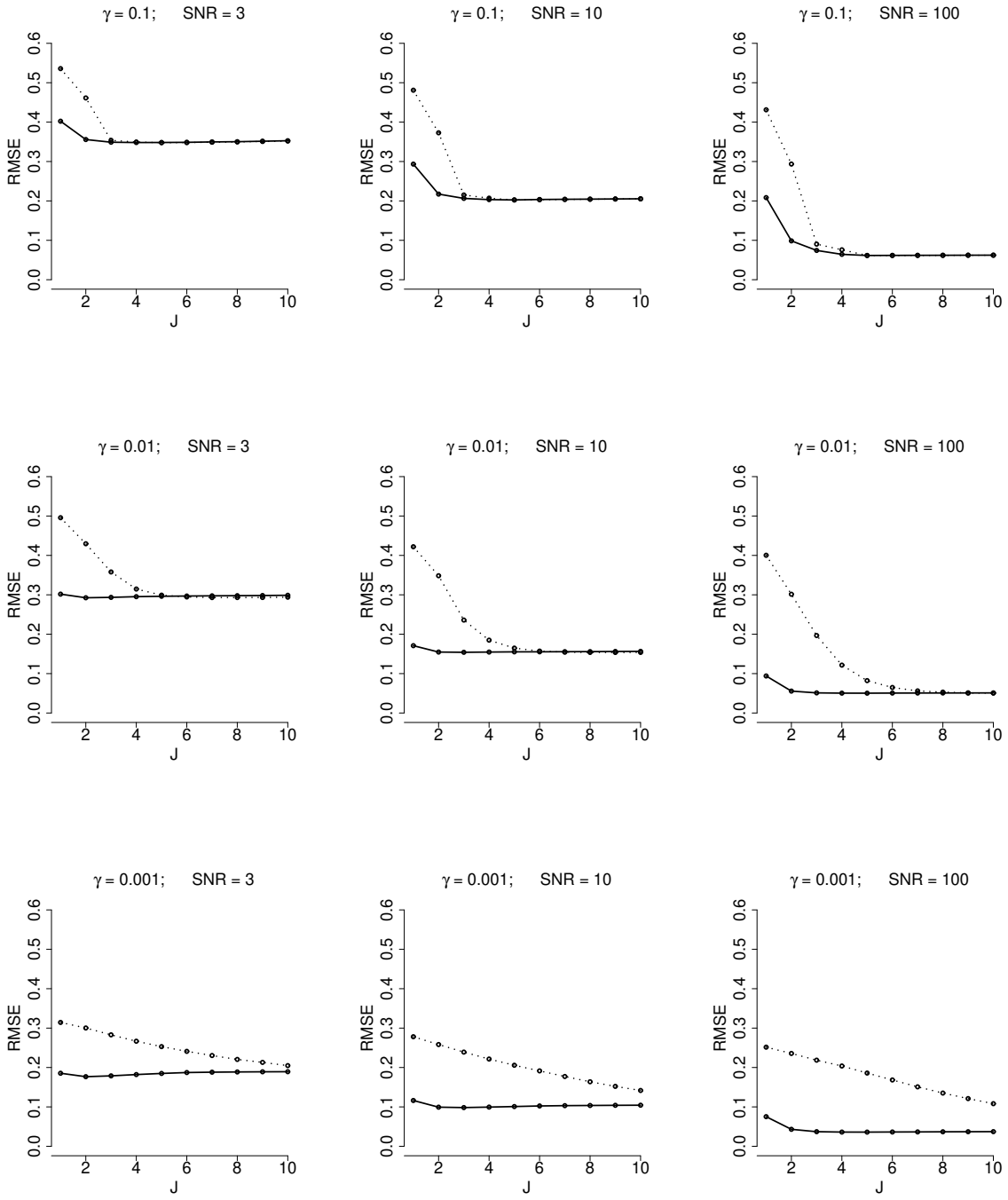

Figure 6.5: Prediction RMSE of FPLSR (solid lines) and FPCR (dotted lines) per number of components. Training sets have size $N = 200$.

6.2 Multiple Functional PLS regression

We can extend the functional regression model (6.1) to the case involving $M > 1$ functional covariates, so that

$$Y_i = a + \sum_{l=1}^{M} \int_{\mathcal{T}} b_l(t) X_{l,i}^c(t) dt + \varepsilon_i, \qquad i = 1, \ldots, N.$$

Analogously to the case where the functional covariate is univariate, this regression equation can be solved by expanding the M-variate $\boldsymbol{X}(\cdot) = \big(X_1(\cdot), \ldots, X_M(\cdot)\big)^\top$ in terms of multivariate FPC basis or multivariate FPLS basis, which lead, respectively, to the multivariate FPCR (MFPCR) and to the multiple FPLSR (MFPLSR).

We propose to adapt the algorithm for empirical APLS, described in the previous section, to the case involving the multivariate functional covariate $\boldsymbol{X}(\cdot) = \big(X_1(\cdot), \ldots, X_M(\cdot)\big)^\top$. Although each element X_l can be observed in a different domain $\mathcal{T}_l$, they can be shifted to have the same domain $\mathcal{T} = [0, 1]$. Therefore, without loss of generality, we assume a common domain.

Let us describe the steps of the adapted algorithm. The implementation consists in concatenating the M functional covariates into a unique long function. Therefore, in the below algorithm, $\boldsymbol{X}$ (and the corresponding observations $\boldsymbol{x}$) should be interpreted as the concatenated function and proceed as if it were a unique functional covariate defined on $\mathcal{T} = [0, M]$. Similarly, basis functions $\hat{\phi}_j$ are also defined on $\mathcal{T} = [0, M]$.

The difference is in the estimation of $\mathrm{Cov}\big[\boldsymbol{X}(s), \boldsymbol{X}(t)\big]$. We suggest using the MGP method, proposed in Chapter 5, to estimate the auto- and cross-covariance functions. The adapted algorithm has the following steps:

[1] Obtain $\hat{K}^1(b)(t)$, i.e. estimate $\mathrm{Cov}\big[Y, \boldsymbol{X}^c(t)\big]$.

[2] Obtain $\hat{k}(s,t)$, i.e. estimate $\mathrm{Cov}\big[\boldsymbol{X}(s), \boldsymbol{X}(t)\big]$.

