



An Investigation into Functional Linear Regression Modeling

Author:

Rene Franck ESSOMBA

Supervisor:

Assoc. Prof. Sugnet LUBBE

*Submitted to the Department of Statistical Sciences in fulfilment of
the requirements for the degree of
Masters of Science in Mathematical Statistics*

at the

UNIVERSITY OF CAPE TOWN

April 2015

I hereby grant the University of Cape Town permission to reproduce and distribute copies of this dissertation in whole or part in any format the University deems fit.

Plagiarism Declaration

I, Rene Franck ESSOMBA, declare that this thesis titled, 'An Investigation into Functional Linear Regression Modeling' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“A man who has never gone to school may steal from a freight car; but if he has a university education, he may steal the whole railroad.”

Theodore Roosevelt

Abstract

Functional data analysis, commonly known as “FDA”, refers to the analysis of information on curves of functions. Key aspects of FDA include the choice of smoothing techniques, data reduction, model evaluation, functional linear modelling and forecasting methods. FDA is applicable in numerous applications such as Bioscience, Geology, Psychology, Sports Science, Econometrics, Meteorology, etc.

This dissertation main objective is to focus more specifically on Functional Linear Regression Modeling (FLRM), which is an extension of Multivariate Linear Regression Modeling. The problem of constructing a Functional Linear Regression modeling with functional predictors and functional response variable is considered in great details. Discretely observed data for each variable involved in the modeling are expressed as smooth functions using: *Fourier Basis*, *B-Splines Basis* and *Gaussian Basis*. The Functional Linear Regression Model is estimated by the *Least Square* method, *Maximum Likelihood* method and more thoroughly by *Penalized Maximum Likelihood* method. A central issue when modeling Functional Regression models is the choice of a suitable model criterion as well as the number of basis functions and an appropriate smoothing parameter. Four different types of model criteria are reviewed: the *Generalized Cross-Validation*, the *Generalized Information Criterion*, the *modified Akaike Information Criterion* and *Generalized Bayesian Information Criterion*. Each of these aforementioned methods are applied to a dataset and contrasted based on their respective results.

Keywords:

Functional Data Analysis, Basis Expansion, Functional Regression, Smoothing Techniques.

Acknowledgements

The success of this study required the help of various individuals. Without them, I would never have been able to finish my dissertation.

I would like to express my deepest gratitude to my supervisor, Associate Professor Sugnet LUBBE for her guidance, care and patience as well as for providing me with an excellent atmosphere for doing research. I would like to thank Professor Anestis ANTONIADIS, who introduced me with some very fruitful insights in my research. I am would like to express my gratitude to Dr. Shuichi KAWANO who assisted me with some source codes. Some of the computations were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing team: <http://hpc.uct.ac.za/>.

I would like to thank my little family: my mother Ernestine ESSOMBA, my brother Olivier ESSOMBA and Vivienne MUTEMBWA for the constant support and unstoppable messages of encouragements.

Finally, I would to thank Jesus Christ, our Lord and Savior, for giving me the wisdom, strenght, inspiration in exploring things and for giving determination to pursue my studies and to make this study possible.

Contents

Plagiarism Declaration	i
Abstract	iii
Acknowledgements	iv
Contents	v
List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Overview	1
1.2 Objectives	2
1.3 Scope	3
1.4 Layout of Document	3
1.5 Notation	4
1.6 Hardware & Software Specifications	6
2 Tools for Functional Data Analysis	7
2.1 Introduction	7
2.2 Smoothing Techniques using Basis Expansion	10
2.2.1 Fourier Basis	11
2.2.2 B-Splines Basis	13
2.2.3 Gaussian Radial Basis Functions	15
2.2.4 Other Basis Functions	18
2.3 Model Estimation	22
2.3.1 Least Squares Method	23
2.3.2 Maximum Likelihood Method	24
2.4 Model Selection	28
2.4.1 Generalized Cross-Validation (GCV)	28
2.4.2 Generalized Information Criteria (GIC)	31
2.4.3 Modified Akaike Information Criteria (mAIC)	34
2.4.4 Generalized Bayesian Information Criteria (GBIC)	37

2.4.5	The optimal number K of Basis Functions	39
2.5	Functional Descriptive Statistics	40
2.5.1	Mean & Variance functions	40
2.5.2	Covariance and Correlation functions	41
2.6	Parallel Computing using R	42
2.6.1	Parallel Backends	42
2.6.2	Using <code>foreach</code>	42
2.7	High Performance Computing (HPC)	44
2.7.1	Connecting to the UCT ICTS HPC cluster	44
2.7.2	Interacting with the Cluster	47
2.8	Closing Comments	49
3	Mathematics of Functional Data Analysis	50
3.1	Hilbert Spaces	51
3.2	Operators in a Hilbert Space	52
3.3	The Space L^2	54
3.4	Stochastic Processes	54
3.5	Karhunen-Loève Expansion	55
3.6	Closing Comments	57
4	Functional Linear Regression Modeling (FLRM)	58
4.1	Preliminary Cases	59
4.1.1	Scalar response and Functional Independent Variables	59
4.1.2	Multivariate Scalar Response and Functional Independent Variables	60
4.2	Functional Response and Functional Independent Variables	62
4.3	Model Estimation	64
4.3.1	Least Square method	64
4.3.2	Maximum Likelihood method	65
4.3.3	Penalized Maximum Likelihood method	66
4.4	Model Selection Criteria	69
4.4.1	Generalized Cross-Validation	69
4.4.2	Modified AIC	69
4.4.3	Generalized Information Criteria	69
4.4.4	Generalized Bayesian Information Criterion	70
4.5	Closing Comments	71
5	Applications: Functional Linear Regression Modeling	72
5.1	Introduction	72
5.2	Methodology	73
5.3	Gaussian Basis Functions	76
5.3.1	Temperature	76
5.3.2	Wind Speed	77
5.3.3	Log-Precipitation	78
5.4	Fourier Basis Functions	80

5.4.1	Temperature	80
5.4.2	Wind Speed	81
5.4.3	Log-Precipitation	82
5.5	B-Splines Basis Functions	84
5.5.1	Temperature	84
5.5.2	Wind Speed	85
5.5.3	Log-Precipitation	85
5.6	Discussion of the Results	88
6	Conclusion	89
6.1	Concluding Remarks about Objectives	89
6.2	Limitations	90
6.3	Recommendations	91
A	R-Functions	92
A.1	Matrices of Basis Functions and Model Selection	92
A.2	Model Criterion	96
B	Derivations and Proofs	100
B.1	Karhunen-Loeve proofs	100
B.2	Derivation of \mathbf{J} -matrix	102
B.3	Derivation of $\mathbf{R}_\Lambda(\theta)$ matrix	104
B.4	Derivation of $\mathbf{Q}_\Lambda(\theta)$ matrix	105
	Bibliography	106

List of Figures

2.1	Temperature data from Alicante & Oviedo	9
2.2	<i>Fourier Basis</i> with $K=7$	11
2.3	<i>Fourier Basis</i> with $K=5$	12
2.4	<i>Fourier Basis</i> applied on Oviedo Temperature data	12
2.5	<i>B-Splines Basis</i> of order 2 with 4 basis functions	14
2.6	<i>B-Splines Basis</i> of order 4 with 4 basis functions	14
2.7	<i>B-Splines Basis</i> applied on the Motorcycle Data	15
2.8	Contrast between K -means clustering method and <i>B-Splines</i> method	17
2.9	Motorcycle impact data fitted with <i>Gaussian Basis</i> functions	18
2.10	<i>Kernel Smoothing</i> regression at Boundaries	21
2.11	<i>Penalized Least Square</i> method using B-Splines	27
2.12	<i>Penalized Maximum Likelihood</i> method using B-Splines	30
2.13	<i>Penalized Maximum Likelihood</i> method evaluated using GIC	34
2.14	<i>Penalized Maximum Likelihood</i> method evaluated using mAIC	36
2.15	<i>Penalized Maximum Likelihood</i> method evaluated using GBIC	38
2.16	<i>Functional Mean</i> applied on Canadian Weather dataset	40
2.17	Login Window	45
2.18	WinSCP Interface	45
2.19	Configuration Window	46
2.20	PuTTY Login	46
2.21	Inside the cluster	47
5.1	Fitting Temperature with <i>Gaussian basis function</i> on A CORUÑA station	76
5.2	Fitting Wind Speed with <i>Gaussian basis function</i> on A CORUÑA station	77
5.3	Fitting Wind Speed with <i>Gaussian basis function</i> on A CORUÑA station	79
5.4	Fitting Temperature with <i>Fourier basis function</i> on A CORUÑA station	80
5.5	Fitting Wind Speed with <i>Fourier basis function</i> on A CORUÑA station	81
5.6	Fitting Wind Speed with <i>B-Splines basis function</i> on A CORUÑA station	83
5.7	Fitting Temperature with <i>Gaussian basis function</i> on A CORUÑA station	84
5.8	Fitting Wind Speed with <i>B-Splines basis function</i> on A CORUÑA station	85
5.9	Fitting Wind Speed with <i>Gaussian basis function</i> on A CORUÑA station	87

List of Tables

2.1	Minimizing the GCV yielding the optimal $\hat{\lambda}$ using <i>Penalized Maximum Likelihood</i> method	29
2.2	Minimizing the GIC yields to the optimal $\hat{\lambda}$ using <i>Penalized Maximum Likelihood</i> method	33
2.3	Minimizing the mAIC yields to the optimal $\hat{\lambda}$ using <i>Penalized Maximum Likelihood</i> method	35
2.4	Minimizing the GBIC yields to the optimal $\hat{\lambda}$ using <i>Penalized Maximum Likelihood</i> method	38
2.5	Summary of the model selection applied on the Motorcycle Data with $K = 40$	39
5.1	Overall \hat{K} -values and $\log_{10}(\hat{\lambda})$ values computed for all stations	75
5.2	Summary of the model selection on the Log-Precipitation using <i>Gaussian basis functions</i>	78
5.3	Summary of the model selection on the Log-Precipitation using <i>Fourier basis functions</i>	82
5.4	Summary of the model selection on the Log-Precipitation using <i>B-Splines basis functions</i>	86
5.5	<i>Average Mean Square Error</i> for the predicted versus observed values of the the functional <i>Log-Precipitation</i>	88

Chapter 1

Introduction

1.1 Overview

Data in many fields come to us through a process naturally described as functional. Functional data analysis (FDA) has been widely used across many disciplines and Statisticians have shown a great interest in this area of study. The very beginning of its development can be extended at least back to the attempts of Gauss and Legendre were to model and estimate the pathway of a comet (Gauss, 1809, Legendre, 1805). Since then, the usage of the term Functional Data Analysis was first developed by Ramsay and Dalzell (1991), and it evolved with a new approach which represented the results mostly through graphical visualization. Many of the methods used in classical Statistics have their counterparts in the concept of Functional Data Analysis. Some methods are simply the extension of existing techniques in conventional Statistics while others need more than exchanging the summation, used in discrete observation, to an integration (which is a continuum). Some of the exploratory data analysis techniques adapted for Functional Data are introduced, and the variability within and between curves using those tools are explored.

Functional Data Analysis provides useful tools for analyzing datasets that have points observed continuously. Functional Linear Regression Modeling, which is the functional form of Multivariate Linear Regression Modeling, is the central issue that is studied in this dissertation. Various procedures for modeling Functional Linear Regression models have been considered. For a functional covariate and a scalar response, a principal components regression model were proposed (Febrero-Bande and Oviedo de la Fuente, 2012). Neural network models and the use of derivatives were proposed for Functional Data. In many studies, Functional Data have mainly been expressed by *Fourier Basis* or *Splines Basis* and the *Generalized Cross-Validation*

criterion has been used to evaluate the model. Ando, Konishi and Imoto (2008) introduced the *Radial Basis* functions which are a class of single hidden layer feed-forward networks which can be expressed as a linear combination of radially symmetric nonlinear basis functions. The most commonly used function in that context is the *Gaussian Basis* function.

Ramsay and Dalzell (1991) considered a Functional Regression model where both predictor and response variables are given as functions, and thereafter Ramsay and Silverman (2005) considered its modeling strategy. They estimated the model by the *Least Squares* method and then evaluated it by the goodness-of-fit, R^2 . Unfortunately, the estimated model (using *Least Square* method) yielded unstable estimates. Matsui, Kawano and Konishi (2009) developed different estimation and evaluation methods for Functional Regression models where there is more than one functional predictor and a functional response. They used the *Gaussian Basis* as it can provide a useful instrument for transforming discrete observations into functional form. In order to estimate the model parameters, a Functional Regression model is estimated using *Least Square*, *Maximum Likelihood* and *Penalized Maximum Likelihood*. A crucial issue when modeling Functional Linear Regression Models is the choice of the smoothing parameter involved in the method of regularization. For this specific case, modified model criteria are implemented to accommodate the presence of the regularization parameter. The model criteria are: *Generalized Information Criterion*, *modified Akaike Information Criterion* and *Generalized Bayesian Information Criterion*.

1.2 Objectives

The objectives of this dissertation are entirely related to the modelling of Functional Linear Regression. The objectives are summarized by the following points:

1. Define Functional Data Analysis and introduce some important basis functions (mainly *Gaussian*, *Fourier* and *B-Splines*) that will help throughout the dissertation with smoothing pointwise data observed over a continuum.
2. Provide an in-depth understanding of the different model criteria and their computations in the R Programming Language.
3. Cement the mathematical foundations of Functional Data Analysis by providing the relevant definitions and theorems.

4. Thoroughly provide derivations of the all different cases of Functional Linear Regression Models.
5. Simultaneously compare the three main Basis functions and the four model selection criteria through relevant illustrations.

1.3 Scope

Functional Linear Regression contains a broad number of topics and procedural aspects. The concepts behind many of these components of Functional Linear Regression theory are themselves vast and be considered as research topics on their own. As a result, most of these will not be investigated and will be partially reviewed or will be taken as given during the presentation of theoretical concepts. An example of one such aspect which will not be investigated is the implementation of Functional Linear Regression using Functional Principal Components (FPC).

The scope, from a computational point of view, is to develop useful pieces of R-functions that may help to ally with much ease the theory of Functional Linear Regression and the practical aspects of it. The basis functions that will be reviewed are: *Gaussian Basis*, *Fourier Basis* and *B-Splines Basis*. The model estimations that will be reviewed: *Least Squares*, *Maximum Likelihood* and *Penalized Maximum Likelihood*. The model criteria that will be reviewed are: *Generalized Information Criterion*, *modified Akaike Information Criterion* and *Generalized Bayesian Information Criterion*.

1.4 Layout of Document

The layout of this document aims to provide sufficient information regarding Functional Linear Regression Modeling. The next five Chapters will cover the followings:

- **Chapter 2** introduces Functional Data Analysis through smoothing techniques. The different types basis functions are derived as well as their computations. There is a section that explains Model Estimation when converting discretized observations to continuous observations. The next section deals with model selection where the four model criteria that are used in Chapter 4 and Chapter 5. For each scenario, an example is provided for clearer insights. The rest of the Chapter will focus a bit more on the ways to overcome computational challenges when computing functional variables.

- **Chapter 3** focuses on the Mathematics of Functional Data Analysis. This Chapter is very mathematical as it touches on *Hilbert Spaces* and L^2 -space. This Chapter also helps to clarify the reason why a stochastic process evolving over a continuum can be written as a linear combination of basis functions; this is called the *Kahrunen-Loeve* Theorem.
- **Chapter 4** introduces the theory of Functional Linear Regression Model. Also, the model estimations and model criteria are defined in the context of Functional Linear Regression models. The derivation of every case is clearly provided.
- **Chapter 5** serves as an illustrative Chapter to show the computational side of Chapter 4. Towards the end of the Chapter 5, all different basis functions and the four model criteria are compared based on the *Average Mean Square Error* calculated for each of them.
- the document ends with **Chapter 6**, giving the conclusions and recommendations that were accumulated throughout the study of Functional Linear Regression Model.

1.5 Notation

For convenience of the reader, the notation used throughout the document will be summarized here. Each item of the list will be introduced in detail in the text. This section merely provides a means of reference.

- FDA: **F**unctional **D**ata **A**alysis;
- FLRM: **F**unctional **L**inear **R**egression **M**odeling;
- RSS: **R**esidual **S**um of **S**quares
- mAIC: **M**odified **A**kaike **I**nformation **C**riterion;
- GIC: **G**eneralized **I**nformation **C**riterion;
- GBIC: **G**eneralized **B**ayesian **I**nformation **C**riterion;
- GCV: **G**eneralized **C**ross-**V**alidation;
- AMSE: **A**verage **M**ean **S**quared **E**rror;
- N : number of functional data;

- K : number of basis functions;
- H : a Hilbert space;
- t : one dimensional argument representing time;
- \mathcal{T} discrete grid of t -values;
- J : cardinal of the set \mathcal{T} ;
- c_{ik} : k^{th} coefficient of the basis expansion for the i^{th} functional datum;
- \mathbf{C} : matrix of coefficients, with dimensions $N \times K$
- $\phi_k(t)$: k^{th} basis function;
- $\boldsymbol{\phi}(t)$: vector of basis functions, with length J ;
- Φ : matrix of basis functions, with dimensions $N \times J$;
- $\psi_k(t)$: k^{th} basis function for functional response;
- $\boldsymbol{\psi}(t)$: vector of basis functions for functional response, with length J ;
- Ψ : matrix of basis functions for functional response, with dimensions $N \times J$;
- μ_k : mean of k^{th} value;
- σ_k : standard deviation of k^{th} value;
- Σ : variance covariance matrix, with dimension $K \times K$;
- X^T : the transpose of matrix X ;
- \bar{u} : conjugate of vector u ;
- $\langle \cdot, \cdot \rangle$: inner product on a Hilbert Space;
- $\| \cdot \|$: norm of the *inner product*;
- \mathcal{I} : identity matrix;
- $\mathbf{1}$: vector of ones;
- $|\mathbf{X}|_+$: product of non-zero eigenvalues of matrix \mathbf{X} ;
- λ : smoothing parameter;
- Λ : matrix of regularization parameters, with dimension $\sum K^x \times \sum K^x$;
- Δ_s : matrix representing the s^{th} difference operator;

- Ω : penalty matrix;
- \odot : Hadamart product;
- \otimes : Kronecker product;
- R-codes will be written in a typewriter-style font, e.g. `Pen_Max_Likelihood`

1.6 Hardware & Software Specifications

It is known that theory without practice is sterile and practice without theory is blind. Although this dissertation will fairly focus on introducing theory and implementing derivations of the concepts mentioned above, computation of these concepts will be mentioned in the form of examples. Most of the computations are done on the following hardware specifications:

- Intel(R) Core(TM) i7-3610QM CPU 2.30GHz;
- 8.00 GB (RAM);
- Windows 7 Home Premium;
- 64-bit Operating System;
- 1TB 5400RPM S-ATAII Hard Drive.

The software specifications are:

- `R version 3.1.2 (2014-10-31) -- "Pumpkin Helmet"`;
- `x86_64-w64-mingw32/x64 (64-bit); RStudio Version 0.98.1062`;
- all the R Packages are up-to-date.

Regarding the clusters specifications of High Performance Computing unit (University of Cape Town), the readers should consult their website <http://hex.uct.ac.za/>.

Chapter 2

Tools for Functional Data Analysis

This chapter will serve to familiarize the reader with the different tools throughout this dissertation to analyse of Functional Data. The emphasis will be on the relevant techniques and methods that are applied throughout this dissertation. Some examples of fitting basis functions will be performed with their respective plots to bring clarity in the transformation of discretized observed data to functional data. A deeper look into the Mathematics of Functional Data Analysis will be covered in the following chapter *Mathematics of Functional Data Analysis*.

This chapter is organized as follows: **section 2.2** introduces the functional basis expansion techniques; **section 2.3** focuses on model estimation; **section 2.4** introduces the different types of model criteria (*Generalized Cross-Validation*, *Generalized Information Criteria*, *modified Akaike Information Criteria* and *Generalized Bayesian Information Criteria*); **section 2.5** defines descriptive statistics in Functional Data framework; **section 2.6** introduces parallel computing using R and **section 2.7** explains how to use the University of Cape Town high performance computing facilities.

2.1 Introduction

In the Functional Data Analysis context, observed data are regarded as depicting an underlying function at various locations; hence each curve is treated as a single functional entity. Smoothing the Functional Data has a primary role in FDA, as it provides insight in the functional behaviour of the stochastic process.

For any data analysis in the FDA context, the very first step is to derive smooth Functional Data; smoothness in the sense of possessing a certain number of derivatives. Let t be a one-dimensional argument sometimes referred as time. Functions of

t are observed over a discrete grid $\{t_1, \dots, t_J\} \in \mathcal{T}$ at sampling values t_j , which may or may not be equally spaced. In order to create a functional *datum*, a basis needs to be specified. The chosen basis is a linear combination of functions defining the functional object. A functional observation X_i is defined as follows:

$$X_i(t) \approx \sum_{k=1}^K c_{ik} \phi_k(t), \quad \forall t \in \mathcal{T} \quad (2.1)$$

where $\phi_k(t)$ (for $k = 1, \dots, K$) is the k^{th} basis function of the expansion and c_{ik} is the corresponding coefficient. Generally, the observed data are filled with observational errors (or noise) that are superimposed on the underlying signal. In the real world, a typical scenario involves N processes being observed at the same time. Let \mathbf{Y} be a vector of N Functional Data $\mathbf{Y} = [\mathbf{Y}_1^T, \dots, \mathbf{Y}_N^T]^T$, where each Functional Data are written as follows:

$$Y_{ij} = X_i(t_j) + \epsilon_{ij}, \quad 1 \leq j \leq J, \quad 1 \leq i \leq N. \quad (2.2)$$

Y_{ij} is a noisy observation of the stochastic process $X_i(t_j)$ and ϵ_{ij} is a random error with zero mean and variance function σ_i^2 associated with the i^{th} functional datum. As an illustration consider the **Aemet** dataset from the R-package **fda.usc** (Febrero-Bande and Oviedo de la Fuente, 2012). It contains daily measurements of Temperature, Wind Speed and Precipitation from $N = 73$ different weather stations in Spain from 1980 to 2009. In this context, a functional observation consists of 365 pairs (t_j, Y_{ij}) with $t_1 = 0.5, \dots, t_{365} = 364.5$ ($J = 365$). Figure 2.1 shows a plot of the raw data for the stations in **Alicante** and in **Oviedo**. In regression problems, it is very likely that the true function $X(t) = \mathbb{E}(Y|t \in \mathcal{T})$ is a nonlinear function of t . Representing $X(t)$ by a linear model is usually appropriate, and sometimes a necessary approximation. It is convenient because a linear model is easy to interpret and is the first-order Taylor approximation to $X(t)$ (Hastie, Tibshirani and Friedman, 2009). In practice, it is impossible to observe the functional values in continuous time. Smoothing methods using basis expansions are used to trim the erratic pattern of the stochastic process. They provide a good approximation to Functional Data given that the basis functions have the same essential characteristics as the process generating the data, hence minimizing the noise in raw data for calculations and analysis.

There are several types of basis expansions that can be applied to Functional Data.

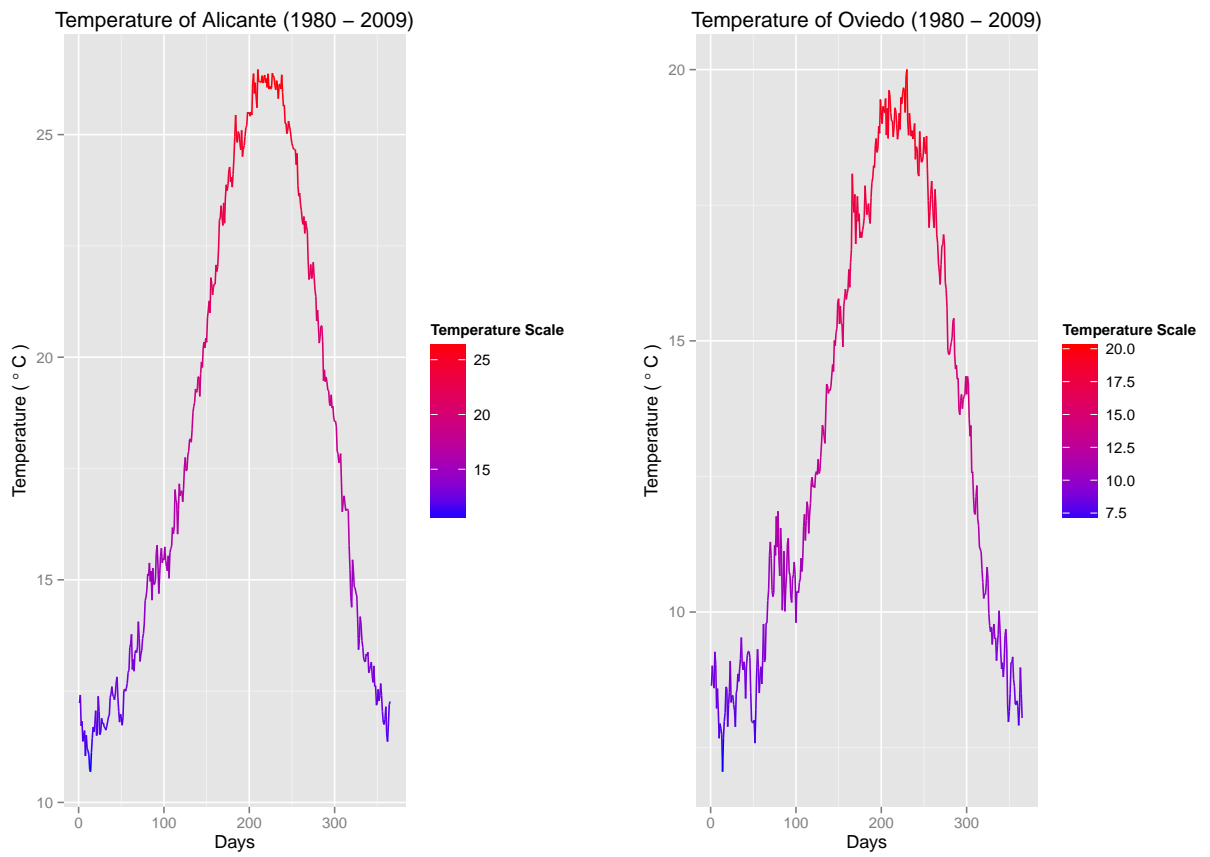


FIGURE 2.1: Temperature data from Alicante & Oviedo stations in Spain (1980 - 2009)

2.2 Smoothing Techniques using Basis Expansion

As stated in the previous section, the first step in FDA is to reconstruct the functional form of the sample curves from their discrete observations. The sample curves are assumed to be observations of a stochastic process $X = \{X(t) : t \in \mathcal{T}\}$ whose sample functions belong to the Hilbert space $L^2(T)$ of square integrable functions with the inner product $\langle X_1, X_2 \rangle_{L^2} = \int_{\mathcal{T}} X_1(t)X_2(t)dt$, $\forall X_1, X_2 \in L^2(\mathcal{T})$. From the previous section, we have seen that any stochastic process can be approximated by taking a weighted sum or *linear combination* of a sufficiently large number K . Equation (2.1) is written as follows:

$$X_i(t) \approx \mathbf{c}_i^T \boldsymbol{\phi}(t), \quad \forall t \in \mathcal{T}, \quad i = 1, \dots, N \quad (2.3)$$

where $\mathbf{c}_i = \begin{bmatrix} c_{i1} \\ c_{i2} \\ \vdots \\ c_{iK} \end{bmatrix}$ and $\boldsymbol{\phi}(t) = \begin{bmatrix} \phi_1(t) \\ \phi_2(t) \\ \vdots \\ \phi_K(t) \end{bmatrix}$. Equation (2.3) can be written in matrix notation as:

$$X_i(\mathbf{t}_i) \approx \boldsymbol{\Phi}(\mathbf{t}_i)\mathbf{c}_i, \quad \forall t \in \mathcal{T}. \quad (2.4)$$

where $\boldsymbol{\Phi}(\mathbf{t}_i) = \begin{pmatrix} \phi_1(t_{i1}) & \dots & \phi_K(t_{i1}) \\ \vdots & \ddots & \vdots \\ \phi_1(t_{iJ}) & \dots & \phi_K(t_{iJ}) \end{pmatrix}$ is a $J \times K$ matrix of basis functions evaluated at each time point t_j .

Basis functions expansion represent the potentially infinite-dimensional universe of functions within the finite-dimensional framework of vectors like \mathbf{c} (Ramsay and Silverman, 2005). A great deal depends on how the vector of basis functions $\boldsymbol{\phi}(t)$ is chosen.

2.2.1 Fourier Basis

The most appropriate basis for periodic functions defined on an interval \mathcal{T} is the *Fourier Basis* where the ϕ_k 's take the following form:

$$\phi_0(t) = 1/\sqrt{|\mathcal{T}|}, \quad \phi_{2r-1}(t) = \frac{\sin(r\omega t)}{\sqrt{|\mathcal{T}|/2}} \quad \text{and} \quad \phi_{2r}(t) = \frac{\cos(r\omega t)}{\sqrt{|\mathcal{T}|/2}} \quad (2.5)$$

for $r = 1, \dots, \frac{K-1}{2}$, where K is the number of basis functions; notice that the K must be an odd number to compute *Fourier Basis*. The frequency ω determines the period and the length of the interval $|\mathcal{T}| = 2\pi/\omega$. The function vector $\boldsymbol{\phi}(t)$ has the form $\boldsymbol{\phi}(t) = [\phi_0(t), \phi_1(t), \phi_2(t), \dots, \phi_{2r}(t)]^T$ evaluated at discrete time points t_j , $j = 1, \dots, J$.

The *Fourier Basis* defined above is said to be orthogonal if the values of t_j are equally spaced on J and the period is equal to the length of \mathcal{T} . Because of the orthogonal property, the cross-product $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ is diagonal and can be made equal to the identity by dividing the basis function by suitable constants $n^{1/2}$ for $j = 0$ and $(n/2)^{1/2}$ for all j . This basis is well known partially due to the Fast Fourier Transform (FFT) Algorithm which makes it possible to compute all the coefficients speedily and efficiently. The figures below are plots of the *Fourier Basis* with different K -values over the interval $[0, 365]$.

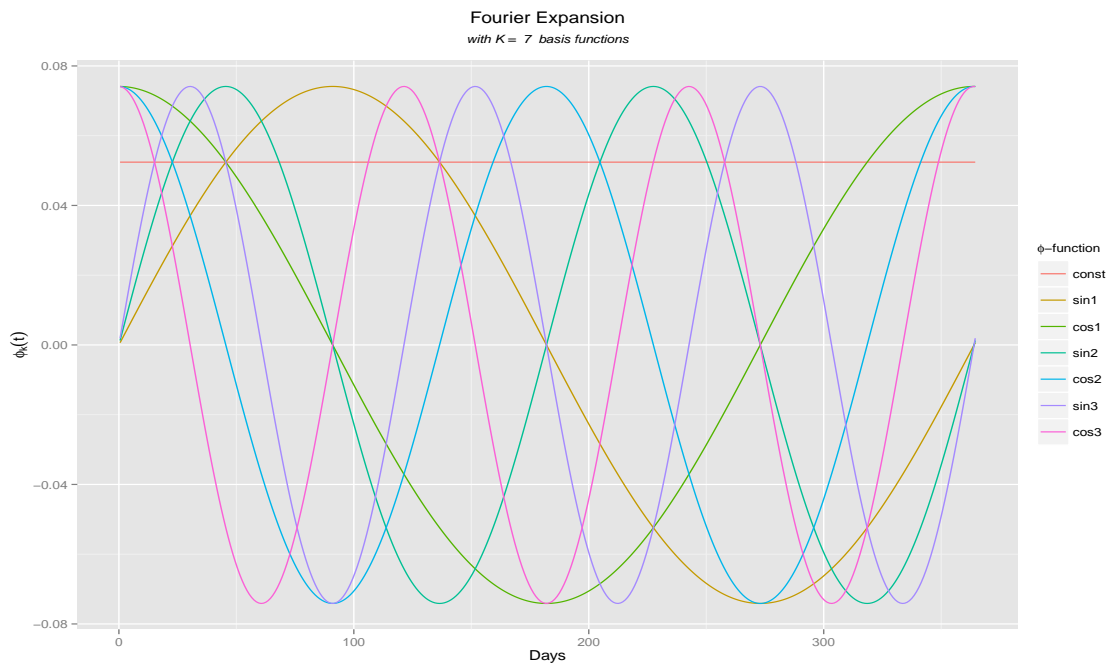


FIGURE 2.2: *Fourier Basis* defined over the interval $[0, 365]$.

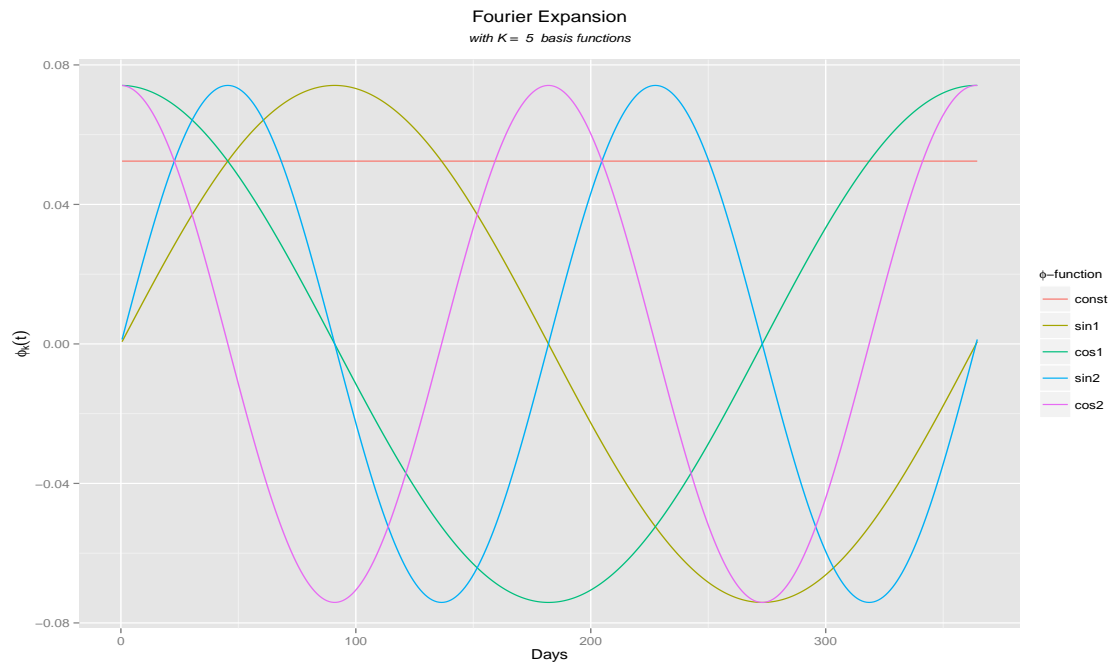
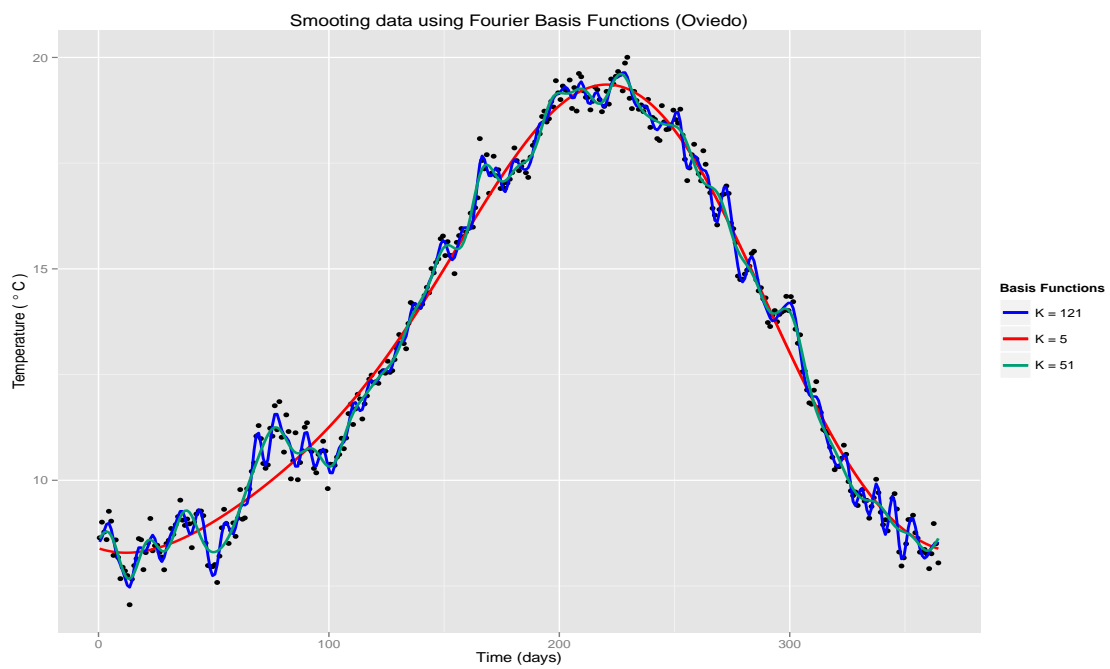
FIGURE 2.3: *Fourier Basis* defined over the interval $[0, 365]$.

Figure 2.4 illustrates the smoothing of the *Temperature* data in the Oviedo using a *Fourier Basis* with $K = 121$, $K = 51$ and $K = 5$.

FIGURE 2.4: *Fourier Basis* applied on Oviedo Temperature data $[0, 365]$.

2.2.2 B-Splines Basis

Originally derived by De Boor (2001), the set of basis splines is a well-known *Functional Basis* for non-periodic data. They are linear combinations of spline functions of specified order over a specified number of breakpoints. A spline is a piecewise polynomial function of order m over each interval, which is smoothly connected at breakpoints. More precisely, the interval \mathcal{T} on which the basis is defined is divided into L subintervals separated by values $\tau_l, l = 0, \dots, L$ called breakpoints or knots. Let $B_{k,m}(t)$ denote the k -th *B-Splines Basis* function of order m defined for any value of t , for the non-decreasing sequence of knots $\{\tau_l\}_{l=0}^L$.