[3] Calculate $\hat{K}^{j+1}(b)(t) = \displaystyle\int_{\mathcal{T}} \hat{K}^j(b)(s)\hat{k}(s,t)ds$ for $1 \leq j \leq J-1$.

[4] Transform $\hat{K}^1(b), \ldots, \hat{K}^J(b)$ into an orthonormal sequence $\hat{\phi}_1, \ldots, \hat{\phi}_J$ using the modified Gram-Schmidt algorithm.

[5] Find $\hat{\beta}_1, \ldots, \hat{\beta}_J$ by solving

$$\hat{\beta}_1, \ldots, \hat{\beta}_J = \underset{\nu_1, \ldots, \nu_J}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left\{ y_i - \bar{y} - \sum_{j=1}^{J} \nu_j \int_{\mathcal{T}} \big(\boldsymbol{x}_i(t) - \bar{\boldsymbol{x}}(t)\big)\phi_j(t)dt \right\}^2.$$

[6] Estimate g by

$$\hat{g}_J(\boldsymbol{x}) = \bar{y} + \sum_{j=1}^{J} \hat{\beta}_j \int_{\mathcal{T}} \big(\boldsymbol{x}(t) - \bar{\boldsymbol{x}}(t)\big)\hat{\phi}_j(t)dt.$$

When the data are not observed regularly at the same locations for all individual functions, the empirical estimators (6.7) and (6.8) cannot be used. Therefore, we suggest using functional singular component analysis (Yang *et al.*, 2011) to estimate $K^1(b)(t)$, obtaining a smooth cross-covariance between the scalar response and the concatenated functional predictor. The covariance function $k(s,t)$ can be estimated nonparametrically (Yao *et al.*, 2005) or by using our proposed MGP model in Chapter 5.

6.2.1 Simulation study

We want to predict a scalar response Y given multiple functional variables $X_1, \ldots, X_M$, for $M = 2, 5$, and 10.

Similarly as we do in eq. (5.3), we simulate M-variate functional data $\boldsymbol{X}(t) = \big(X_1(t), \ldots, X_M(t)\big)^\top$ based on a truncated multivariate Karhunen-Loève expansion representation

$$\boldsymbol{X}(t) = \sum_{j=1}^{J} \xi_j \boldsymbol{\phi}_j(t), \qquad t \in [0,1]\,. \tag{6.10}$$

The multivariate basis functions $\boldsymbol{\phi}_j = (\phi_j^{(1)}, \phi_j^{(2)}, \ldots, \phi_j^{(M)})^\top \in \mathbb{R}^M$ are constructed from the first 15 orthonormal eigenfunctions of the Wiener process on the domain $\mathcal{T} = [0, M]$, which are then split into M parts. All the parts are shifted to have common domain $\mathcal{T} = [0,1]$. The scores ξ_j are simulated independently from a Gaussian distribution with zero mean and variance decreasing linearly towards 0. The elements of $\boldsymbol{X}$, i.e. $X_l,\ l = 1, \ldots, M$, are sampled on an equispaced grid of $n_1 = n_2 = \cdots = n_M = 100$ sampling points.

Figure 6.6 shows bivariate noise curves generated via (6.10) using $J = 15$ components. The curves have SNR = 100.

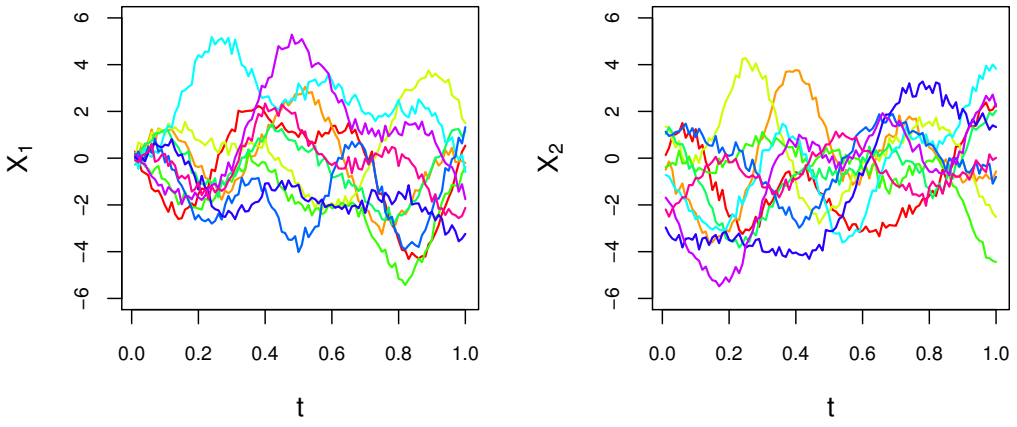

Figure 6.6: Bivariate functional data generated by using Wiener processes.

Then we generate a fictitious scalar response Y from

$$Y_i = \sum_{l=1}^{M} \int \beta_l(t) X_{il}(t) dt + \varepsilon_i, \quad i = 1, \ldots, N, \tag{6.11}$$

where σ_ε^2 is chosen in such a way that SNR $= 100$.

The functional coefficients β_l are constructed in the following way. We first estimate the leading 12 multivariate functional principal components (MFPCs) of $\boldsymbol{X}(t)$ (i.e. $\hat{\boldsymbol{\phi}}_1^{PC}(t), \ldots, \hat{\boldsymbol{\phi}}_{12}^{PC}(t)$) and extract the parts corresponding to the M elements, namely $\hat{\phi}_{l,1}^{PC}(t), \ldots, \hat{\phi}_{l,12}^{PC}(t)$, $l = 1, \ldots, M$. Next, we generate y using (6.11) by assuming $\beta_l(t) = \sum_{k=1}^{K} a_{l,k} \hat{\phi}_{l,k}^{PC}(t)$, where we consider four different cases:

(i) $a_{l,k} = \text{sign}(k) \cdot 1\{1 \leq k \leq 3\}$;

(ii) $a_{l,k} = \text{sign}(k) \cdot 1\{4 \leq k \leq 6\}$;

(iii) $a_{l,k} = \text{sign}(k) \cdot 1\{7 \leq k \leq 9\}$;

(iv) $a_{l,k} = \text{sign}(k) \cdot 1\{10 \leq k \leq 12\}$,

where $\text{sign}(k) = -1$ or 1 with equal probabilities. In this way, case (i) represents a setting where the interaction between y and $\boldsymbol{X}$ is represented by the first three MFPC basis functions. Therefore, in this case we expect to see good performance of MFPCR by using a small number of components. However, as we go the cases (ii), (iii), and (iv), in this order, MFPCR should find more and more difficult to explain the interaction between Y and $\boldsymbol{X}$ by using few components.