In this case, $\phi_k(t)$ is defined as follows:

$$\phi_k(t) = B_{k,m}(t), \quad \forall t \in \mathcal{T}, \quad k = 1, \dots, m + L - 2 \quad (2.6)$$

Let $\xi_0 < \xi_1$ and $\xi_K < \xi_{K+1}$ be two boundary knots defining the domain over which the spline is evaluated. The augmented knot sequence τ is defined as:

- $\tau_1 \leq \tau_2 \leq \dots \leq \tau_M \leq \xi_0$;
- $\tau_{j+M} = \xi_j, \quad j = 1, \dots, K$;
- $\xi_{K+1} \leq \tau_{K+M+1} \leq \tau_{K+M+2} \leq \dots \leq \tau_{K+2M}$.

Any additional knots beyond the boundary are arbitrary, and the usual scenario is to make them all the same and equal to ξ_0 and ξ_{K+1} . The set of basis functions $B_{k,m}(t)$ of order m for the knot-sequence τ (where $m < M$) is derived using a recursion formula as follows:

$$B_{k,1}(t) = \begin{cases} 1, & t \in [\tau_l, \tau_{l+1}] \\ 0, & \text{otherwise} \end{cases} \quad (2.7)$$

for $k = 1, \dots, K + 2M - 1$. These functions are called Haar basis functions.

$$B_{k,m}(t) = \frac{t - \tau_l}{\tau_{k+m-1} - \tau_k} B_{k,m-1}(t) + \frac{\tau_{k+m} - t}{\tau_{k+m} - \tau_{k+1}} B_{k+1,m-1}(t), \quad \forall t \in \mathcal{T}, \quad m \geq 2. \quad (2.8)$$

for $k = 1, \dots, K + 2M - m$. In this case, the function vector $\phi(t)$ defined in equation (2.3) has $K + 2M - m$ basis functions evaluated at discrete time points t_j , where $j = 1, \dots, J$.

In other words, the number of basis functions is defined by its order and its number of knots. The main advantages of the *B-Splines Basis* are its flexibility as well as its fast computation.

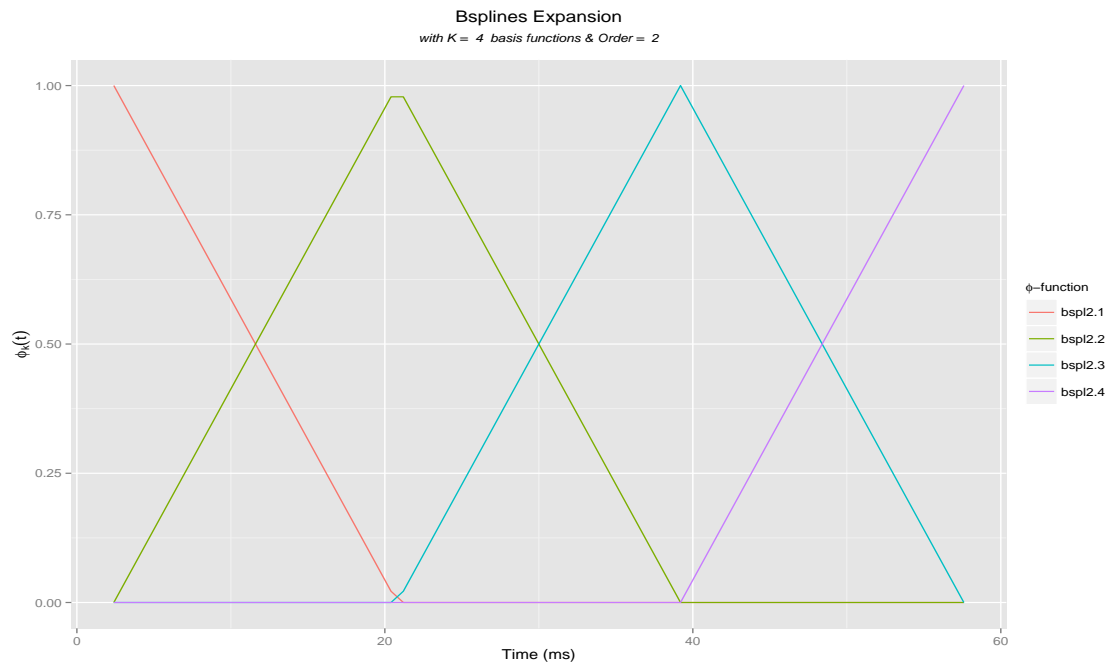


FIGURE 2.5: *B-Splines Basis* of order 2 with 4 basis functions

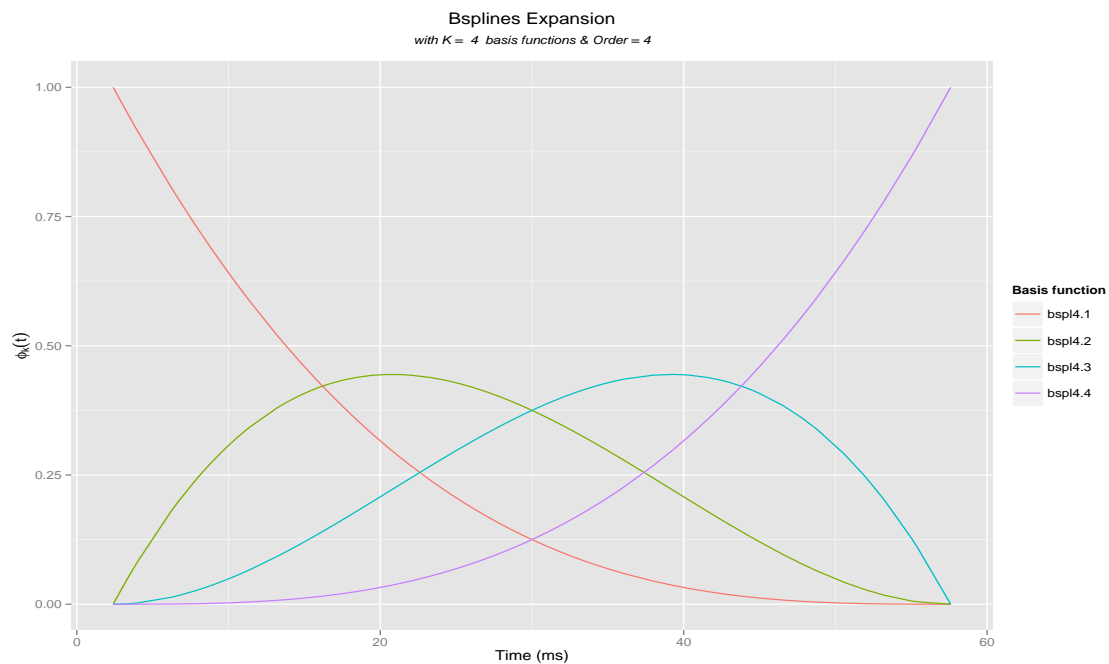


FIGURE 2.6: *B-Splines Basis* of order 4 with 4 basis functions

Smoothed estimates of the observed data are derived when applying this smoothing technique onto a non-periodic dataset, but it depends on the parameter K . For illustration, consider the **Motorcycle Data** which has been widely used by Silverman (1985) and Härdle (1994). For more information on the dataset, refers to the R-package **adlift** created by Knight (2012). Figure 2.7 depicts the **Motorcycle Data** smoothed with the *B-Splines* function for $K = 20$ and $K = 40$.

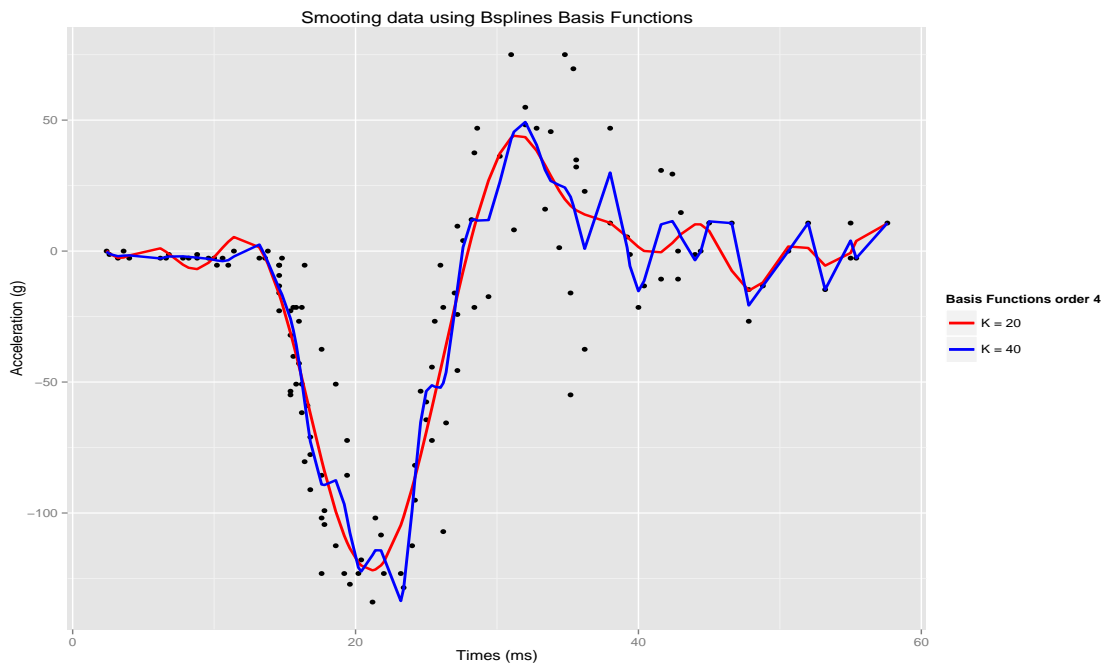


FIGURE 2.7: *B-Splines Basis* Basis applied on the **Motorcycle Data**

2.2.3 Gaussian Radial Basis Functions

Radial basis functions is a class of single hidden layer feedforward networks which can be expressed as a linear combination of radially symmetric nonlinear basis functions (Ando et al., 2008). Each basis function forms a localized receptive field in the input space. The most commonly used function is the *Gaussian Basis* functions which is given by:

$$\phi_k(t; \mu_k, \sigma_k^2) = \exp\left(-\frac{\|t - \mu_k\|^2}{2\sigma_k^2}\right), \quad k = 1, \dots, K \quad (2.9)$$

where μ_k is a parameter determining the center of the basis function, σ_k^2 is a parameter that determines the width and $\|\cdot\|$ is the Euclidian norm. *Gaussian Basis* functions have a number of useful analytical and practical properties (see Bishop (1995)). The basis functions overlap with each other to capture the information about \mathbf{t} . More

importantly, the width parameter play an essential role to capture the structure in the data over the region of input data. The parameters featuring in each basis function are often determined heuristically based on the structure of the observed data.

Moody and Darken (1989) used the **K-means** clustering algorithm to determine both the center and the width parameter of the basis function. This algorithm splits the observational space \mathcal{T} into K clusters $\{C_1, C_2, \dots, C_K\}$ that correspond to the number of basis functions. They are determined by:

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{t_j \in C_k} t_j, \quad \hat{\sigma}_k^2 = \frac{1}{N_k} \sum_{t_j \in C_k} \|t_j - \hat{\mu}_k\|^2, \quad (2.10)$$

where N_k is the number of observations which belongs to the k^{th} cluster. However, this method does not produce unique parameters for a unique set of observations, due to the stochastic nature of the starting value in the clustering algorithm. Because of that feature, the *K-means* clustering underperforms when capturing all the information from the data. This is noticeable when the set of Functional Data are observed at equidistant points.

Ando et al. (2008) proposed to include the hyper-parameter ν (> 0) to control the amount of overlapping as a mean to overcome the lack of overlapping among basis functions. The transformed *Gaussian Basis* are now given by:

$$\phi_k(t; \mu_k, \sigma_k^2) = \exp\left(-\frac{\|t - \mu_k\|^2}{2\nu\sigma_k^2}\right), \quad k = 1, \dots, K \quad (2.11)$$

In order to stabilize the estimation of the Gaussian basis functions parameters, Kawano and Konishi (2007) proposed basis functions where the centers and the width parameters are determined by preassigned knots similar to *B-Splines* basis functions. Consider the observations $\{x_j; j = 1, \dots, n\}$ arranged by magnitude, the knots t_k ($k = 1, \dots, K + 4$) are set up as follows:

$$t_1 < t_2 < t_3 < t_4 = x_1 < t_5 < \dots < t_K < t_{K+1} = x_K < t_{K+2} < t_{K+3} < t_{K+4} \quad (2.12)$$

where the knots are equally spaced. By setting the knots in this way the n observations are divided into $(K - 3)$ intervals:

$$[t_4, t_5], [t_5, t_6], \dots, [t_K, t_{K+1}]. \quad (2.13)$$

The *Gaussian Basis* functions are now defined with a center t_k and a width $h = (t_k - t_{k-2})/3$ for $k = 3, \dots, K + 2$ as follows:

$$\phi_k(x; t_k, h^2) = \exp\left(-\frac{\|x - t_k\|^2}{2h^2}\right) \quad (2.14)$$

$$h = \frac{t_k - t_{k-2}}{3}, \quad k = 3, \dots, K + 2$$

Figure 2.8 shows two different *Gaussian Basis* functions (with 8 functions) plotted on the same set of observations. Note that the areas under the curves for *K-means* clustering are different, whereas the areas for *Gaussian Basis* using *B-Splines* approach are consistent.

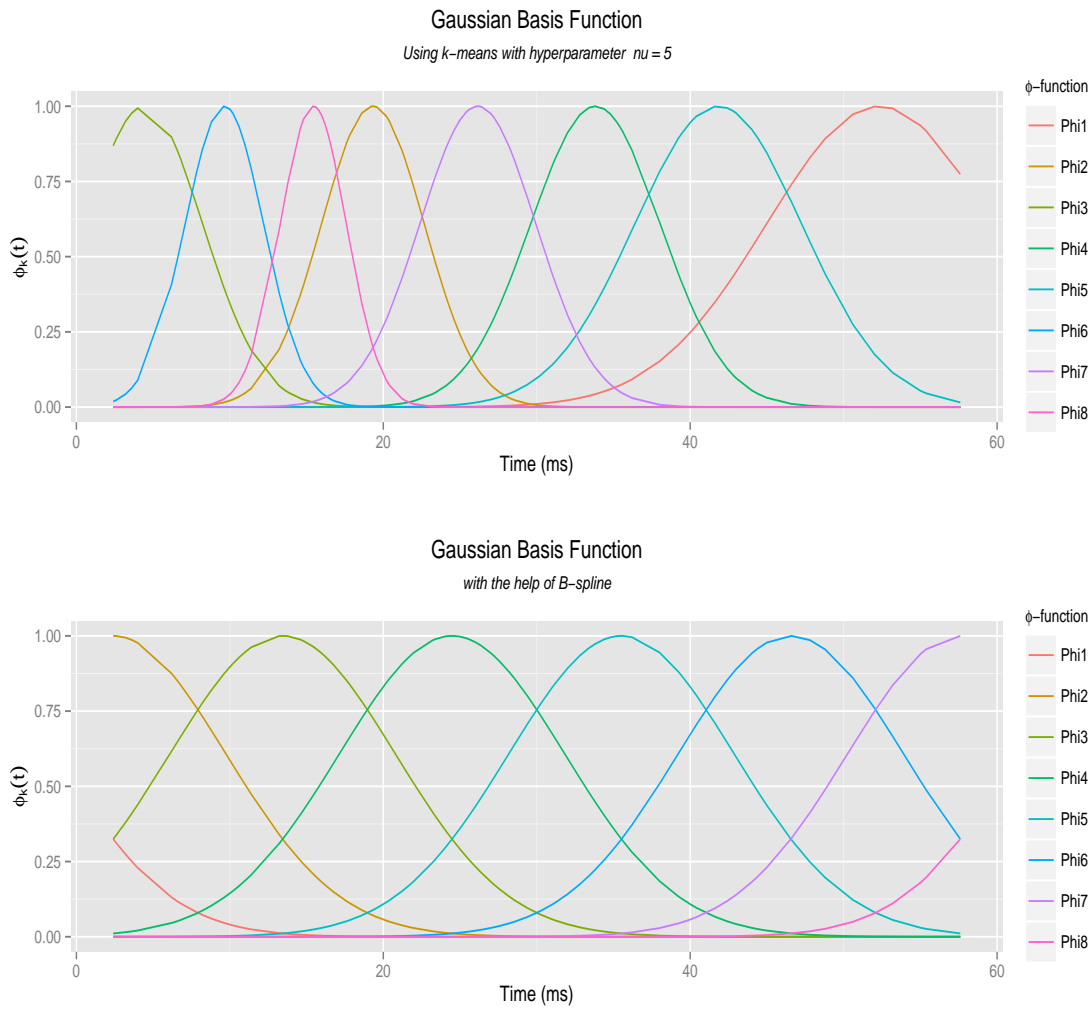


FIGURE 2.8: Contrast between *K-means* clustering method and *B-Splines* method

Figure 2.9 shows smooth curves obtained when applying the abovementioned methods on the motorcycle impact data using the following basis functions: (1) *Gaussian Basis* using *B-Splines* ($K = 20$); (2) *Gaussian Basis* using *K-means* ($K = 20, \nu = 2$).



FIGURE 2.9: Motorcycle impact data fitted with *Gaussian Basis* functions

2.2.4 Other Basis Functions

Recent developments in the study of Functional Data have led to a number of other potentially important basis systems. For instance, the *Haar Wavelets* which combine the frequency-specific approximating power of the *Fourier Basis* with the time- or spatially-localized features of *Splines*. Another example, *Simple Bases* such as step functions and polynomial bases:

Haar Wavelets

The *Haar Wavelets* transform is useful to model a multiresolution stochastic process. It exploits the idea that a basis is constructed by choosing a suitable scaling function ϕ (the *Father Wavelet*) and the function ψ (the *Mother Wavelet*) of the *Haar Wavelets* defined on $[0, 1]$. The functions $\phi(t)$ and $\psi(t)$ are given by:

$$\phi(t) = \begin{cases} 1, & \text{if } 0 \leq t < 1; \\ 0, & \text{if otherwise;} \end{cases}$$

$$\psi(t) = \begin{cases} 1, & \text{if } 0 \leq t < 1/2; \\ -1, & \text{if } 1/2 \leq t < 1; \\ 0, & \text{if otherwise.} \end{cases}$$

The Haar wavelets are then generated in the form of translations and dilations of the above father and mother wavelet functions as

$$\begin{aligned} \phi_{j,k}(t) &= \sqrt{2^j} \phi(2^j t - k), \\ \psi_{j,k}(t) &= \sqrt{2^j} \psi(2^j t - k), \end{aligned}$$

where $j = 0, 1, \dots$ and $k = 0, 1, \dots, 2^j - 1$. The index j refers to dilations and k refers to translations and $\sqrt{2}$ is the normalizing factor. The mother wavelet is constructed to ensure that the basis is orthogonal. The *Wavelet Basis* idea is easily adapted to deal with functions defined on a bounded interval, most simply if periodic boundary conditions are imposed (Ramsay and Silverman, 2005). The coefficients of ψ_{jk} yield information about f near position $2^{-j}k$ on scale 2^{-j} . In contrast to *Fourier Basis*, *Wavelet Basis* expansions cope well with discontinuities and rapid changes in behavior, that allows them to accomodate a wide variety of functional forms. See Walker (2008) for more details on *Wavelet Basis*.

Polynomial bases

The basis functions could be expressed as $\phi_k(t) = (t - \omega)^k$, $\forall t \in \mathcal{T}$, $k = 0, \dots, n$, where ω is a shift parameter that is usually chosen to be in the center of the interval of approximation. Like the *Fourier Basis* expansion, *Polynomials Basis* cannot exhibit local features without using a large number of basis functions. They tend to fit well in the center of the data but exhibit rather unattractive behavior around the tails. Although derivatives of *Polynomials Basis* expansion are simple to compute, they are usually a poor basis for extrapolation or forecasting.

Kernel Smoothing

Smoothing problems, in a statistical framework, are appropriate under the consideration that the data set is merely a realization of a random sample from a certain population. For a smoothing method to make sense, the value of the function estimated at a point $\phi(t_j)$ must be influenced mostly by the observations near that point.

An intuitive estimator $\hat{\phi}_i(t)$ is the locally weighted average. Generally speaking a kernel smoother defines a set of weights $\{W_i^h(t)\}_{i=1}^n$ for each t . Let \mathcal{K} be a real-valued function assigning weights. The function \mathcal{K} , also called the *Kernel* function, is usually a symmetric probability density. Let h be a bandwidth which is a nonnegative number controlling the size of the local neighborhood. Ramsay and Silverman (2005) described the *Kernel Basis* as a function that has most of its mass concentrated close to 0, and either decay rapidly or disappear entirely for $|u| \geq 1$. The most popular *kernel* functions are:

- **Gaussian:** $\mathcal{K}(u) = \frac{1}{\sqrt{2\pi}} \exp[-u^2/2]$
- **Epanechnikov:** $\mathcal{K}(u) = \frac{3}{4} \mathbb{1}_{[-1,1]}(1 - u^2)$
- **Triweight:** $\mathcal{K}(u) = \frac{35}{32} \mathbb{1}_{[-1,1]}(1 - u^2)^3$
- **Uniform:** $\mathcal{K}(u) = \frac{1}{2} \mathbb{1}_{[-1,1]}(u)$
- **Cosine:** $\mathcal{K}(u) = \frac{\pi}{4} \mathbb{1}_{[-1,1]} \cos(\pi \times u/2)$
- **Quartic:** $\mathcal{K}(u) = \frac{15}{16} \mathbb{1}_{[-1,1]}(1 - u^2)^2$

The estimate at a given point is a linear combination of local observations,

$$\hat{\phi}_i(t) = \sum_{j=1}^p \hat{W}_i^h(t_j) Y_j$$

for some suitably defined weight functions $W^h(t_j)$. Nadaraya (1964) and Watson (1964) developed one of the most popular kernel estimator the *Nadaraya-Watson* estimator given by:

$$\hat{W}^h(t_j) = \frac{\sum_{j=1}^p \mathcal{K}^h(t_j - t) Y_j}{\sum_{j=1}^p \mathcal{K}^h(t_j - t)} \quad (2.15)$$

where $\mathcal{K}^h(\cdot) = \mathcal{K}(\cdot/h)/h$. Gasser and Müller (1979), Gasser and Müller (1984) constructed the weights as follows:

$$\hat{W}^h(t_j) = \sum_{j=1}^p \int_{\bar{t}_j}^{\bar{t}_{j-1}} \mathcal{K}^h(u - x) du \times Y_j \quad (2.16)$$

with $\bar{t}_j = (t_{j+1} + t_j)/2$, $1 \leq j \leq n$, $\bar{t}_0 = t_1$ and $\bar{t}_p = t_p$. This estimator was originally proposed for *equispaced designs*, but can also be used for *non-equispaced designs*. Figure 2.10 depicts a visualization of *Kernel smoothing regression*.

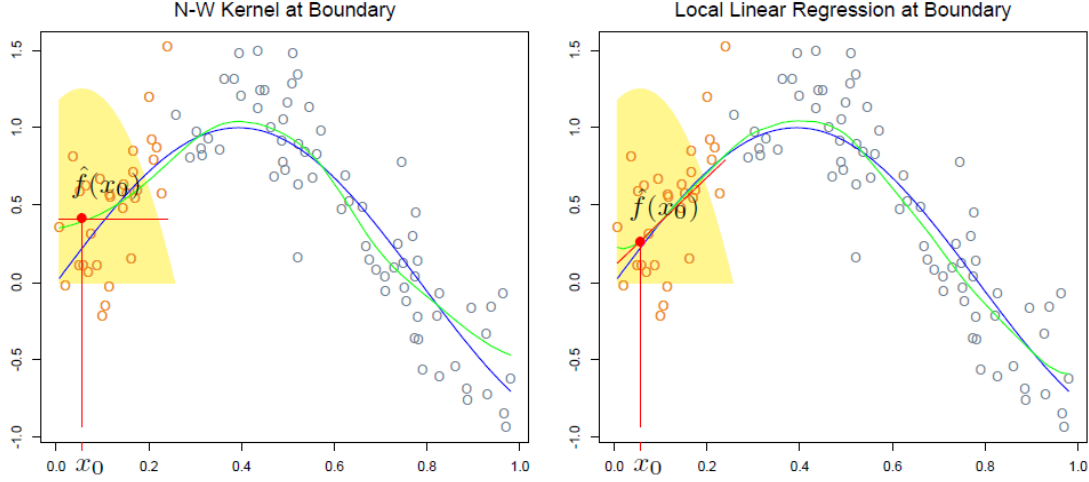


FIGURE 2.10: An illustration of *Kernel Smoothing* regression technique (Image taken from Hastie et al., 2009)

Local Polynomials Fitting

Local polynomials fitting was originally proposed by Cleveland (1979) and further developed by Fan and Gijbels (1995). Consider the bivariate data $(t_1, Y_1), \dots, (t_p, Y_p)$, an i.i.d. sample from a population. The interest is in estimating the regression function $\hat{\phi}(t_0)$ and its derivatives $\hat{\phi}'(t_0), \hat{\phi}''(t_0), \dots, \hat{\phi}^{(m)}(t_0)$. The unknown regression function $X(t)$ is approximated locally by a polynomial of order p at the point t_0 . A Taylor expansion in a neighborhood of t_0 , gives:

$$\phi(t) \approx \phi(t_0) + \phi'(t_0)(t - t_0) + \frac{\phi''(t_0)}{2!}(t - t_0)^2 + \dots + \frac{\phi^{(m)}(t_0)}{m!}(t - t_0)^m. \quad (2.17)$$

This polynomial is fitted locally by a weighted least squares regression problem: minimize

$$\min \left(\sum_{j=1}^p \left\{ Y_j - \sum_{k=1}^m \beta_k (t_j - t_0)^k \right\}^2 \mathcal{K}^h(t_j - t_0) \right) \quad (2.18)$$

where h is a bandwidth controlling the size of the local neighborhood and $\mathcal{K}^h(\cdot) = \mathcal{K}(\cdot/h)/h$ is the function assigning weights to each datum point. Denote by $\hat{\beta}_k$, $k = 0, \dots, m$, the solution to the least squares problem (2.18).

It is clear from the Taylor expansion in (2.17) that $\hat{\phi}_k = k! \hat{\beta}_k$ is an estimator for $\phi^{(k)}(t_0)$, $k = 0, 1, \dots, p$. To estimate the entire function $\phi^{(k)}(\cdot)$ we solve the above weighted least squares equation for all points in the domain of interest. For local polynomial fitting $p - k$ should be taken to be odd as shown in (Fan and Gijbels, 1995, Ruppert and Wand, 1994). Unlike the Nadaraya-Watson and the Gasser-Muller estimators, local polynomial fitting adapts to various types of designs such as random and fixed designs, highly clustered and nearly uniform designs. With local polynomial fitting, there is an absence of boundary effects: the bias at the boundary stays automatically the same as in the interior, without specific boundary kernels.

2.3 Model Estimation

Green and Silverman (1994) point out that a good fit to the data is not the one and only aim of curve fitting; another, often conflicting, aim is to obtain a curve estimate that does not display too much rapid fluctuation. The regularization approach assists modelling and quantifying these rapid fluctuations.

This section describes two different approaches for model estimation when using basis functions, namely the *Least Squares* method (with and without penalty) and the *Maximum Likelihood* method (with and without penalty). Assessing these models consist of estimating several parameters involved in the modelling:

- the multiplier of the penalty term (denoted as λ);
- the number of basis functions (denoted as K);
- some additional parameters based on the model assumptions;
- the coefficients c_{ik} .

For small values of λ the estimated curve becomes more variable since it is being penalized less for its roughness. In other words, as $\lambda \rightarrow 0$, the curve fits the discrete points exactly at almost all sampling points, leading to an interpolation problem. On the other hand, when $\lambda \rightarrow \infty$ the variability in the function $X(t)$ becomes so small that the fitted curve approaches standard linear regression.

2.3.1 Least Squares Method

In a Functional Data framework, the method of Least Square (LS) estimation is a standard approach that consists of minimizing the Residual Sum of Squares (RSS) with $RSS(\mathbf{Y}_i) = \sum_j \left[Y_{ij} - \sum_k c_{ik} \phi_k(t_j) \right]^2, \forall i \in \{1, \dots, N\}$. The RSS in matrix notation is:

$$RSS(\mathbf{Y}_i) = (\mathbf{Y}_i - \Phi \mathbf{c}_i)^T (\mathbf{Y}_i - \Phi \mathbf{c}_i), \quad (2.19)$$

where \mathbf{Y}_i is a vector of observed functional values of length J . The coefficients vector can be estimated by minimizing the RSS namely $\hat{\mathbf{c}}_i = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{Y}_i$. Simple LS approximation is appropriate in situations where it is assumed that the residuals ϵ_i are independently and identically distributed with mean vector $\mu = 0$ and constant variance σ_i .

Unfortunately, fitting basis expansions by least squares implies clumsy discontinuous control over the degree of smoothing. The aim is to look for a model that provides a smooth approach as well as control the degree of smoothness. The basic idea of the regularization approach is similar to the *Least Square* estimation except that we include a penalty term in formula (2.19). The penalized residual sum of squares estimate is defined as:

$$PRSS_{\lambda_i}(Y_i) = (\mathbf{Y}_i - \Phi \mathbf{c}_i)^T (\mathbf{Y}_i - \Phi \mathbf{c}_i) + \lambda_i \times PEN_m(X), \quad (2.20)$$

where $PEN_m(X)$ is the integrated squared m^{th} derivative of $X(t)$ namely $PEN_m(X) = \int_{\mathcal{T}} [D^m X(s)]^2 ds$ and the smoothing parameter λ controls the roughness. Ramsay et al. (2009) extended the definition of *roughness* to situations where a function departs from some baseline "smooth" behavior. For periodic functions of known period that can vary in level, such as mean temperature curves, the baseline behavior can be considered to be shifted sinusoidal variation. The *harmonic acceleration operator* also called *differential operator* is the function defined by $L = \omega^2 D + D^3$. For more details on *harmonic acceleration operator*, interested readers should refer to Ramsay and Silverman (2005).

The roughness penalty $PEN_m(X)$ is re-expressed in matrix terms as:

$$\begin{aligned}
 PEN_m(X) &= \int_{\mathcal{T}} [D^m X(s)]^2 ds \\
 &= \int_{\mathcal{T}} [\mathbf{c}' D^m \boldsymbol{\phi}(s) D^m \boldsymbol{\phi}^T(s) \mathbf{c}] ds \\
 &= \mathbf{c}' \int_{\mathcal{T}} [D^m \boldsymbol{\phi}(s) D^m \boldsymbol{\phi}^T(s) ds] \mathbf{c} \\
 &= \mathbf{c}^T \mathbf{R} \mathbf{c}
 \end{aligned} \tag{2.21}$$

where

$$\mathbf{R} = \int_{\mathcal{T}} D^m \boldsymbol{\phi}(s) D^m \boldsymbol{\phi}^T(s) ds \tag{2.22}$$

Adjusting the results from equations (2.22) and (2.19) gives the following:

$$PRSS_{\lambda_i}(Y_i) = (\mathbf{Y}_i - \boldsymbol{\Phi} \mathbf{c}_i)^T (\mathbf{Y}_i - \boldsymbol{\Phi} \mathbf{c}_i) + \lambda_i \times \mathbf{c}_i^T \mathbf{R} \mathbf{c}_i. \tag{2.23}$$

From the above equation, the estimated coefficient vector is derived as:

$\hat{\mathbf{c}}_i = \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \hat{\lambda}_i \mathbf{R} \right)^{-1} \boldsymbol{\Phi}^T \mathbf{Y}_i$. We define $\mathbf{S}_{\lambda, \phi}^{\text{LS}}$ to be the order N matrix also called *projection operator* onto the basis system $\boldsymbol{\Phi}$ written as

$$\mathbf{S}_{\lambda, \phi}^{\text{LS}} = \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \hat{\lambda}_i \mathbf{R} \right)^{-1} \boldsymbol{\Phi}^T. \tag{2.24}$$

2.3.2 Maximum Likelihood Method

In nonlinear regression, models are usually characterized by a large number of parameters that ought to be estimated. In order to capture the fluctuations in any particular intervals that are much more rapid than those elsewhere, it is important to derive a model that takes into account these parameters. One of the most common approaches is the *Maximum Likelihood* method, which simply maximizes the *Likelihood* function to estimate model parameters. Suppose N independent observations $\{(\mathbf{Y}_i, \mathbf{t}_i); i = 1, \dots, N\}$, where each \mathbf{Y}_i is a vector of J random points observed at $\{t_{i1}, t_{i2}, \dots, t_{iJ}\}$. Equation (2.2) allows to extract information from the data using the Gaussian nonlinear regression model, where $X_i(\cdot)$ is the smooth function and the errors ϵ_i are independently, normally distributed where each element has a mean zero and a variance σ_i^2 with $j = 1, 2, \dots, J$. Hence the regression model with Gaussian noise is expressed as

$$f(Y_{ij}|t_{ij}; \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp \left[-\frac{\{Y_{ij} - \mathbf{c}_i^T \boldsymbol{\phi}(t_{ij})\}^2}{2\sigma_i^2} \right] \quad (2.25)$$

where $\boldsymbol{\theta} = (\mathbf{c}_i^T, \sigma_i^2)^T$. The unknown parameter vector $\boldsymbol{\theta}$ is estimated by maximizing the log-likelihood function:

$$\begin{aligned} l_i(\mathbf{c}_i^T, \sigma_i^2) &= \sum_{j=1}^J \log f(Y_{ij}|t_{ij}; \boldsymbol{\theta}) \\ &= -\frac{J}{2} \log(2\pi\sigma_i^2) - \frac{1}{2\sigma_i^2} (\mathbf{Y}_i - \boldsymbol{\Phi}\mathbf{c}_i)^T (\mathbf{Y}_i - \boldsymbol{\Phi}\mathbf{c}_i). \end{aligned} \quad (2.26)$$

By differentiating $l_i(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta} = (\mathbf{c}_i^T, \sigma_i^2)^T$ and setting the result to 0, the *Maximum Likelihood* estimators for \mathbf{c}_i and σ_i^2 are given by:

$$\hat{\mathbf{c}}_i = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{Y}_i \text{ and } \hat{\sigma}_i^2 = \frac{1}{J} \{ \mathbf{Y}_i - \boldsymbol{\Phi} \hat{\mathbf{c}}_i \}^T \{ \mathbf{Y}_i - \boldsymbol{\Phi} \hat{\mathbf{c}}_i \} \quad (2.27)$$

It can be noticed that the maximum likelihood estimator of \mathbf{c}_i coincides with the least squares estimator (see equation (2.19)).

In the estimation of nonlinear regression models for analyzing data with complex structure, the *Maximum Likelihood* method often yields unstable parameter estimates and complicated regression curves or surfaces (Konishi and Kitagawa, 2008). Originally introduced by Good and Gaskins (1971), the *Penalized Maximum Likelihood* method or *Regularization* method was implemented to account for the *trade-off* between the smoothness of the function and the goodness of fit to the data. The maximized penalized log-likelihood function (or regularized log-likelihood function) is given by:

$$l_{\lambda_i}(\boldsymbol{\theta}) = \sum_{j=1}^J \log f(Y_{ij}|t_{ij}; \boldsymbol{\theta}) - \frac{J}{2} \lambda_i H(\mathbf{c}_i), \quad (2.28)$$

where the regularized parameter λ_i (> 0) allows the function to control the trade-off between the bias and the variance of $X_i(t)$. Depending on the regression functions and data structure under consideration, candidate regularization terms $H(\mathbf{c})$ are: [1] the discrete approximation of the integration of a second-order derivative that takes the curvature of the function into account; [2] the finite differences of the coefficient parameters; and [3] the sum of squares of the coefficients.

The regularization terms are given by:

$$[1] \quad H_1(\mathbf{c}) = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^J \left\{ \frac{\partial^2 X(t_i; \mathbf{c})}{\partial t_j^2} \right\}^2,$$

$$[2] \quad H_2(\mathbf{c}) = \mathbf{c}^T \mathbf{\Delta}_m^T \mathbf{\Delta}_m \mathbf{c} = \mathbf{c}^T \mathbf{\Omega} \mathbf{c} \quad (m^{\text{th}} \text{ order}),$$

$$[3] \quad H_3(\mathbf{c}) = \mathbf{c}^T \mathbb{I}_K \mathbf{c}.$$

The penalty term $H_2(\mathbf{c})$ contains the difference operator represented by the $(K - m) \times K$ matrix $\mathbf{\Delta}_m$ as:

$$\mathbf{\Delta}_m = \begin{bmatrix} {}_m C_0 & -{}_m C_1 & \dots & (-1)^m {}_m C_m & 0 & \dots & 0 \\ 0 & {}_m C_0 & -{}_m C_1 & \dots & (-1)^m {}_m C_m & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & \dots & 0 & {}_m C_0 & -{}_m C_0 & \dots & (-1)^m {}_m C_m \end{bmatrix}$$

with the binomial coefficient ${}_a C_b = \binom{a}{b}$. In practice, it is preferred to use the second-order difference term (i.e. $m = 2$). This is given by:

$$\mathbf{\Delta}_2 = \begin{bmatrix} 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & -2 & 1 \end{bmatrix}.$$

When it comes to the penalized log-likelihood function, the expression is:

$$\begin{aligned} l_{\lambda_i}(\boldsymbol{\theta}) &= \sum_{j=1}^J \log f(Y_{ij} | t_{ij}; \boldsymbol{\theta}) - \frac{J}{2} \lambda_i \mathbf{c}_i^T \mathbf{\Omega} \mathbf{c}_i \\ &= -\frac{J}{2} \log(2\pi\sigma_i^2) - \frac{1}{2\sigma_i^2} \sum_{j=1}^J \{Y_{ij} - \mathbf{c}_i^T \boldsymbol{\phi}(t_{ij})\}^2 - \frac{J}{2} \lambda_i \mathbf{c}_i^T \mathbf{\Omega} \mathbf{c}_i \\ &= -\frac{J}{2} \log(2\pi\sigma_i^2) - \frac{1}{2\sigma_i^2} \{\mathbf{Y}_i - \mathbf{\Phi} \mathbf{c}_i\}^T \{\mathbf{Y}_i - \mathbf{\Phi} \mathbf{c}_i\} - \frac{J}{2} \lambda_i \mathbf{c}_i^T \mathbf{\Omega} \mathbf{c}_i \end{aligned} \quad (2.29)$$

where $\mathbf{\Omega} (= \mathbf{\Delta}_m^T \mathbf{\Delta}_m)$ is a $K \times K$ matrix of rank $(K - m)$ (Konishi et al., 2004). By differentiating $l_{\lambda_i}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta} = (\mathbf{c}_i^T, \sigma_i^2)^T$ and setting the result to 0, the derivation of *Maximum Likelihood* estimators for \mathbf{c}_i and σ_i^2 are given by:

$$\hat{\mathbf{c}}_i = \left(\mathbf{\Phi}^T \mathbf{\Phi} + J \hat{\lambda}_i \hat{\sigma}_i^2 \mathbf{\Omega} \right)^{-1} \mathbf{\Phi}^T \mathbf{Y}_i \quad \text{and} \quad \hat{\sigma}^2 = \frac{1}{J} \{\mathbf{Y}_i - \mathbf{\Phi} \hat{\mathbf{c}}_i\}^T \{\mathbf{Y}_i - \mathbf{\Phi} \hat{\mathbf{c}}_i\} \quad (2.30)$$

$S_{\lambda,\phi}^{ML}$ is defined as the *projection operator* onto the basis system Φ written as:

$$S_{\lambda,\phi}^{ML} = \Phi \left(\Phi' \Phi + J \hat{\lambda} \hat{\sigma}^2 \Omega \right)^{-1} \Phi'. \quad (2.31)$$

It is important to note that the above result is derived for the i^{th} functional datum. Figure 2.11 depicts the effect of changing the smoothing parameter λ on the shape of the smoothed function.