We simulate $N = 200$ observations $\big((y_1, \boldsymbol{x}_1), \ldots, (y_N, \boldsymbol{x}_N)\big)$. Next, we randomly choose a training sample of size $N_{\text{train}} \in \{30, 100\}$ and a test sample of size $N_{\text{test}} = N - N_{\text{train}}$ and

evaluate the predictions of y values in the test set given the model with $J = 1, 2, \ldots,$ or 12 components estimated by using only the training data. We use the prediction RSMSE error given by

$$\text{RSMSE} = \left[N_{\text{test}}^{-1} \sum_{i=1}^{N_{\text{test}}} (\hat{y}_i - y_i)^2 / \text{Var}\,[y] \right]^{1/2}.$$

The standardisation is required to compare settings in which the scale of y is different. In addition, RSMSE is approximately 1 when we use the sample mean of the training observations y as the predictor $\hat{y}$.

This procedure was repeated 100 times, that is, we simulate 100 datasets and obtain the prediction error for each. The boxplots of these prediction errors are shown in Figures 6.7, 6.8, and 6.9, which illustrate the cases of $M = 2, 5,$ and 10 functional predictors, respectively.

The results indicate that MFPLSR needs fewer components than MFPCR does to capture the interaction between the scalar response and the functional predictors, especially when we go from case (i) to case (iv). The results are very similar for $M = 2, 5,$ and 10.

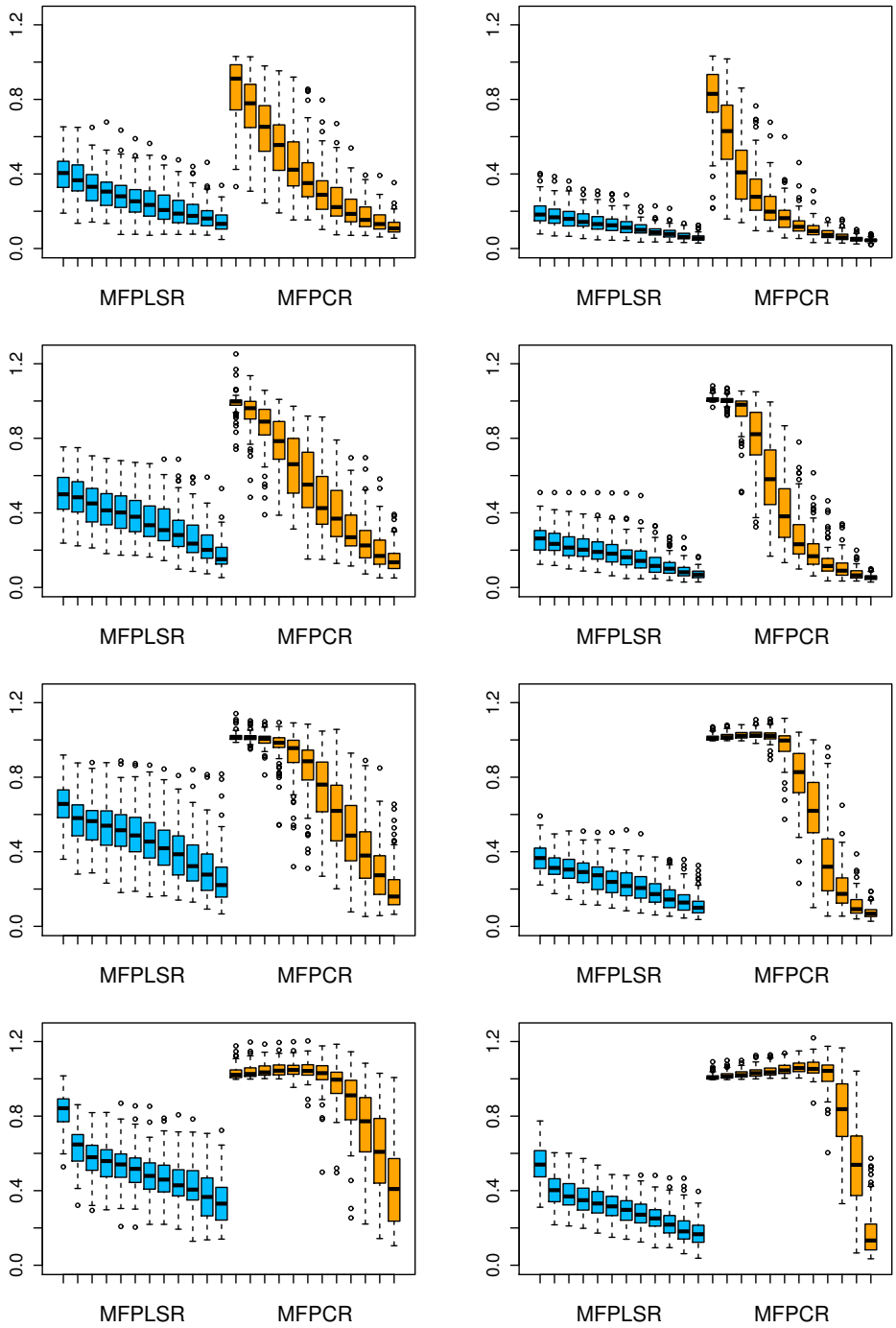

Figure 6.7: Boxplots of the prediction RSMSE using MFPLSR (first group of 12 boxes) and MFPCR (last group of 12 boxes) calculated from 100 datasets of sizes $N_{\text{train}} = 30$ (first column) and $N_{\text{train}} = 100$ (second column), all generated from **bivariate** functional data simulated using eq. (6.10), where γ_j are eigenfunctions of Wiener processes. The functional coefficients β_l corresponding to the cases (i),(ii),(iii), and (iv), are shown in rows 1,2,3, and 4, respectively. From left to right, each group of 12 boxplots corresponds to the cases where the methods used $J = 1, \ldots, 12$ components.