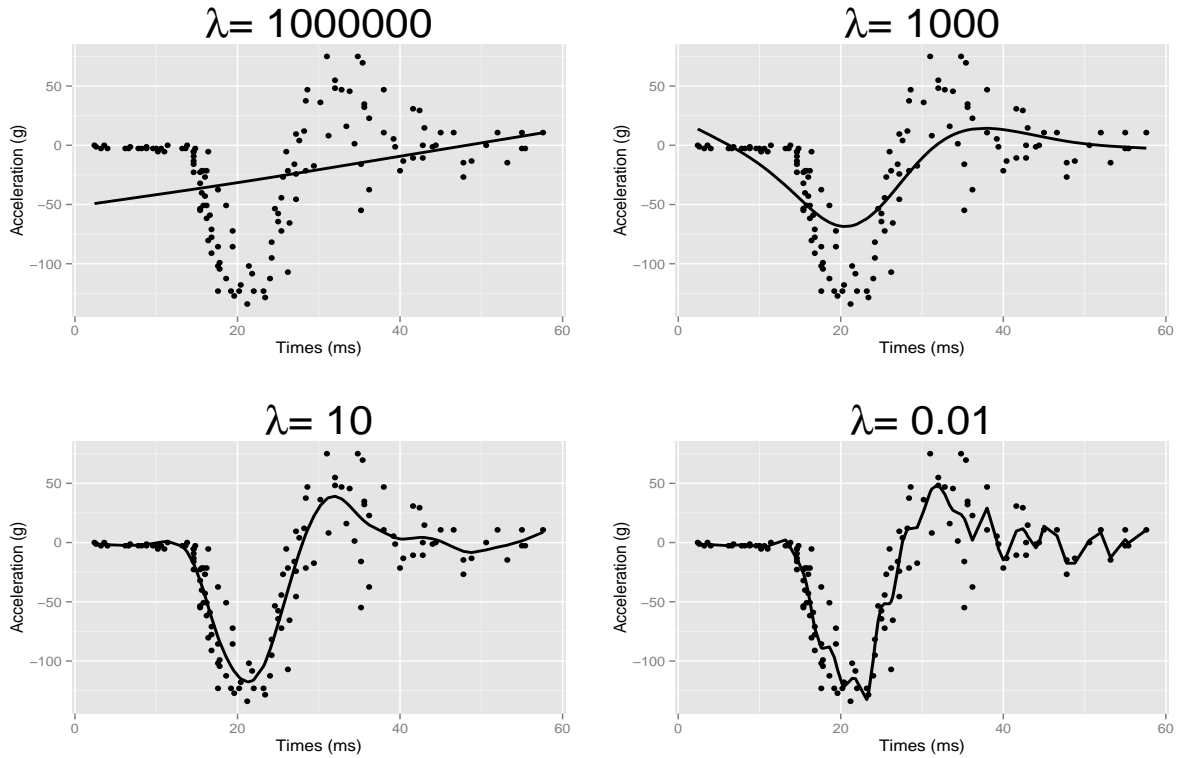


FIGURE 2.11: *Penalized Least Square* method using B-Splines on the Motorcycle Data with different values for the smoothing parameter

2.4 Model Selection

The task of statistical model selection is to choose a family of distributions among a possible set of families, which is the best approximation of reality manifested in the observed data (Rao and Wu, 2001).

2.4.1 Generalized Cross-Validation (GCV)

A measure that is popular in the spline smoothing literature is the *Generalized Cross-Validation* (GCV) developed by Craven and Wahba (1978). This data-driven method helps to estimate the smoothing parameter, λ , which controls the trade-off between the fit of the data and the variability in the function. It is defined to be:

$$\begin{aligned} GCV(\lambda) &= \frac{J^{-1} \text{RSS}}{[J^{-1} \text{trace}(\mathbb{I}_J - \mathbf{S}_{\lambda, \phi})]^2} \\ &= \left(\frac{J}{J - df(\lambda)} \right) \left(\frac{\text{RSS}}{J - df(\lambda)} \right), \end{aligned} \quad (2.32)$$

where $df(\lambda) = \text{trace}(\mathbf{S}_{\lambda, \phi})$. Ramsay and Silverman (2005) refer to the quantity in formula (2.32) as the "*twice-discounted mean squared error measure*". The minimization of GCV with respect to λ involves trying a large set of values of λ . The GCV criterion can be expressed as:

$$GCV(\lambda) = \frac{J \times \text{trace} \{ \mathbf{Y}_i^T [\mathbb{I}_J - \mathbf{S}_{\lambda, \phi}]^{-2} \mathbf{Y}_i \}}{\{ \text{trace} [\mathbb{I}_J - \mathbf{S}_{\lambda, \phi}] \}^2}, \quad (2.33)$$

with \mathbf{Y}_i be the $J \times 1$ vector of observed functional values, Φ the $J \times K$ matrix of basis functions and the *hat* matrix $\mathbf{S}_{\phi, \lambda}$ which is $J \times J$ matrix. With respect to the values of the smoothing parameter λ_i , the selected values of $\hat{\lambda}_i$ that minimize the *Generalized Cross-Validation* value is the optimal value.

Numerical Example: Finding the optimal λ using GCV

Consider the **Motorcycle Data** smoothed using a *Penalized Maximum Likelihood* method as explained in equation (2.29). Table 2.1 shows the values of GCV that are derived from the $\log_{10}(\lambda)$'s ranging from -4.1 to -3.95 . The optimal value for the

TABLE 2.1: $\log_{10}(\lambda)$ against $\text{GCV}(\lambda)$ smoothing the **Motorcycle Data**

$\log_{10}(\lambda)$	$\text{GCV}(\lambda)$
-4.1	567.6346763
-4.09	567.6130784
-4.08	567.6067192
-4.07	567.616065
-4.06	567.6415955
-4.05	567.6838038
-4.04	567.7431973
-4.03	567.820297
-4.02	567.9156388
-4.01	568.0297766
-4	568.1632693
-3.99	568.3167001
-3.98	568.4906646
-3.97	568.6857742
-3.96	568.902656
-3.95	569.1419531

smoothing parameter is at $\hat{\lambda} = 10^{-4.08}$. Figure 2.12 outputs: **(a)** the progression of the GCV-values as the $\log_{10}(\lambda)$'s change with the red line showing the point where the GCV is at its lowest; **(b)** the smooth curve following the pattern of the **Motorcycle Data** using B-splines basis functions with $K = 40$ and $\hat{\lambda} = 8.317638 \times 10^{-5}$ as the smoothing parameter. Note that in this case $\hat{\sigma}^2 = 2116.593$.

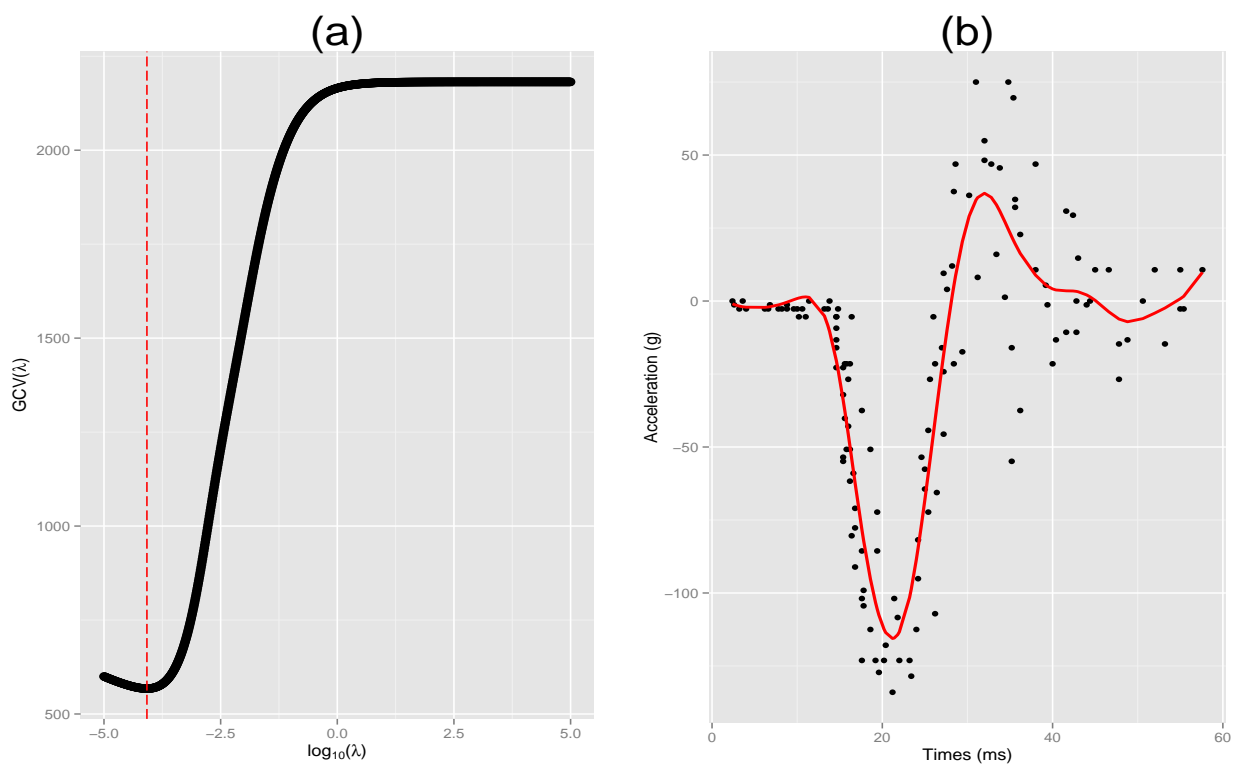


FIGURE 2.12: (a) $\log_{10}(\lambda)$ against $GCV(\lambda)$; (b) Motorcycle Data smoothed using B -Splines Basis functions with $K = 40$ and GCV criterion yielding $\hat{\lambda} = 10^{-4.08}$

2.4.2 Generalized Information Criteria (GIC)

First introduced by Konishi and Kitagawa (1996), the GIC can be applied to evaluate statistical models constructed by various types of estimation procedures, more specifically the models estimated by maximum penalized log-likelihood procedures.

Let $G(Y)$ be the true distribution function with density $g(Y)$ that generated data, and let $\hat{G}(Y)$ be the empirical distribution function based on J observations, $\mathbf{Y}_i = (Y_{i1}, Y_{i2}, \dots, Y_{iJ})^T$, drawn from $G(Y)$. Let $\hat{\boldsymbol{\theta}}_{GIC}$ be the estimator that maximizes the penalized log-likelihood function (2.29). It is clear that the estimator $\hat{\boldsymbol{\theta}}_{GIC}$ is given as the solution to the following equation:

$$\sum_{j=1}^J \boldsymbol{\psi}_{GIC}(Y_{.j}, \boldsymbol{\theta}) = 0, \quad (2.34)$$

where

$$\boldsymbol{\psi}_{GIC}(Y_{.j}, \boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} \left\{ \log f(Y_{.j}|t_{.j}; \boldsymbol{\theta}) - \frac{\lambda_i}{2} \mathbf{c}^T \boldsymbol{\Omega} \mathbf{c} \right\} \quad (2.35)$$

An information criterion for the model $f(Y|\mathbf{t}; \hat{\boldsymbol{\theta}}_{GIC})$ with $\hat{\boldsymbol{\theta}}_{GIC}$ obtained by maximizing (2.29) is given by:

$$\text{GIC}_{PML} = -2 \sum_{j=1}^J \log f(Y_{.j}|t_{.j}; \hat{\boldsymbol{\theta}}_{GIC}) + 2 \text{tr} \left\{ \mathbf{R}(\boldsymbol{\psi}_{GIC}, \hat{G})^{-1} \mathbf{Q}(\boldsymbol{\psi}_{GIC}, \hat{G}) \right\}, \quad (2.36)$$

where $\mathbf{R}(\boldsymbol{\psi}_{GIC}, \hat{G})$ and $\mathbf{Q}(\boldsymbol{\psi}_{GIC}, \hat{G})$ are $(K+1) \times (K+1)$ matrices given by:

$$\mathbf{R}(\boldsymbol{\psi}_{GIC}, \hat{G}) = -\frac{1}{J} \sum_{j=1}^J \frac{\partial \boldsymbol{\psi}_{MP}(Y_{.j}, \boldsymbol{\theta})^T}{\partial \boldsymbol{\theta}} \bigg|_{\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_{GIC}}, \quad (2.37)$$

$$\mathbf{Q}(\boldsymbol{\psi}_{GIC}, \hat{G}) = -\frac{1}{J} \sum_{j=1}^J \boldsymbol{\psi}(Y_{.j}, \boldsymbol{\theta}) \frac{\partial \log f(Y_{.j}|t_{.j}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \bigg|_{\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_{GIC}}. \quad (2.38)$$

By setting $l_j(\boldsymbol{\theta}) = \log f(Y_{ij}|t_{ij}; \boldsymbol{\theta})$ (as in equation (2.26)), its first and second partial derivatives with respect to $\boldsymbol{\theta} = (\mathbf{c}_i^T, \sigma_i^2)^T$ are given by:

$$\begin{aligned}\frac{\partial l_j(\boldsymbol{\theta})}{\partial \sigma_i^2} &= -\frac{1}{2\sigma_i^2} + \frac{1}{2\sigma_i^4} \{Y_{ij} - \mathbf{c}_i^T \boldsymbol{\phi}(t_{ij})\}^2, \\ \frac{\partial l_j(\boldsymbol{\theta})}{\partial \mathbf{c}_i} &= \frac{1}{\sigma_i^2} \{Y_{ij} - \mathbf{c}_i^T \boldsymbol{\phi}(t_{ij})\} \boldsymbol{\phi}(t_{ij}),\end{aligned}\tag{2.39}$$

and

$$\begin{aligned}\frac{\partial^2 l_j(\boldsymbol{\theta})}{\partial \sigma_i^2 \partial \sigma_i^2} &= -\frac{1}{2\sigma_i^4} - \frac{1}{\sigma_i^6} \{Y_{ij} - \mathbf{c}_i^T \boldsymbol{\phi}(t_{ij})\}^2, \\ \frac{\partial^2 l_j(\boldsymbol{\theta})}{\partial \mathbf{c}_i \partial \mathbf{c}_i^T} &= -\frac{1}{\sigma_i^2} \boldsymbol{\phi}(t_{ij}) \boldsymbol{\phi}(t_{ij})^T, \\ \frac{\partial^2 l_j(\boldsymbol{\theta})}{\partial \sigma_i^2 \partial \mathbf{c}_i} &= -\frac{1}{\sigma_i^4} \{Y_{ij} - \mathbf{c}_i^T \boldsymbol{\phi}(t_{ij})\} \boldsymbol{\phi}(t_{ij}).\end{aligned}\tag{2.40}$$

The matrices $\mathbf{R}(\cdot)$ & $\mathbf{Q}(\cdot)$ can be derived as follows:

$$\frac{\partial \psi_{MP}(Y_{.j}, \boldsymbol{\theta})^T}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial^2 l_j(\boldsymbol{\theta})}{\partial \mathbf{c}_i \partial \mathbf{c}_i^T} - \lambda_i \boldsymbol{\Omega} & \frac{\partial^2 l_j(\boldsymbol{\theta})}{\partial \mathbf{c}_i \partial \sigma_i^2} \\ \frac{\partial^2 l_j(\boldsymbol{\theta})}{\partial \sigma_i^2 \partial \mathbf{c}_i} & \frac{\partial^2 l_j(\boldsymbol{\theta})}{\partial \sigma_i^2 \partial \sigma_i^2} \end{bmatrix},$$

$$\begin{aligned}\psi_{MP}(Y_{.j}, \boldsymbol{\theta}) \frac{\partial \log f(Y_{.j}|t_{.j}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \begin{bmatrix} \frac{\partial l_j(\boldsymbol{\theta})}{\partial \mathbf{c}_i} \frac{\partial l_j(\boldsymbol{\theta})}{\partial \mathbf{c}_i^T} - \lambda_i \boldsymbol{\Omega} \mathbf{c}_i \frac{\partial l_j(\boldsymbol{\theta})}{\partial \mathbf{c}_i^T} & \frac{\partial l_j(\boldsymbol{\theta})}{\partial \mathbf{c}_i} \frac{\partial l_j(\boldsymbol{\theta})}{\partial \sigma_i^2} - \lambda_i \boldsymbol{\Omega} \mathbf{c}_i \frac{\partial l_j(\boldsymbol{\theta})}{\partial \sigma_i^2} \\ \frac{\partial l_j(\boldsymbol{\theta})}{\partial \sigma_i^2} \frac{\partial l_j(\boldsymbol{\theta})}{\partial \mathbf{c}_i^T} & \left\{ \frac{\partial l_j(\boldsymbol{\theta})}{\partial \sigma_i^2} \right\}^2 \end{bmatrix},\end{aligned}$$

therefore:

$$\mathbf{R}(\psi_{GIC}, \hat{G}) = \frac{1}{J\sigma_i^2} \begin{bmatrix} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + J\lambda_i \hat{\sigma}_i^2 \boldsymbol{\Omega} & \frac{1}{\hat{\sigma}_i^2} \boldsymbol{\Phi}^T \boldsymbol{\Lambda}_i \mathbf{1}_J \\ \frac{1}{\hat{\sigma}_i^2} \mathbf{1}_J^T \boldsymbol{\Lambda}_i \boldsymbol{\Phi} & \frac{J}{2\sigma_i^2} \end{bmatrix},$$

$$\mathbf{Q}(\psi_{GIC}, \hat{G}) = \frac{1}{J\sigma_i^2} \begin{bmatrix} \frac{1}{2\sigma_i^2} \Phi^T \Lambda_i^2 \Phi - \lambda_i \Omega \mathbf{c}_i \mathbf{1}_J^T \Lambda_i \Phi & \frac{1}{2\hat{\sigma}_i^4} \Phi^T \Lambda_i^3 \mathbf{1}_J - \frac{1}{2\hat{\sigma}_i^2} \Phi^T \Lambda_i \mathbf{1}_J \\ \frac{1}{2\hat{\sigma}_i^4} \mathbf{1}_J^T \Lambda_i^3 \Phi - \frac{1}{2\hat{\sigma}_i^4} \mathbf{1}_J^T \Lambda_i \Phi & \frac{1}{4\hat{\sigma}_i^6} \mathbf{1}_J^T \Lambda_i^4 \mathbf{1}_J - \frac{J}{4\sigma_i^2} \end{bmatrix},$$

where $\mathbf{1}_J = (1, 1, \dots, 1)^T$ is a J -dimensional vector of 1's, and Λ is a $J \times J$ diagonal matrix defined by

$$\Lambda_i = \text{diag} [Y_{i1} - \hat{\mathbf{c}}_i^T \phi(t_{i1}), Y_{i2} - \hat{\mathbf{c}}_i^T \phi(t_{i2}), \dots, Y_{iJ} - \hat{\mathbf{c}}_i^T \phi(t_{iJ})]$$

Numerical Example: Finding the optimal λ using GIC

Consider the **Motorcycle Data** smoothed using the *Penalized Maximum Likelihood* method as explained in equation (2.29). Table 2.2 shows the values of GIC that are derived from the $\log_{10}(\lambda)$'s ranging from -4.3 to -4.1 . The optimal value for the

TABLE 2.2: $\log_{10}(\lambda)$ against $\text{GIC}(\lambda)$ smoothing the **Motorcycle Data**

$\log_{10}(\lambda)$	$\text{GIC}(\lambda)$
-4.3	1216.693
-4.29	1216.679
-4.28	1216.666
-4.27	1216.655
-4.26	1216.646
-4.25	1216.639
-4.24	1216.633
-4.23	1216.630
-4.22	1216.629
-4.21	1216.630
-4.2	1216.633
-4.19	1216.638
-4.18	1216.646
-4.17	1216.656
-4.16	1216.669
-4.15	1216.684
-4.14	1216.702
-4.13	1216.723

smoothing parameter is at $\hat{\lambda} = 10^{-4.22}$. Figure 2.13 outputs: **(a)** the progression of the GIC-values as the $\log_{10}(\lambda)$'s change with the red line showing the point where the GIC is at its lowest; **(b)** the smooth curve following the pattern of the **Motorcycle Data** using *B-Splines Basis* functions with $K = 40$ and $\hat{\lambda} = 6.025596 \times 10^{-5}$ as the smoothing parameter. It is important to note that $\hat{\sigma}^2 = 462.5911$.