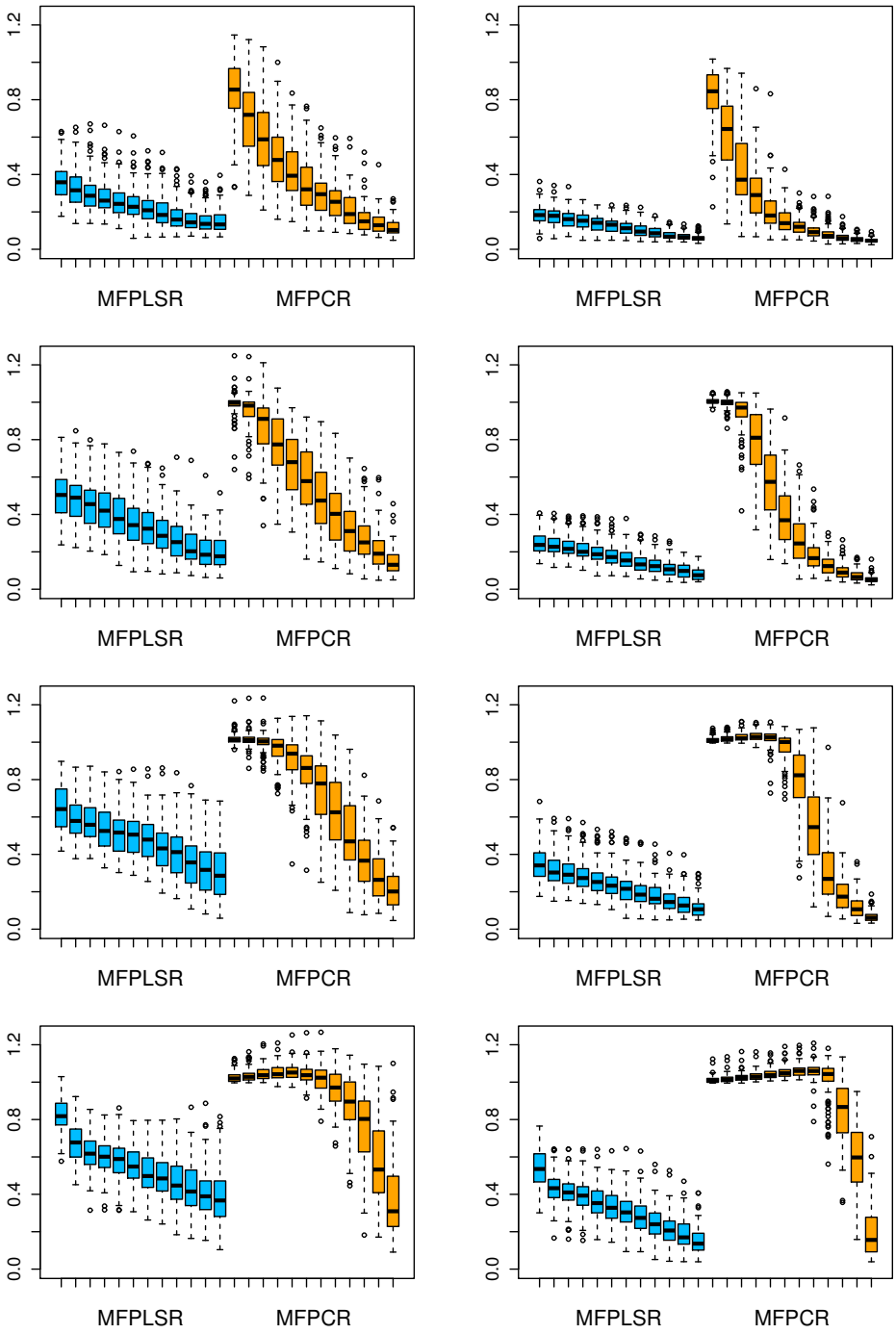

Figure 6.8: Boxplots of the prediction RSMSE using MFPLSR (first group of 12 boxes) and MFPCR (last group of 12 boxes) calculated from 100 datasets of sizes $N_{\text{train}} = 30$ (first column) and $N_{\text{train}} = 100$ (second column), all generated from **5-variate** functional data simulated using eq. (6.10), where γ_j are eigenfunctions of Wiener processes. The functional coefficients β_l corresponding to the cases (i),(ii),(iii), and (iv), are shown in rows 1,2,3, and 4, respectively. From left to right, each group of 12 boxplots corresponds to the cases where the methods used $J = 1, \ldots, 12$ components.

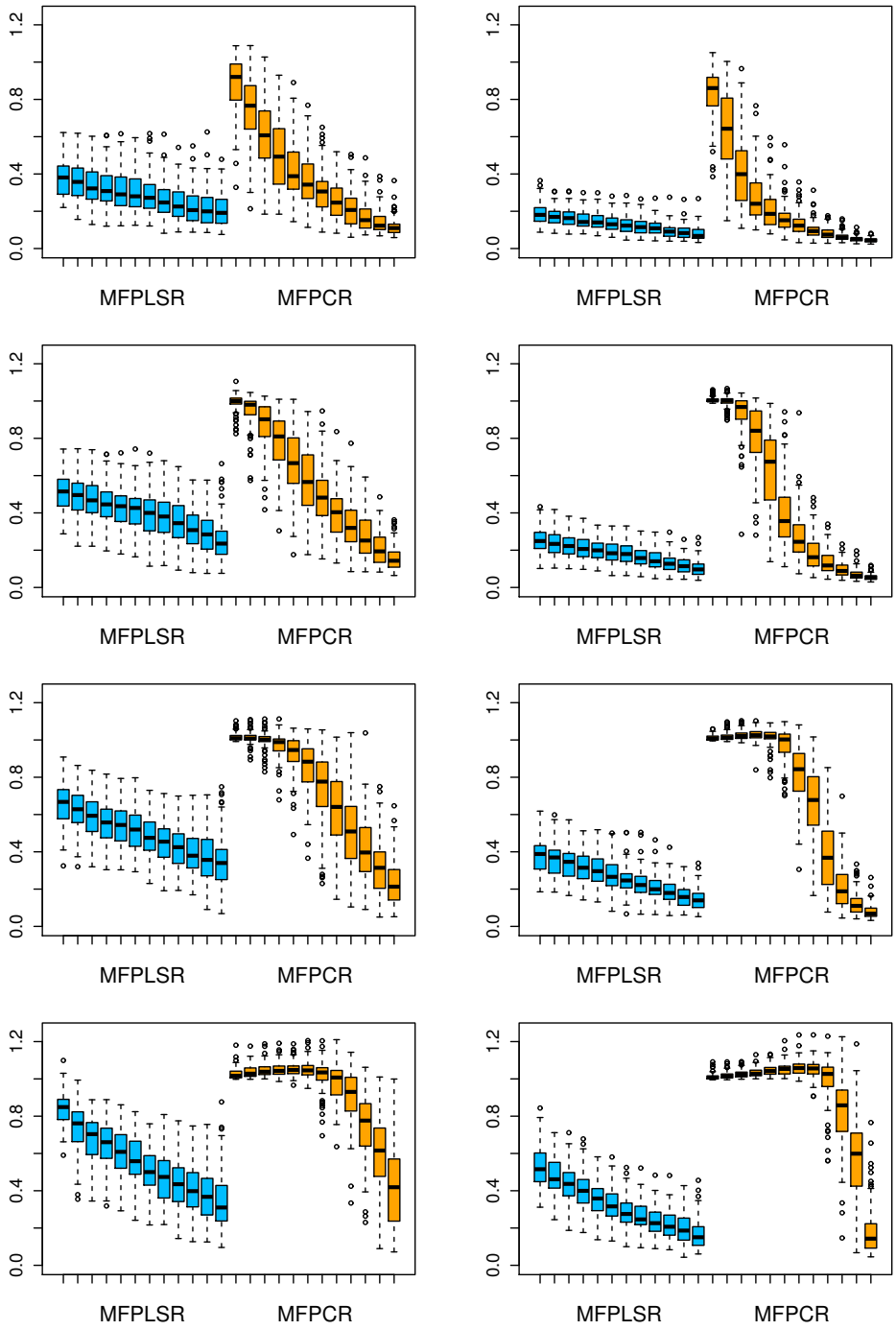

Figure 6.9: Boxplots of the prediction RSMSE using MFPLSR (first group of 12 boxes) and MFPCR (last group of 12 boxes) calculated from 100 datasets of sizes $N_{\text{train}} = 30$ (first column) and $N_{\text{train}} = 100$ (second column), all generated from **10-variate** functional data simulated using eq. (6.10), where γ_j are eigenfunctions of Wiener processes. The functional coefficients β_l corresponding to the cases (i),(ii),(iii), and (iv), are shown in rows 1,2,3, and 4, respectively. From left to right, each group of 12 boxplots corresponds to the cases where the methods used $J = 1, \ldots, 12$ components.

6.3 Implementation

In this section, we make a few remarks about some steps in the MFPLSR algorithm. Implementation details related to the covariance structure estimation are discussed in Section 5.3.

The MFPLSR algorithm, shown in Section 6.2, requires the calculation of integrals to obtain the estimates of $K^1(b), \ldots, K^p(b)$ and of $g_J(\boldsymbol{x})$. We used numerical integration by trapezoidal rule. This is also used in the well developed R package fda (Ramsay *et al.*, 2018).

The modified Gram-Schmidt algorithm, used to transform $\hat{K}^1(b), \ldots, \hat{K}^J(b)$ into an orthonormal sequence $\hat{\phi}_1, \ldots, \hat{\phi}_J$, can be implemented by using the R package pracma Borchers (2018). However, we have written an inline C++ version of this algorithm for efficiency purposes.

6.4 Conclusion

The numerical simulation study in Section 6.2 shows that our extension of the FPLSR model (Delaigle & Hall, 2012) to the case which involves multiple function-valued covariates can estimate more accurately the slope functions than MFPCR does. Therefore, it can use fewer components than MFPCR does to provide excellent prediction results for the scalar response. MFPLSR can be applied to many applications in which the data are sparsely, irregularly sampled. In addition, it can be more appropriate than variable selection methods for functional covariates especially when these functional covariates are cross-correlated.

Chapter 7

Conclusions and future work

In this chapter, we discuss the main contributions and conclusions in Section 7.1 and future work in Section 7.2.

7.1 Contributions and conclusions

GPR models enable us to model a wide class of nonlinear functions, including those defined on multidimensional domains, and to easily account for uncertainty on predictions. The two major difficulties are the potentially high computational costs and the choice of the covariance function family. We have investigated approximate implementation methods that reduce dramatically the computational costs while keeping good prediction ability. The performance of these approximations mainly depends on how smooth the sample paths are, which is determined by the decay of eigenvalues of the covariance function. Although the subset size m in the Subset of Regressors and Subset of Data should be as large as possible, we have shown that we can choose a m significantly smaller than the sample size n when the sample paths are smooth. On the other hand, if the sample paths are rough, we should not use too small a value of m.

We have also shown that the decomposition of GPs, in the spirit of FPCA, is useful to describe the main modes of variation in the data by using only leading components, namely the eigenfunctions or eigensurfaces. The visualisation of eigensurfaces can indicate interactions between coordinate directions, revealing nonseparability features in the covariance function.

Whereas nonparametric models for the covariance function are flexible but difficult to estimate in multidimensional domains due to the curse of dimensionality problems, parametric models can be easily estimated but their flexibility is limited by the choice of parametric covariance function, which is usually either stationary or separable. In Chapter

4, we propose to use a flexible, convolution-based approach which allows for nonstationarity and nonseparability, crucial properties to achieve good fit of the covariance function, extract the most important modes of variation in the data and obtain better estimates of uncertainty in predictions. This approach is readily applied to multidimensional domains.

The nonstationarity is defined by the $Q \times Q$ varying anisotropy matrix $\boldsymbol{\Sigma}(\boldsymbol{t})$ and the standard deviation $\sigma(\boldsymbol{t})$, both varying along $\boldsymbol{t} \in \mathcal{T} \subset \mathbb{R}^Q$ or along $\boldsymbol{\tau} \in \mathcal{T}^* \subset \mathbb{R}^{Q^*}$, where $Q^* \leq Q$. The nonseparability is achieved by allowing non-zero off-diagonal entries in $\boldsymbol{\Sigma}(\boldsymbol{t})$.

The unconstrained estimation of the parameters enables us to model them as a function of time (or spatial location) easily. They can be further modelled as a function of time (or spatially) dependent covariates and even additional covariates which bring information for each subject. In any of these cases, the function can be represented by a variety of basis functions, among which we have found B-splines basis very suitable for ensuring smoothness and being flexible.