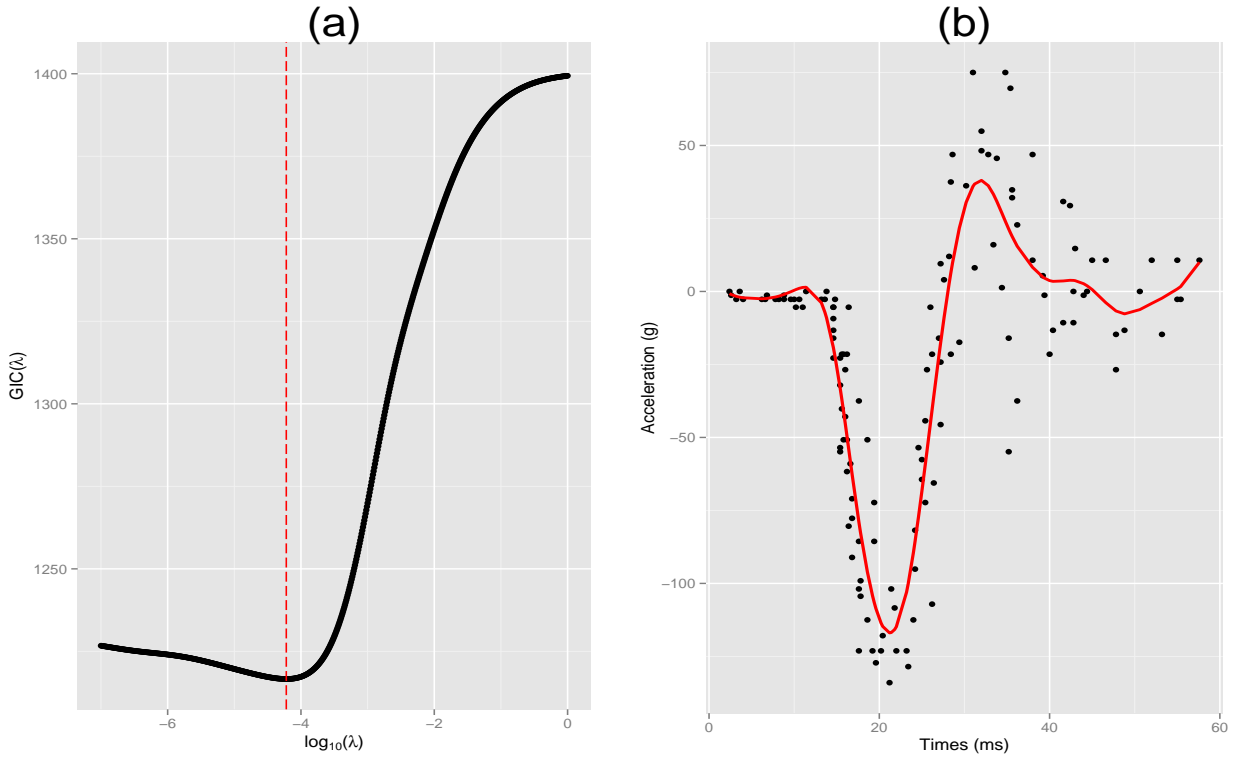


FIGURE 2.13: (a) $\log_{10}(\lambda)$ against $GIC(\lambda)$; (b) **Motorcycle Data** smoothed using *B-Splines Basis* functions with $K = 40$ and GIC criterion yielding $\hat{\lambda} = 10^{-4.22}$ & $\hat{\sigma}^2 = 462.5911$

2.4.3 Modified Akaike Information Criteria (mAIC)

The Akaike's information criteria (1973) was derived as an estimator of the Kullback and Leibler (1951) information from the predictive point of view. It is given by:

$$-2l(\mathbf{Y}_i|\hat{\boldsymbol{\theta}}_{ML}) + 2(\text{number of parameters}) \quad (2.41)$$

where $l(\hat{\boldsymbol{\theta}}_{ML})$ is the log-likelihood of a model estimated by the *Maximum Likelihood* and the “number of parameters” is a measure of complexity of the model. However, in nonlinear modelling, the “number of parameters” is not an appropriate measure of model complexity since it may depend on both the regularization term and the observed data. Fujikoshi and Satoh (1997) considered using the trace *smoother operator* (see equations (2.24) & (2.31)) as an approximation to the effective “number of parameters”. By replacing the last term in (2.41) by $\text{tr}(S_{\lambda_i})$, the *mAIC* is given by:

$$\text{mAIC} = J \log(2\pi\hat{\sigma}_i^2) + J + 2 \text{tr}(S_{\lambda_i}) \quad (2.42)$$

Numerical Example: Finding the optimal λ using mAIC

The `Motorcycle Data` is smoothed using the *penalized maximum likelihood*. Table 2.3 is showing the values of mAIC that are derived from the $\log_{10}(\lambda)$'s ranging from -4.2 to -4.05 . The optimal value for the smoothing parameter is at $\hat{\lambda} = 10^{-4.12}$.

TABLE 2.3: $\log_{10}(\lambda)$ against mAIC(λ) smoothing the `Motorcycle Data`

$\log_{10}(\lambda)$	mAIC(λ)
-4.2	1216.633
-4.19	1216.638
-4.18	1216.646
-4.17	1216.656
-4.16	1216.669
-4.15	1216.684
-4.14	1216.702
-4.13	1216.723
-4.12	1216.747
-4.11	1216.774
-4.1	1216.803
-4.09	1216.836
-4.08	1216.873
-4.07	1216.913
-4.06	1216.956
-4.05	1217.003

Figure 2.14 outputs: **(a)** the progression of the mAIC-values as the $\log_{10}(\lambda)$'s change with the red line showing the point where the mAIC is at its lowest; **(b)** the smooth curve following the pattern of the `Motorcycle Data` using *B-Splines Basis* functions with $K = 40$ and $\hat{\lambda} = 7.585776 \times 10^{-5}$ as the smoothing parameter. Note that $\hat{\sigma}^2 = 466.3463$

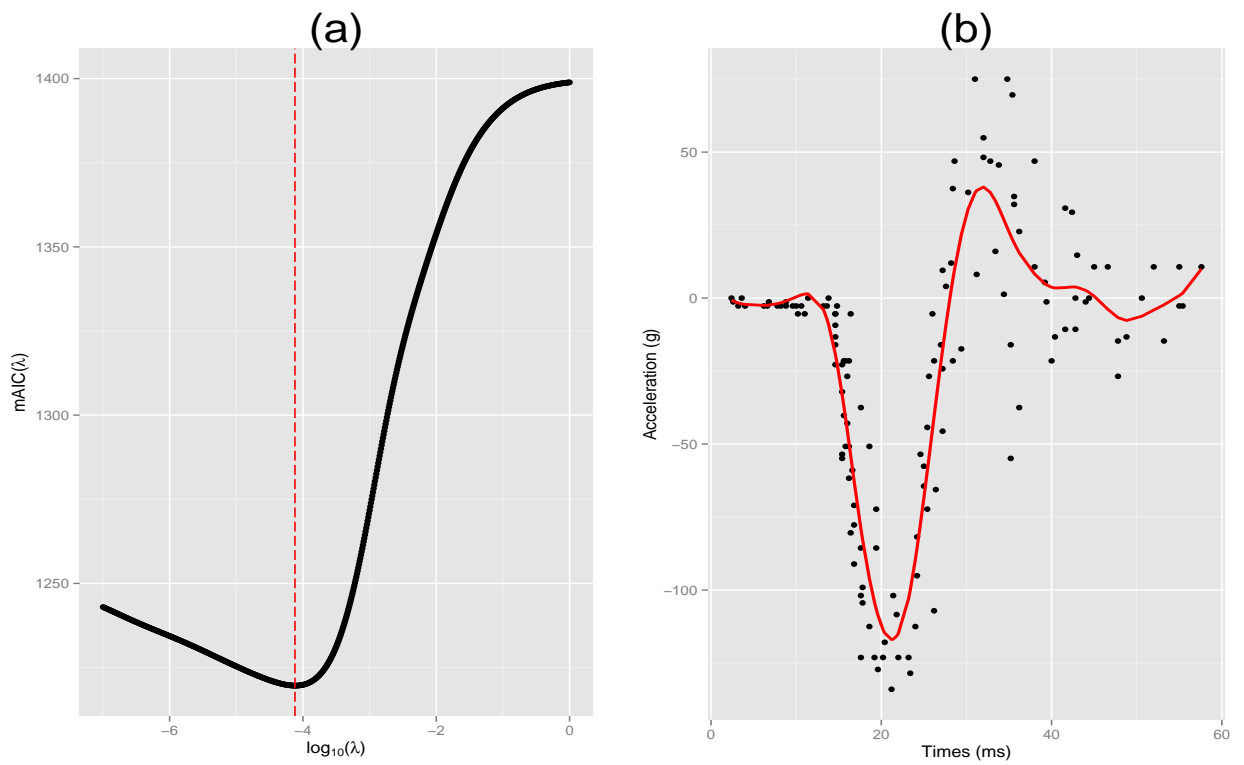


FIGURE 2.14: (a) $\log_{10}(\lambda)$ against $\text{mAIC}(\lambda)$; (b) Motorcycle Data smoothed using B-splines basis functions with $K = 40$ and mAIC criterion yielding $\hat{\lambda} = 10^{-4.12}$ & $\hat{\sigma}^2 = 466.3463$

2.4.4 Generalized Bayesian Information Criteria (GBIC)

Derived from the well known *Bayesian Information Criteria* (BIC), it is a model selection criterion used to evaluate models fitted by the *Penalized Maximum Likelihood* method or the method of *Regularization*. Konishi et al. (2004) derived this criterion in order to estimate the smoothing parameters as well as other parameters such as the number of basis functions.

Suppose that the suitable model is constructed by maximizing equation (2.29) yielding the maximum likelihood estimators for \mathbf{c}_i and σ_i^2 (see equation 2.30). Considering $\beta_i = \lambda_i \sigma_i^2$ and substituting back in equation (2.30), the *Generalized Bayesian Information Criteria* is given by:

$$\begin{aligned} \text{GBIC} = & (J + K - 1) \log \hat{\sigma}_i^2 + J \beta_i \hat{\mathbf{c}}_i^T \boldsymbol{\Omega} \hat{\mathbf{c}}_i / \hat{\sigma}_i^2 + J + (J - 3) \log(2\pi) + 3 \log J \\ & + \log |\mathbf{Q}_{\beta_i}^{(G)}(\hat{\mathbf{c}}_i^T, \hat{\sigma}_i^2)| - \log |\boldsymbol{\Omega}|_+ - (K - 1) \log \beta_i \end{aligned} \quad (2.43)$$

where $|\boldsymbol{\Omega}|_+$ denotes the product of nonzero eigenvalues of $\boldsymbol{\Omega}$ and

$$\mathbf{Q}_{\beta_i}^{(G)}(\hat{\mathbf{c}}_i^T, \hat{\sigma}_i^2) = \frac{1}{J \hat{\sigma}_i^2} \begin{bmatrix} \boldsymbol{\Phi}^T \boldsymbol{\Phi} + J \beta_i \boldsymbol{\Omega} & \boldsymbol{\Phi}^T \mathbf{e} / \hat{\sigma}_i^2 \\ \mathbf{e}^T \boldsymbol{\Phi} / \hat{\sigma}_i^2 & \frac{J}{2 \hat{\sigma}_i^2} \end{bmatrix}.$$

Note that \mathbf{e} is a J -dimensional vector given by

$$\mathbf{e} = [Y_{i1} - \hat{\mathbf{c}}_i^T \boldsymbol{\phi}(t_{i1}), Y_{i2} - \hat{\mathbf{c}}_i^T \boldsymbol{\phi}(t_{i2}), \dots, Y_{iJ} - \hat{\mathbf{c}}_i^T \boldsymbol{\phi}(t_{iJ})]^T.$$

For a more extensive derivation of the above result, consult the journal article written by Konishi and Kitagawa (1996).

Numerical Example: Finding the optimal λ using GBIC

The **Motorcycle Data** is smoothed using the *Penalized Maximum Likelihood*. Table 2.4 is showing the values of GBIC that are derived from the $\log_{10}(\lambda)$'s ranging from -4.3 to -4.15 . The optimal value for the smoothing parameter is at $\hat{\lambda} = 10^{-4.23}$. Figure 2.15 outputs: **(a)** the progression of the mAIC-values as the $\log_{10}(\lambda)$'s change with the red line showing the point where the mAIC is at its lowest; **(b)** the smooth curve following the pattern of the **Motorcycle Data** using *B-Splines Basis* functions with $K = 40$ and $\hat{\lambda} = 5.888437 \times 10^{-5}$ as the smoothing parameter. Note that $\hat{\sigma}^2 = 462.2833$

TABLE 2.4: $\log_{10}(\lambda)$ against $\text{GBIC}(\lambda)$ smoothing the *Motorcycle* Data

$\log_{10}(\lambda)$	$\text{GBIC}(\lambda)$
-4.3	1258.324
-4.29	1258.288
-4.28	1258.257
-4.27	1258.232
-4.26	1258.213
-4.25	1258.200
-4.24	1258.193
-4.23	1258.192
-4.22	1258.196
-4.21	1258.208
-4.2	1258.225
-4.19	1258.249
-4.18	1258.279
-4.17	1258.315
-4.16	1258.358
-4.15	1258.408

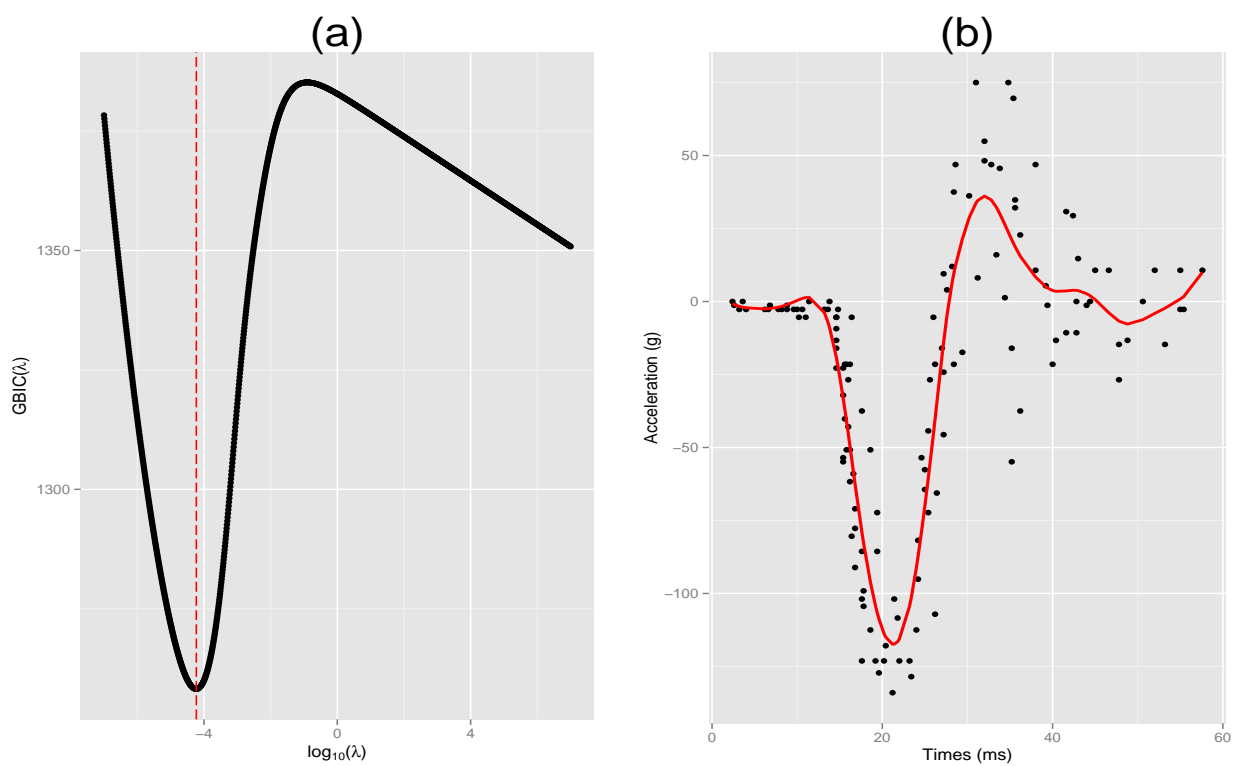


FIGURE 2.15: (a) $\log_{10}(\lambda)$ against $\text{GBIC}(\lambda)$; (b) *Motorcycle* Data smoothed using *B-splines* basis functions with $K = 40$ and GBIC criterion yielding $\hat{\lambda} = 10^{-4.23}$ & $\hat{\sigma}^2 = 462.2833$

Although the *Generalized Cross-Validation* criterion is widely used for the regularization parameter selection, the computational time is very large and high variability and tendency to undersmooth are not negligible in the analysis of Functional Data (Matsui et al., 2009). Table 2.5 illustrates that argument with a high value for the GCV Mean Square Errors (MSE) as well as its estimated $\hat{\sigma}_{GCV}^2$; it is important to point out that the number of basis functions is the same for all model criteria.