In particular, our proposed spherical parametrisation for $\boldsymbol{\Sigma}_{Q\times Q}(\boldsymbol{t})$, which allows us to easily deal with input dimensions higher than two, is specified by a decomposition whose parameters have statistical interpretation. This is important for many applications. For example, if random trajectories fluctuate over time more quickly in the winter than in other seasons, then time (or a time-dependent covariate) seems a natural input for the corresponding length-scale parameter. The spherical parametrisation should also be helpful if we wish to conduct inference on those parameters.

Chapter 5 is dedicated to the modelling of the covariance structure of multivariate function-valued processes $\boldsymbol{X}(\cdot) = \big(X_1(\cdot), \ldots, X_M(\cdot)\big)^\top$. We extend a convolution-based model proposed for the bivariate case to the multivariate case. In this approach, each element $X_l(\cdot)$ has covariance function with different parameter values and can be defined on different domains. Finally, the cross-covariance functions are explicitly modelled and can be important for predictions and analysis of multivariate functional data. Via simulation studies we have shown that predictions of individual elements $X_l(\cdot)$ can be improved by using the available information of other elements at nearby locations where $X_l(\cdot)$ has missing data. The numerical results have also shown that MGP can provide similar or better predictions than MFPCA does using the same number of sampling points. Finally, the application to human fertility data gives interesting insights by visualising the eigensurfaces, which can reveal nonseparability features of the covariance structure.

The estimation of the covariance structure of $\boldsymbol{X}(\cdot)$ is used as a building block in Chapter 6, where we discuss scalar-on-functions regression models. We extend the PLS algorithm for a scalar response and a function-valued covariate (Delaigle & Hall, 2012) to the case involving multiple function-valued covariates and call this the MFPLSR model.

This is especially useful when the covariates are not regularly sampled, which makes impossible the use of the empirical covariance estimator, and not densely sampled, which makes nonparametric estimation difficult. Numerical simulation studies show that MFPLSR can provide better predictions for a scalar response variable Y using less components than MFPCR does, indicating that it is worth to consider the linear dependence between Y and the functional predictors when constructing the basis functions to represent the slope functions. This is particularly important when Y is explained by later FPCs of the functional covariates.

We should also highlight that our modelling framework for both uni- and multivariate function-valued processes can easily deal with sparsely sampled functional data and measurement error.

7.2 Future work

The decomposition of GPs may be important for developing efficient approximation for big data, non-Gaussian data (Wang & Shi, 2014) and heavy-tailed data (Shah *et al.*, 2014; Wang *et al.*, 2017; Cao *et al.*, 2018). For non-Gaussian and heavy-tailed data, the decomposition might be used instead of their predictive distributions which are usually complicated. It can also be important for further analysis of scalar-on-functions or function-on-functions regression models, where we try to reduce the dimension of data by using a small number of components.

Another topic for future research is considering varying kernels in MGP similarly as we have seen for univariate GP. One major difficulty will be the potentially large number of hyperparameters and the need for a more parsimonious model. We have assumed that each function-valued process is a sum of two convolved GPs. One way to deal with nonstationarity is by using an additional GP with a nonstationary covariance function, such as the linear covariance function (Shi & Choi, 2011), and assuming some constant parameters (e.g. constant anisotropy matrices).

Further investigation can be made on clustering and classification methods for functional data. In clustering of functional data, taking mean functions as cluster centres may not be adequate when the covariance structure is important to distinguish clusters (Wang *et al.*, 2016). Therefore, learning the covariance structure is crucial and (semi)parametric models may be used when nonparametric estimation is complicated. Classification of functional data are usually based on functional regression models which have a binary response variable and functional predictors (e.g. functional generalized linear models in Müller *et al.* (2005), Wang *et al.* (2016) and references therein). In this context, methods which take into account the covariance between the response and the functional predictors

when constructing basis functions, such as FPLS (Delaigle & Hall, 2012), can be preferable and may use a covariance structure of the multiple functional predictors estimated by a (semi)parametric model.

Bibliography

Abrahamsen, P. 1997 A review of Gaussian random fields and correlation functions. *Tech. Rep.*. Norwegian Computing Center, Oslo, Norway.

Adler, R. J. & Taylor, J. E. 2007 *Random fields and geometry*. Springer, New York.

Allen, G. I., Grosenick, L. & Taylor, J. 2014 A generalized least-square matrix decomposition. *Journal of the American Statistical Association* **109** (505), 145–159.

Aston, J. A., Pigoli, D., Tavakoli, S. *et al.* 2017 Tests for separability in nonparametric covariance operators of random surfaces. *The Annals of Statistics* **45** (4), 1431–1461.

Ba, S. & Joseph, V. R. 2012 Composite Gaussian process models for emulating expensive functions. *Ann. Appl. Stat.* **6** (4), 1838–1860.

Banerjee, S., Carlin, B. P. & Gelfand, A. E. 2015 *Hierarchical modeling and analysis for spatial data*, 2nd edn. CRC Press.

Basawa, I. V. & Prakasa Rao, B. L. S. 1980 *Statistical Inference for Stochastic Processes*. Academic Press.

Berrendero, J. R., Justel, A. & Svarc, M. 2011 Principal components for multivariate functional data. *Computational Statistics and Data Analysis* **55** (9), 2619–2634.

de Boor, C. 2001 *A Practical Guide to Splines*. New York: Springer.

Borchers, H. W. 2018 *pracma: Practical Numerical Math Functions*. R package version 2.2.2.

Bosq, D. 2000 *Linear Processes in Function Spaces: Theory and Applications*. Springer New York.

Boyle, P. & Frean, M. 2004 Dependent Gaussian processes. In *Advances in neural information processing systems*, pp. 217–224.

Calandra, R., Peters, J., Rasmussen, C. E. & Deisenroth, M. P. 2016 Manifold Gaussian processes for regression. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp. 3338–3345. IEEE.

Cao, C., Shi, J. Q. & Lee, Y. 2018 Robust functional regression model for marginal mean and subject-specific inferences. *Statistical Methods in Medical Research* **27** (11), 3236–3254.

Cappello, C., De Iaco, S. & Posa, D. 2018 Testing the type of non-separability and some classes of space-time covariance function models. *Stochastic Environmental Research and Risk Assessment* **32** (1), 17–35.

Carlin, B. P. & Louis, T. A. 2008 *Bayesian methods for data analysis*. CRC Press.

Chen, K., Delicado, P. & Müller, H.-G. 2017 Modelling function-valued stochastic processes, with applications to fertility dynamics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **79** (1), 177–196.

Chen, K. & Lynch, B. 2017 Weak separablility for two-way functional data: Concept and test. arXiv preprint arXiv:1703.10210.

Chen, K. & Müller, H.-G. 2012 Modeling repeated functional observations. *Journal of the American Statistical Association* **107** (500), 1599–1609.

Chiou, J.-M., Chen, Y.-T. & Yang, Y.-F. 2014 Multivariate functional principal component analysis: A normalization approach. *Statistica Sinica* pp. 1571–1596.

Chiou, J.-M. & Müller, H.-G. 2014 Linear manifold modelling of multivariate functional data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **76** (3), 605–626.

Constantinou, P., Kokoszka, P. & Reimherr, M. 2017 Testing separability of space-time functional processes. *Biometrika* **104** (2), 425–437.

Cressie, N. & Huang, H.-C. 1999 Classes of nonseparable, spatio-temporal stationary covariance functions. *Journal of the American Statistical Association* **94** (448), 1330–1339.

De Iaco, S., Myers, D. E. & Posa, D. 2002 Nonseparable space-time covariance models: some parametric families. *Mathematical Geology* **34** (1), 23–42.

Deisenroth, M. & Ng, J. W. 2015 Distributed Gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning* (ed. F. Bach & D. Blei), *Proceedings of Machine Learning Research*, vol. 37, pp. 1481–1490. Lille, France: PMLR.

Delaigle, A. & Hall, P. 2012 Methodology and theory for partial least squares applied to functional data. *The Annals of Statistics* **40** (1), 322–352.

Dunlop, M. M., Girolami, M. A., Stuart, A. M. & Teckentrup, A. L. 2018 How deep are deep Gaussian processes? *Journal of Machine Learning Research* **19** (54), 1–46.

Ecker, M. D. & Gelfand, A. E. 1999 Bayesian modeling and inference for geometrically anisotropic spatial data. *Mathematical Geology* **31** (1), 67–83.

Eddelbuettel, D. & Sanderson, C. 2014 Rcpparmadillo: Accelerating r with high-performance c++ linear algebra. *Computational Statistics and Data Analysis* **71**, 1054–1063.

Fan, J. & Gijbels, I. 1996 Local polynomial modelling and its applications. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Finley, A., Banerjee, S. & yvind Hjelle 2017 *MBA: Multilevel B-Spline Approximation*. R package version 0.0-9.

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F. & Hothorn, T. 2019 *mvtnorm: Multivariate Normal and t Distributions*. R package version 1.0-7.

Gneiting, T. 2002 Nonseparable, stationary covariance functions for space–time data. *Journal of the American Statistical Association* **97** (458), 590–600.

Gramacy, R. B. & Lee, H. K. H. 2008 Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association* **103** (483), 1119–1130.

Hall, P., Müller, H.-G. & Wang, J.-L. 2006 Properties of principal component methods for functional and longitudinal data analysis. *The annals of statistics* pp. 1493–1517.

Happ, C. 2018*a* funData: An S4 Class for Functional Data. R package version 1.2.

Happ, C. 2018*b* MFPCA: Multivariate Functional Principal Component Analysis for Data Observed on Different Dimensional Domains. R package version 1.2-3.

Happ, C. & Greven, S. 2018 Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association* pp. 1–11.

Higdon, D. 1998 A process-convolution approach to modelling temperatures in the north atlantic ocean. *Environmental and Ecological Statistics* **5** (2), 173–190.

Higdon, D. 2002 Space and space-time modeling using process convolutions. In *Quantitative methods for current environmental issues*, pp. 37–56. Springer.

Higdon, D., Swall, J. & Kern, J. 1999 Non-stationary spatial modeling. *Bayesian statistics* **6** (1), 761–768.

Horváth, L. & Kokoszka, P. 2012 *Inference for functional data with applications*. Springer Science & Business Media.

Human Fertility Database 2017 Max Planck Institute for Demographic Research (Germany) and Vienna Institute of Demography (Austria). [Online; accessed 17-March-2017].

Jidling, C., Wahlström, N., Wills, A. & Schön, T. B. 2017 Linearly constrained Gaussian processes. In *Advances in Neural Information Processing Systems 30* (ed. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett), pp. 1215–1224. Curran Associates, Inc.

Kanagawa, M., Hennig, P., Sejdinovic, D. & Sriperumbudur, B. K. 2018 Gaussian processes and kernel methods: A review on connections and equivalences. arXiv preprint arXiv:1807.02582.

Kokoszka, P. & Reimherr, M. 2017 *Introduction to functional data analysis*. CRC Press.

Lange, K. 2010 *Numerical analysis for statisticians*. Springer Science & Business Media.

Lawrence, N. D., Seeger, M. & Herbrich, R. 2003 Fast sparse Gaussian process methods: The Informative Vector Machine. In *Advances in Neural Information Processing Systems 15* (ed. S. Becker, S. Thrun & K. Obermayer), pp. 625–632. MIT Press.

LaZerte, S. E. & Albers, S. 2018 weathercan: Download and format weather data from environment and climate change canada. *The Journal of Open Source Software* **3** (22), 571.

Leng, C., Zhang, W. & Pan, J. 2010 Semiparametric mean–covariance regression analysis for longitudinal data. *Journal of the American Statistical Association* **105** (489), 181–193.

Li, Y., Hsing, T. *et al.* 2010 Uniform convergence rates for nonparametric regression and principal component analysis in functional/longitudinal data. *The Annals of Statistics* **38** (6), 3321–3351.

Liu, H., Cai, J., Ong, Y.-S. & Wang, Y. 2018 Understanding and comparing scalable Gaussian process regression for Big Data. arXiv preprint arXiv:1811.01159.

Morris, J. S. 2015 Functional regression. *Annual Review of Statistics and Its Application* **2**, 321–359.

Müller, H.-G., Stadtmüller, U. *et al.* 2005 Generalized functional linear models. *the Annals of Statistics* **33** (2), 774–805.

Novomestky, F. 2013 *orthopolynom: Collection of functions for orthogonal and orthonormal polynomials*. R package version 1.0-5.