TABLE 2.5: Summary of the model selection applied on the `Motorcycle` Data smoothed using B-splines basis functions with $K = 40$

	$\log_{10}(\hat{\lambda})$	$\hat{\sigma}^2$	MSE
GCV	-4.08	2116.593	468.1117
GIC	-4.22	462.5911	462.5911
mAIC	-4.12	466.3463	466.3463
GBIC	-4.23	462.2833	462.2833

2.4.5 The optimal number K of Basis Functions

Choosing the optimal number K of basis functions is an important task when converting the discrete observations into Functional Data. The larger K the better the fit to the data, but at the same time the risk of fitting noise or variation that should not be ignored. On the other hand, if K is too small, some important aspects of the smooth function might be disregarded when trying to estimate the function (Ramsey and Silverman, 2005). One of the main reasons for smoothing is to reduce the influence of noise as well as to capture meaningful regularities on the estimates. The idea of the penalization is to rather overfit the data and then penalize to obtain a bias-variance trade-off. The methods for model selection (mentioned above) may offer some guidance in the choice of the optimal K , however for each value of the number of basis functions there is an optimal value for $\hat{\lambda}$ & $\hat{\sigma}^2$.

2.5 Functional Descriptive Statistics

One of the most important parts in data analysis is the exploratory part: Estimating means and standard deviations. Because the functional nature of the data, the associated descriptive statistics are therefore functional.

2.5.1 Mean & Variance functions

Estimating the *Mean Function* based on discretely sampled noisy observations is one of the most basic problems in Functional Data Analysis. The *Mean Function* is a simple analogue of the classical mean for univariate data. It can be calculated by averaging the functions point-wise across the replications, since Functional Data Analysis sees each curve as a distinct datum itself. The mean function is defined as $\nu_{\mathcal{X}}(t) = \mathbb{E}(X(t))$, $\forall t \in \mathcal{T}$. The sample mean curve is:

$$\bar{X}(t) = \frac{1}{N}(X_1(t) + \cdots + X_N(t)), \quad \forall t \in \mathcal{T} \quad (2.44)$$

where N is the number of curves or replications and $X_i(t)$ is the i -th function evaluated at time t . Below is a plot illustrating the concept of *Functional Mean* applied on the **Canadian Weather** dataset from Ramsay and Silverman (2005). The *Functional Mean* is calculated for five weather stations namely *St. Johns*, *Halifax*, *Sydney*, *Yarmouth* & *Charlottville* represented using a *Fourier Basis* expansion with $K = 65$:

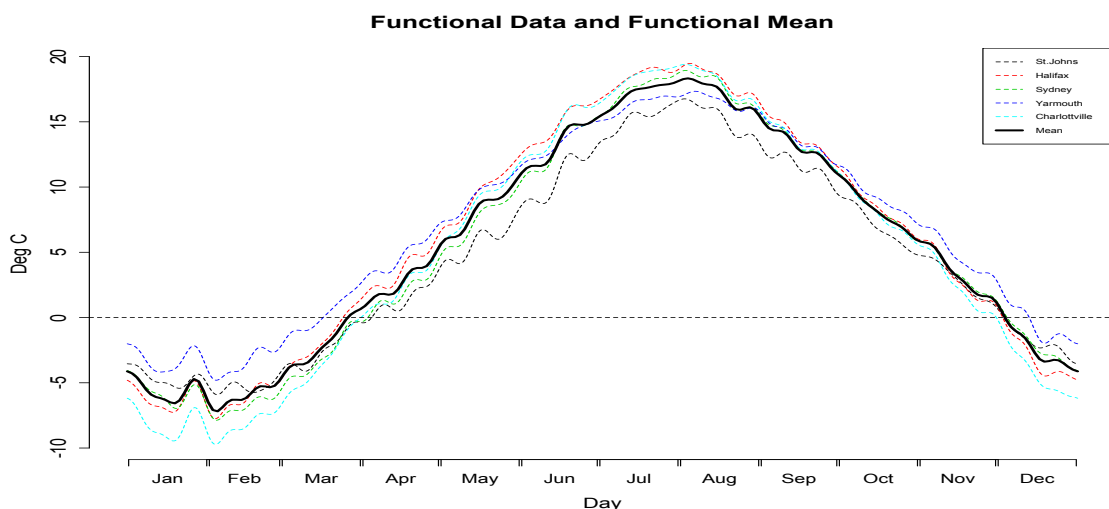


FIGURE 2.16: Canadian Weather data: Mean Curve ¹.

Likewise, the estimation of the *Functional Variance* is very similar to the classical variance for univariate data. It is defined as $\sigma_{\mathcal{X}}^2(t) = \mathbb{E}[(X(t) - \nu_{\mathcal{X}}(t))^2]$, $\forall t \in \mathcal{T}$. The sample variance curve is:

$$\mathbf{var}_{\mathcal{X}}(t) = \frac{1}{N} \sum_{i=1}^N [X_i(t) - \bar{X}(t)]^2 \quad (2.45)$$

and the standard deviation function is the square root of the variance function.

2.5.2 Covariance and Correlation functions

The *covariance function* summarizes the dependence of records across different argument values. We define $\Gamma_{\mathcal{X}}$ to be the covariance function

$$\Gamma_{\mathcal{X}}(t_1, t_2) = \mathbb{E}[(X(t_1) - \nu_{\mathcal{X}}(t_1))(X(t_2) - \nu_{\mathcal{X}}(t_2))], \quad \forall t_1, t_2 \in \mathcal{T},$$

and $\hat{\Gamma}$ to be the sample covariance function

$$\hat{\Gamma}(t_1, t_2) = \frac{1}{N} \sum_{i=1}^N \{X_i(t_1) - \bar{X}(t_1)\} \{X_i(t_2) - \bar{X}(t_2)\}, \quad \forall t_1, t_2 \in \mathcal{T}. \quad (2.46)$$

The associated *correlation function* is

$$\mathbf{corr}_{\mathcal{X}}(t_1, t_2) = \frac{\hat{\Gamma}_{\mathcal{X}}(t_1, t_2)}{\sqrt{\mathbf{var}_{\mathcal{X}}(t_1)\mathbf{var}_{\mathcal{X}}(t_2)}} \quad (2.47)$$

2.6 Parallel Computing using R

Dealing with large data sets has become common practice when working with Functional Data. Statisticians usually find the need to perform some operations repeatedly for model selection or simply to execute functions with multiple arguments. Repeated executions can be done manually, but it becomes quite tedious to execute repeated operations even with the use of command line editing (Leach, 2014). Nowadays, all computers are equipped with multicore processors which allow splitting tasks across a number of cores for execution and therefore reducing computation time.

2.6.1 Parallel Backends

Running codes in parallel is not a default feature of R, so executing parallelism requires to first make the desired number of cores available to R by registering a *parallel backend* which effectively creates a cluster to which computations can be sent to. Several packages have been developed to handle this process:

- **doMC** (Revolution Analytics and Steve Weston, 2014a)
- **doSNOW** (Revolution Analytics and Steve Weston, 2014c)
- **doParallel** (Revolution Analytics and Steve Weston, 2014b)

Creating a cluster is done using the following lines of codes:

```
suppressPackageStartupMessages(library(doParallel))
detectCores() # how many cores are available
workers <- makeCluster(6) # create a cluster with 6 cores
registerDoParallel(workers) # register cluster
getDoParWorkers() # Number of cores that will be used
```

2.6.2 Using foreach

The **foreach** package provides a new looping construct for processing R codes repeatedly (Revolution Analytics and Steve Weston, 2014d). It supports *parallel execution*, in other words it can process replicated operations on multiple cores on the computer or on multiple nodes of a cluster.

For illustrations purpose, consider the temperature data from the **Aemet** dataset in R. Given a set of values for K ranging from 5 to 360, the *Generalized Cross-Validation* is computed for each K and the time taken to process the R-script with and without **foreach** is recorded.

Without foreach

```
#### Temperature
data(aemet, package = "fda.usc")
tt <- aemet$temp$argvals
temp <- aemet$temp$data
cent.temp <- apply(X = temp, MARGIN = 2, FUN = scale, scale=FALSE)
m <- seq(5, 360)
temp_gcv <- rep(0, length(m))
count <- 0
ptime <- system.time(for (i in m){
+   count <- count + 1
+   temp_gcv[count] = GCV.Gauss_bs(data = t(cent.temp), tt = tt, m = i)
+   cat("basis function ", i, "\n")
+ }) [3]
ptime # time in seconds
elapsed
35.38
```

Without using `doParallel` and `foreach`, the for-loop is executed in 35.38 seconds.

With foreach

```
detectCores()
[1] 8
workers <- makeCluster(8)
registerDoParallel(workers)
getDoParWorkers()

#### Temperature
data(aemet, package = "fda.usc")
tt <- aemet$temp$argvals
temp <- aemet$temp$data
cent.temp <- apply(X = temp, MARGIN = 2, FUN = scale, scale=FALSE)
m <- seq(5, 360)
temp_gcv <- rep(0, length(m))
ptime <- system.time(foreach (i = icount(length(m)), .combine = 'c') %dopar% {
+   GCV.Gauss_bs(data = t(cent.temp), tt = tt, m = m[i])
+ }) [3]
ptime # time in seconds
elapsed
6.27
```

Using `doParallel` and `foreach` reduced the running time to 6.27 seconds. In other words, an appropriate utilization of parallel computing helps to save time. Note that the time taken with 8 cores did not reduce eight-fold as $6.27 \times 8 = 50.16$ seconds. Additional time is taken for splitting the iterations and combining the final result, however for the user to complete section of the code executed more than five times faster. In the above example, the function `GCV.Gauss_bs` calculates the *Generalized*

Cross-Validation for the centered Temperature data evaluated for a set of K basis functions.

2.7 High Performance Computing (HPC)

In practice, executing an algorithm that runs over a large number of iterations delays the output. In other words, computational methods in science require lots of processing time. One way to overcome this obstacle is to aggregate computing power in a way that delivers much higher performance than a single desktop computer or workstation. These are very exotic computers by virtue of the elements inside them, and the scale at which they are built. The University of Cape Town via the [Information and Communication Technology Services \(ICTS\)](#) offers such facilities with the aim of supporting the scientific community.

This section serves as a mini-manual to access the UCT ICTS HPC cluster II for scientists using Windows as operating system. For further information the interested readers should access the service via the following link:

<http://srvslnhpc001.uct.ac.za/eresearch/>. A list of available softwares that are on the clusters by default can be found at

http://srvslnhpc001.uct.ac.za/eresearch/?page_id=73

2.7.1 Connecting to the UCT ICTS HPC cluster

The following softwares must be downloaded in order to facilitate the access to the UCT ICTS HPC cluster as well as file transfers, scripts editing, job submissions:

- PuTTY which is a free implementation of SSH for Windows platform. The download page for PuTTY is <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.
- WinSCP is an open source free SFTP Client, FTP Client, WebDAV client and SCP client for Windows. Its main function is file transfer between a local computer and a remote computer, the HPC cluster to be more precise. The download page for WinSCP is <http://sourceforge.net/projects/winscp/>.

Connecting with WinSCP

WinSCP allows the user to navigate through the folders in the cluster as well as to copy files or folders from a local computer to the cluster and vice versa. Figure 2.17 shows the window where the following details would have to be typed in order to login:

- **Host name:** `hex.uct.ac.za`
- **User name:** campus ID number
- **Password:** supplied by the HPC cluster administrator.

After the abovementioned details have been provided, WinSCP prompts the user to a new window as in figure 2.18:

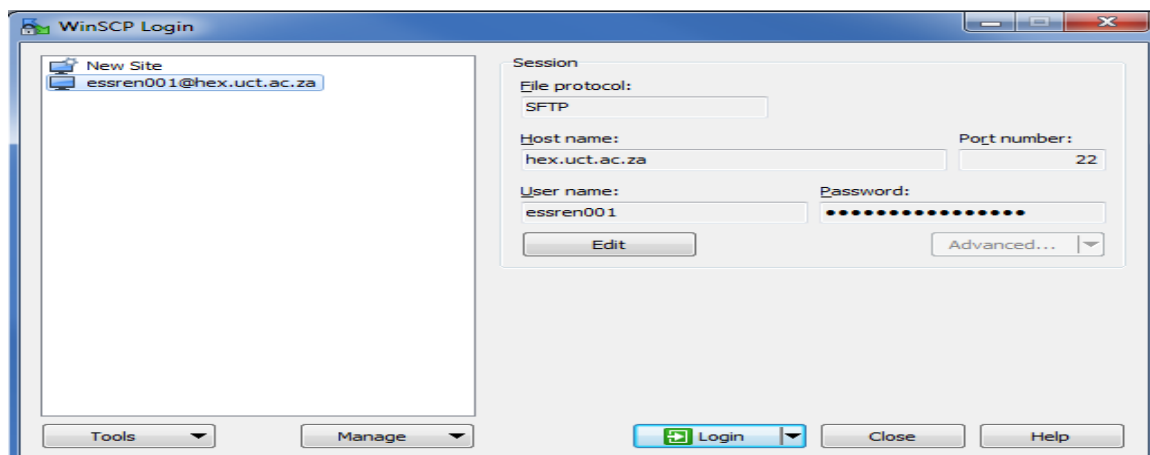


FIGURE 2.17: Login Window

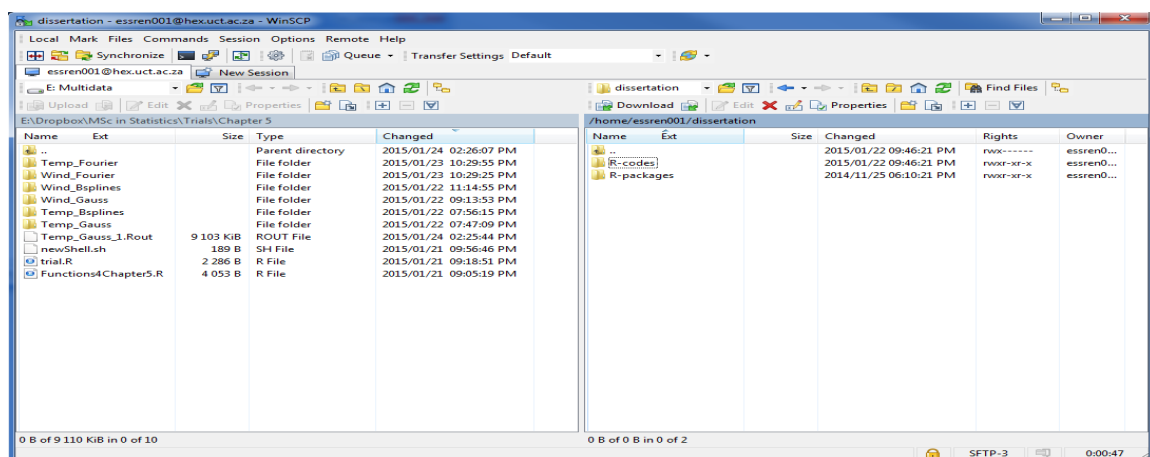


FIGURE 2.18: WinSCP Interface

Connecting with PuTTY

Through PuTTY, users access the cluster using an SSH protocol. SSH (which stands for 'secure shell') ensures a highly protected connection against eavesdropping, hijacking and other attacks. Connecting to the UCT HPC cluster using PuTTY only requires the user to enter the **Host name** `hex.uct.ac.za` as it is shown on Figure 2.19. Once the personal profile details have been entered, PuTTY prompts the user to a new window as in Figure 2.20 where the user should type their *campus_id_number* and *password*. Then once the abovementioned steps are executed, PuTTY prompts the

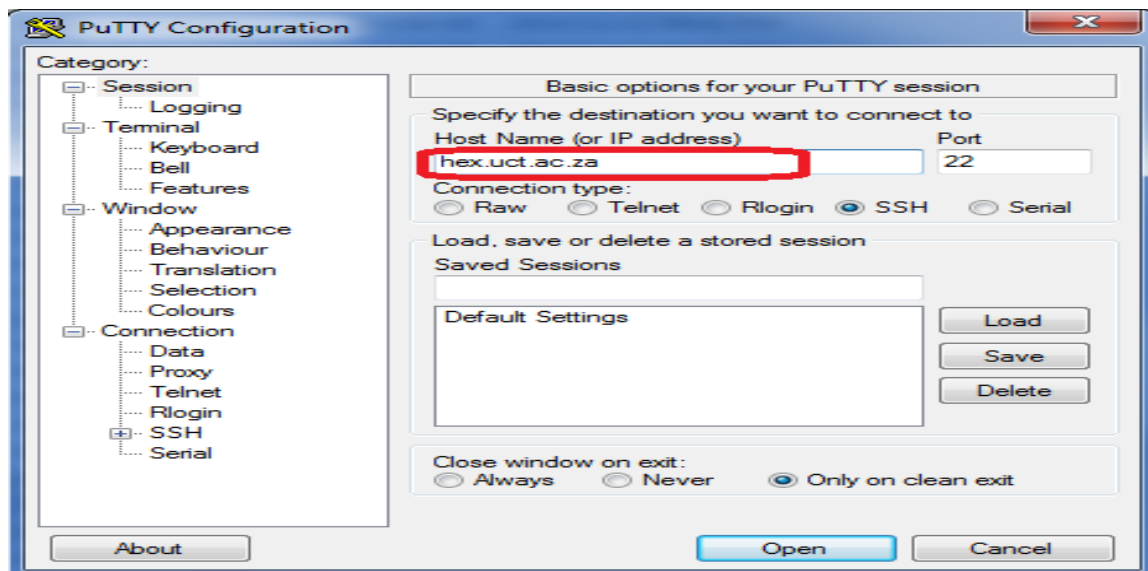


FIGURE 2.19: Configuration Window

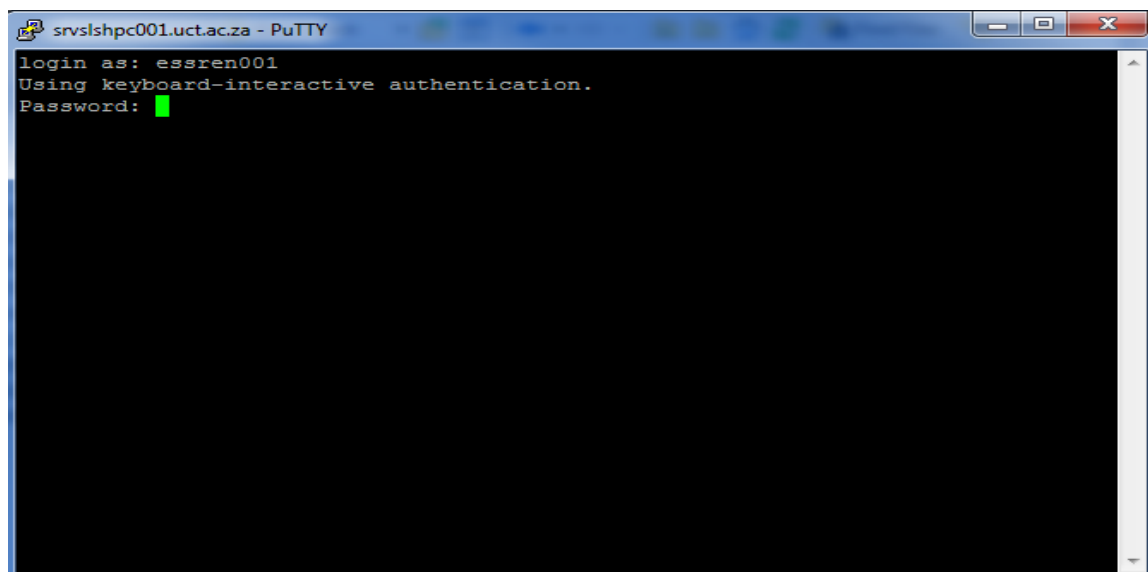
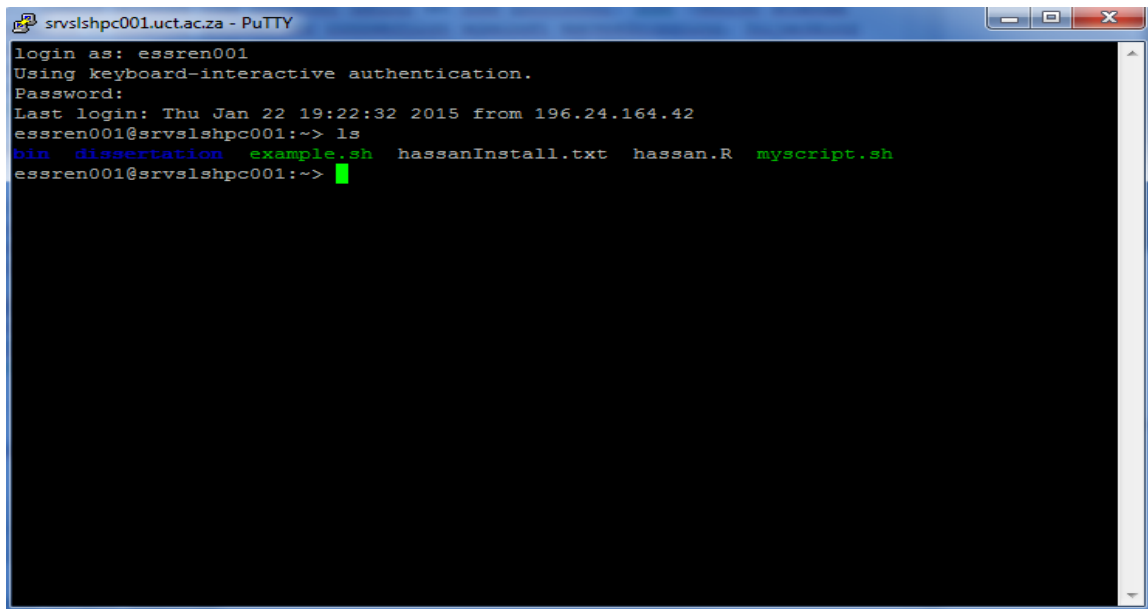


FIGURE 2.20: PuTTY Login

user to figure 2.21.