Nychka, D., Furrer, R., Paige, J. & Sain, S. 2017 fields: Tools for spatial data. R package version 9.6.

O'Hagan, A. 1978 Curve fitting and optimal design for prediction (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)* **40** (1), 1–42.

Paciorek, C. J. & Schervish, M. J. 2006 Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics* **17** (5), 483–506.

Pinheiro, J. C. & Bates, D. M. 1996 Unconstrained parametrizations for variance-covariance matrices. *Statistics and computing* **6** (3), 289–296.

Poggio, T. & Girosi, F. 1990 Networks for approximation and learning. *Proceedings of the IEEE* **78** (9), 1481–1497.

Pourahmadi, M. 1999 Joint mean-covariance models with applications to longitudinal data: Unconstrained parameterisation. *Biometrika* **86** (3), 677–690.

Pourahmadi, M. 2013 *High-dimensional covariance estimation: with high-dimensional data*, , vol. 882. John Wiley & Sons.

Qiu, Y., Mei, J. & authors of the ARPACK library. 2016 *rARPACK: Solvers for Large Scale Eigenvalue and SVD Problems*. R package version 0.11-0.

R Core Team 2018 *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ramsay, J. & Silverman, B. W. 2005 *Functional Data Analysis*, 2nd edn. Springer.

Ramsay, J. O., Wickham, H., Graves, S. & Hooker, G. 2018 *fda: Functional Data Analysis*. R package version 2.4.8.

Rapisarda, F., Brigo, D. & Mercurio, F. 2007 Parameterizing correlations: a geometric interpretation. *IMA Journal of Management Mathematics* **18** (1), 55–73.

Rasmussen, C. & Williams, C. 2006 *Gaussian Processes for Machine Learning*. University Press Group Limited.

Reiss, P. T., Goldsmith, J., Shang, H. L. & Ogden, R. T. 2016 Methods for scalar-on-function regression. *International Statistical Review* **85** (2), 228–249.

Risser, M. & Calder, C. 2017 Local Likelihood Estimation for Covariance Functions with Spatially-Varying Parameters: The convoSPAT Package for R. *Journal of Statistical Software, Articles* **81** (14), 1–32.

Rougier, J. 2017 A representation theorem for stochastic processes with separable covariance functions, and its implications for emulation. arXiv preprint arXiv:1702.05599.

Sampson, P. D. & Guttorp, P. 1992 Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association* **87** (417), 108–119.

Sanderson, C. 2010 Armadillo: An open source c++ algebra library for fast prototyping and computationally intensive experiments. *Tech. Rep.*. Technical report, NICTA. URL http://arma. sf. net.

Seeger, M., Williams, C. & Lawrence, N. 2003 Fast forward selection to speed up sparse Gaussian process regression. In *Artificial Intelligence and Statistics 9*.

Shah, A., Wilson, A. & Ghahramani, Z. 2014 Student-t Processes as Alternatives to Gaussian Processes. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics* (ed. S. Kaski & J. Corander), *Proceedings of Machine Learning Research*, vol. 33, pp. 877–885. Reykjavik, Iceland: PMLR.

Shi, J. Q. & Choi, T. 2011 *Gaussian process regression analysis for functional data*. CRC Press.

Shi, J. Q., Wang, B., Murray-Smith, R. & Titterington, D. M. 2007 Gaussian process functional regression modeling for batch data. *Biometrics* **63** (3), 714–723.

Silverman, B. W. 1985 Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)* **47** (1), 1–52.

Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., Francois, R. & Soetaert, K. 2018 *inline: Functions to Inline C, C++, Fortran Function Calls from R*. R package version 0.3.15.

Stein, M. L. 1999 *Interpolation of Spatial Data: Some Theory for Kriging*. Springer: NY.

Stein, M. L. 2005 Space–time covariance functions. *Journal of the American Statistical Association* **100** (469), 310–321.

Sung, C.-L., Hung, Y., Rittase, W., Zhu, C. & Wu, C. 2017 A generalized Gaussian process model for computer experiments with binary time series. arXiv preprint arXiv:1705.02511.

Tibshirani, R. & Hastie, T. 1987 Local likelihood estimation. *Journal of the American Statistical Association* **82** (398), 559–567.

Tran, D., Ranganath, R. & Blei, D. M. 2015 The variational Gaussian process. arXiv preprint arXiv:1511.06499.

Wahba, G. 1990 *Spline models for observational data*. SIAM, Philadelphia.

Wang, B. & Shi, J. Q. 2014 Generalized Gaussian process regression model for non-Gaussian functional data. *Journal of the American Statistical Association* **109** (507), 1123–1133.

Wang, B. & Xu, A. 2019 Gaussian process methods for nonparametric functional regression with mixed predictors. *Computational Statistics & Data Analysis* **131**, 80–90, high-dimensional and functional data analysis.

Wang, J.-L., Chiou, J.-M. & Müller, H.-G. 2016 Functional data analysis. *Annual Review of Statistics and Its Application* **3**, 257–295.

Wang, Z., Shi, J. Q. & Lee, Y. 2017 Extended t-process regression models. *Journal of Statistical Planning and Inference* **189**, 38 – 60.

Weidmann, J. 1980 Graduate texts in mathematics. In *Linear operators in Hilbert spaces*. Springer.

van der Wilk, M., Rasmussen, C. E. & Hensman, J. 2017 Convolutional Gaussian Processes. In *Advances in Neural Information Processing Systems 30* (ed. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett), pp. 2849–2858. Curran Associates, Inc.

Williams, C. K. & Seeger, M. 2001 Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pp. 682–688.

Williams, C. K. I., Rasmussen, C. E., Schwaighofer, A. & Tresp, V. 2002 Observations on the Nyström Method for Gaussian Processes. *Tech. Rep.*.

Wood, S. 2018 *orthopolynom: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation*. R package version 1.8-23.

Yang, W., Müller, H.-G. & Stadtmüller, U. 2011 Functional singular component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73** (3), 303–324.

Yao, F., Müller, H.-G. & Wang, J.-L. 2005 Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association* **100** (470), 577–590.

Zhang, B., Sang, H. & Huang, J. Z. 2019 Smoothed full-scale approximation of Gaussian process models for computation of large spatial datasets. Statistica Sinica.

Zhang, W., Leng, C. & Tang, C. Y. 2015 A joint modelling approach for longitudinal studies. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **77** (1), 219–238.