A screenshot of a PuTTY terminal window titled 'srvslshpc001.uct.ac.za - PuTTY'. The terminal shows a login session for user 'essren001'. The prompt is 'login as: essren001'. The user is prompted for a password, and the session is established. The terminal shows the last login time as 'Thu Jan 22 19:22:32 2015 from 196.24.164.42'. The user runs the command 'ls' and the output shows the contents of the home directory: 'bin', 'dissertation', 'example.sh', 'hassanInstall.txt', 'hassan.R', and 'myscript.sh'. The prompt is 'essren001@srvslshpc001:~>'.

```
login as: essren001
Using keyboard-interactive authentication.
Password:
Last login: Thu Jan 22 19:22:32 2015 from 196.24.164.42
essren001@srvslshpc001:~> ls
bin  dissertation  example.sh  hassanInstall.txt  hassan.R  myscript.sh
essren001@srvslshpc001:~>
```

FIGURE 2.21: Inside the cluster

2.7.2 Interacting with the Cluster

At this point, it is possible to perform various operations using command lines. A typical set of operations that can be done on the cluster is the following:

- `ls` list information about files in current directory
- `cd` change directory
- `cp` copy and paste
- `cd ~` home directory
- `mkdir` create a new directory
- `mv` move file
- `rm` delete files
- `vim` text editor
- `qstat` request the status of jobs, queues
- `qsub` job submission to the cluster.

An exhaustive list of all the command lines by accessing the help file: `man ls`

Shell Scripts

A shell script is a plain text file with **Bash** commands that is interpreted by a shell process. Below is an example of a shell script where the user can specify the number of nodes (computers) and the number of cores per node:

```
#PBS -N filename
#PBS -q UCTlong
#PBS -l nodes=1:ppn=1:series600
cd /home/essren001/dissertation/R-codes
mpirun -hostfile $PBS_NODEFILE /opt/exp_soft/R-3.0.2/bin/R --slave CMD BATCH
    filename.R
```

Before submitting a job to the cluster, the user must ensure that both the shell script and the R-file are in the same folder. The R-script that is processed looks like this

```
.libPaths(c(.libPaths(), "/home/essren001/dissertation/R-packages"))

options(scipen = 999)
suppressPackageStartupMessages(library(fda))
suppressPackageStartupMessages(library(fda.usc))
suppressPackageStartupMessages(library(matrixcalc))

setwd("/home/essren001/dissertation/R-codes")

data(aemet, package = "fda.usc")
dtc <- aemet$temp$argvals
temp <- as.data.frame(aemet$temp$data, row.names=F)
cent.temp <- data.frame(apply(X = temp, MARGIN = 2, FUN = scale, scale=FALSE))
range.dtc <- aemet$temp$rangeval

#####

source("Functions4Chapter5.R")

#####

##### Split the dataset in training set (70%) and test set (30%)
df.temp <- splitdf(dataframe = cent.temp, seed = 808, split = 70)
train.temp <- df.temp$trainset

##### Basis function: Gaussian basis
#### Temperature
K1 = seq(5,363) # number of basis functions
nK1 = length(K1)
y <- as.matrix(train.temp[1,]) # station 1
loglam <- seq(-10,10,0.01) # lambda values
nlam <- length(loglam)
GIC_mat <- matrix(0,nK1,nlam)

for(i in 1:nK1){
  B = Gaussian_bsplines(tt = dtc,m = K1[i]) # basis functions
  n = K1[i]
```

```

for(k in 1:nlam){
  ob <- Pen_Max_Likelihood(B = B,n = n,lambda = loglam[k],y = y)
  GIC_mat[i,k] <- gic_fun(y = y,ob = ob,n = n)
  cat("basis function:",K1[i],i,"lambda ",loglam[k],"\n")
}
}
save.image("R_Output.RData")

```

2.8 Closing Comments

In this chapter, the tools for converting high frequency observed data points to continuous functions were discussed. If the observed points exhibit periodic features then the *Fourier Basis* functions are suited for smoothing the data. For non-periodic data, the *B-Splines Basis* functions are recommended to smooth the data. Other basis functions such as *Gaussian Basis* and *Haar Wavelets* have the ability to fit both periodic and non-periodic data as long as an appropriate number of basis functions and the optimal smoothing parameter are determined. Three model selections were studied: *Least Square* method, *Maximum Likelihood* method and *Penalized Maximum Likelihood* method. Once a model is selected, it needs to be evaluated. Four kinds of model criteria were discussed in that regard: *Generalized Cross-Validation*, *Generalized Information Criteria*, *modified Akaike Information Criteria* and *Generalized Bayesian Information Criteria*. The resulting functions mimicking the random trajectory of the observed data are the Functional Data. Functional descriptive statistics such as the *Functional Mean*, *Functional Variance* can be derived from the Functional Data. The last sections of this chapter touched on important aspects related to computing Functional Data. The use of parallel computing seems to be viable solution to the computationally intensive algorithms.

In the next chapter, a theoretical discussion of Functional Data Analysis is presented by delving into the Mathematics of Functional Data Analysis.

Chapter 3

Mathematics of Functional Data Analysis

Chapter 2 introduced the some relevant tools used when one has to perform analysis in the Functional Data framework. In this Chapter, the focus will be on providing mathematical foundations to understand the connection between Functional Analysis and Functional Data Analysis. One of the most important results of this Chapter will be the *Karhunen-Loève* Theorem which provides solid explanations to the existence of equation (2.1). This chapter is organised as follows: **section 3.1** provides some important definitions and theorems of Hilbert Space; **section 3.2** cements the concept of operators in Hilbert Space; **section 3.3** introduces important definitions and theorems of the L^2 Space; **section 3.4** recalls key results related to stochastic processes, **section 3.5** delves into the *Karhunen-Loève* Theorem and **section 3.6** summarises the key concepts of this Chapter.

Throughout this Chapter, it is assumed that the all vector spaces considered are over the field $\mathbb{K} = \mathbb{R}$ or \mathbb{C} where appropriate.

3.1 Hilbert Spaces

This section serves a quick recall of some important results and definitions from Topology focusing on Hilbert spaces that will be used later in the chapter.

Definition 3.1.1. Let H be a vector space. An *inner product* on H is a function $\langle \cdot, \cdot \rangle : H \times H \rightarrow \mathbb{K}$ such that for every $u, v, w \in H$ and $\alpha, \beta \in \mathbb{K}$,

1. $\langle u, u \rangle \geq 0$ and $\langle u, u \rangle = 0$ if and only if $u = 0_H$
2. $\langle u, v \rangle = \overline{\langle v, u \rangle}$ (where \bar{u} is defined as the conjugate of a vector u)
3. $\langle \alpha u + \beta w, v \rangle = \alpha \langle u, v \rangle + \beta \langle w, v \rangle$.

The pair $(H, \langle \cdot, \cdot \rangle)$ is called an *inner product* or *pre-Hilbert space*.

Theorem 3.1.2. If $(H, \langle \cdot, \cdot \rangle)$ is an inner product space, then for every $u, v \in H$

$$|\langle u, v \rangle|^2 \leq \langle u, u \rangle \langle v, v \rangle. \quad (3.1)$$

The above inequality is known as the *Cauchy-Schwartz inequality* and is used to show the following:

Theorem 3.1.3. If $(H, \langle \cdot, \cdot \rangle)$ is an inner product space, then the function $\|\cdot\| : H \rightarrow \mathbb{R}$ defined by

$$\|v\| := \sqrt{\langle v, v \rangle} \quad \forall v \in H \quad (3.2)$$

is a norm on H . This makes $(H, \langle \cdot, \cdot \rangle)$ a normed space and a metric space.

A sequence $\{v_n\}$ in a normed space H is said to converge to $v \in H$ if for every $\epsilon > 0$ there exists $N \in \mathbb{N}^+$ such that for every $n \geq N$, $\|v_n - v\| \leq \epsilon$.

Definition 3.1.4. A sequence $\{v_n\}$ in a normed space H is a *Cauchy sequence* if for every $\epsilon > 0$ there exists $N \in \mathbb{N}^+$, where if $n, m \geq N$, then $\|v_n - v_m\| < \epsilon$.

It can be proven that every convergent sequence is *Cauchy*, but the converse does not hold in general.

Definition 3.1.5. An inner product space H is *complete* if for every Cauchy sequence $\{v_n\}$ in H , there exists $v \in H$ such that $v_n \rightarrow v$; i.e. H is complete if every Cauchy sequence in H converges to an element of H . A subset V of H is complete if every Cauchy sequence in V converges to an element of V .

A complete inner product space is called a *Hilbert space*. An analogous result is also true for normed spaces in general; complete normed spaces are called *Banach spaces*.

3.2 Operators in a Hilbert Space

Definition 3.2.1. A Hilbert space with an inner product $\langle \cdot, \cdot \rangle$ is separable if and only if there exists a countable set $U = \{u_n : n \in \mathbb{N}\}$ such that $\bar{U} = H$, in other words if and only if it has a countable dense subset. \bar{U} is the closed set of U .

Consider a separable Hilbert space H with inner product $\langle \cdot, \cdot \rangle$ which generates the norm $\|\cdot\|$, and denote by \mathcal{L} the space of bounded (continuous) linear operators on H with the norm defined as

$$\|\Psi\|_{\mathcal{L}} = \sup\{\|\Psi(x)\|_{\mathcal{L}} : \|x\| \leq 1\} < \infty. \quad (3.3)$$

An operator $\Psi : H \rightarrow H$ is *compact* if the image of every bounded subset of H is relatively compact. It is well known from the spectral theory of compact operators that if H is a separable Hilbert space then there exist two orthonormal sequences $\{e_n\}$ and $\{f_n\}$ and a real sequence $\{\lambda_j\}$ converging to zero such that

$$\begin{aligned} \Psi(x) &= \sum_{i=1}^{\infty} \lambda_i \langle x, e_i \rangle f_i, \quad x \in H \\ &= \lim_{m \rightarrow \infty} \sum_{i=1}^m \lambda_i \langle x, e_i \rangle f_i, \quad x \in H. \end{aligned} \quad (3.4)$$

The λ_i may be assumed to be positive because one can replace f_i by $-f_i$ if needed. The existence of representation (3.4) is equivalent to the condition Ψ maps every bounded set into a compact set. Equation (3.4) is called the *singular value decomposition* of Ψ . A compact operator satisfying equation (3.4) with the property that $\sum_{i=1}^{\infty} \lambda_i^2 < \infty$ is said to be a *Hilbert-Schmidt operator*. Consider \mathcal{S} the space of Hilbert-Schmidt operators. \mathcal{S} is said to be a separable Hilbert space with the scalar product if

$$\langle \Psi_1, \Psi_2 \rangle_{\mathcal{S}} = \sum_{i=1}^{\infty} \langle \Psi_1(e_i), \Psi_2(e_i) \rangle \quad (3.5)$$

where $\{e_i\}$ is an arbitrary orthonormal basis. The value of (3.5) does not depend on the chosen orthonormal basis. An operator $\Psi \in \mathcal{L}$ is said to be :

- symmetric if $\langle \Psi(x), y \rangle = \langle x, \Psi(y) \rangle, \quad x, y \in H,$
- positive semi-definite if $\langle \Psi(x), x \rangle \geq 0, \quad x \in H$

A symmetric positive semi-definite Hilbert-Schmidt operator Ψ admits the decomposition

$$\Psi(x) = \sum_{i=1}^{\infty} \lambda_i \langle x, e_i \rangle e_i, \quad x \in H \quad (3.6)$$

with the orthonormal set $\{e_i\}$ which are the eigenfunctions of Ψ , i.e. $\forall e_i, \exists \lambda_i \in \mathbb{K}$ such that $\Psi(e_i) = \lambda_i e_i$. Using the Zorn's Lemma, it can be shown that the set $\{e_i\}$ can be extended to a basis by adding a complete orthonormal system in the orthogonal complement of the subspace spanned by the original $\{e_i\}$.

Definition 3.2.2. Let H be a Hilbert space with inner product $\langle \cdot, \cdot \rangle : H \times H \rightarrow \mathbb{K}$. A linear operator $\Psi : H \rightarrow H$ is called *self-adjoint* if

$$\langle \Psi(x), y \rangle = \langle x, \Psi(y) \rangle, \quad \forall x, y \in H. \quad (3.7)$$

Compact self-adjoint operators on infinite dimensional Hilbert spaces resemble many properties of the symmetric matrices. The spectral decomposition of a compact self-adjoint operator is given by the following:

Theorem 3.2.3. Let H be a Hilbert space and let $\Psi : H \rightarrow H$ be a compact self-adjoint operator. Then, H has an orthonormal basis $\{e_i\}$ of eigenvectors of Ψ corresponding to eigenvalues λ_i . In addition, the following points hold:

1. The eigenvalues λ_i are real having zero as the only point of accumulation.
2. The eigenspaces corresponding to distinct eigenvalues are mutually orthogonal.
3. The eigenspaces corresponding to non-zero eigenvalues are finite dimensional.

In the case of a positive compact self-adjoint operator, it is known that the eigenvalues are non-negative. Therefore, the eigenvalues may be ordered as follows

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$$

3.3 The Space L^2

The space $L^2 = L^2(\mathcal{T})$ is the set of measurable real-valued functions x defined on $\mathcal{T} = [a, b]$ satisfying $\int_{\mathcal{T}} x^2(t)dt < \infty$. The space L^2 is a separable Hilbert space with inner product

$$\langle x, y \rangle = \int_{\mathcal{T}} x(t)y(t)dt. \quad (3.8)$$

An important class of operators in L^2 are the integral operators defined by

$$\Psi(x)(t) = \int_{\mathcal{T}} \Psi(t, s)x(s)ds, \quad x \in L^2 \quad (3.9)$$

where $\Psi : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{K}$ is the real kernel. Such operators are Hilbert-Schmidt if and only if

$$\iint_{\mathcal{T}} \Psi^2(t, s)dt ds < \infty, \quad (3.10)$$

in which case

$$\|\Psi\|_S^2 = \iint_{\mathcal{T}} \Psi^2(t, s)dt ds. \quad (3.11)$$

The operator is symmetric if $\Psi(s, t) = \Psi(t, s)$ and positive semi-definite if $\iint_{\mathcal{T}} \Psi(t, s)x(t)x(s)dt ds \geq 0$, $\forall x \in L^2$. In this case there is an *orthonormal basis* $\{e_i\}$ of $L^2(\mathcal{T})$ consisting of eigenfunctions of Ψ such that the corresponding sequence of eigenvalues $\{\lambda_i\}$ is nonnegative. It also follows that Ψ has the representation

$$\Psi(t, s) = \sum_{i=1}^{\infty} \lambda_i e_i(t)e_i(s) \text{ in } L^2(\mathcal{T} \times \mathcal{T}) \quad (3.12)$$

If Ψ is continuous, the above expansion holds for all $s, t \in \mathcal{T}$ and the series converges absolutely and uniformly on $\mathcal{T} \times \mathcal{T}$. This result is known as *Mercer's Theorem*.

3.4 Stochastic Processes

It is assumed that $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space, where Ω is a sample space, \mathcal{F} is an appropriate σ -algebra on Ω and \mathbb{P} is probability measure. A random variable X on $(\Omega, \mathcal{F}, \mathbb{P})$ is an $\mathcal{F}/\mathcal{B}(\mathbb{R})$ -measurable mapping $(\Omega, \mathcal{F}, \mathbb{P})$ on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$, where $\mathcal{B}(\mathbb{R})$ is the *Borel* set on \mathbb{R} . The expectation and variance of a random variable X is denoted by,

$$\mathbb{E}[X] := \int_{\Omega} X(\omega)d\mathbb{P}(\omega), \quad \text{Var}[X] := \mathbb{E}[(X - \mathbb{E}[X])^2].$$

$L^2(\Omega, \mathcal{F}, \mathbb{P})$ denotes the Hilbert space of real valued square integrable random variables on Ω :

$$L^2(\Omega, \mathcal{F}, \mathbb{P}) = \left\{ X : \Omega \rightarrow \mathbb{R} : \int_{\Omega} |X(\omega)|^2 d\mathbb{P}(\omega) < \infty \right\},$$

with inner product $\langle X, Y \rangle = \mathbb{E}[XY] = \int_{\Omega} XY d\mathbb{P}$ and norm $\|X\| = \langle X, X \rangle^{1/2}$.

Let $\mathcal{T} = [a, b] \subseteq \mathbb{R}$, a stochastic process is a mapping $X : \mathcal{T} \times \Omega \rightarrow \mathbb{R}$, such that $X(t, \cdot)$ is measurable for every $t \in \mathcal{T}$; alternatively a stochastic process is a family of random variables, $X_t : \Omega \rightarrow \mathbb{R}$ with $t \in \mathcal{T}$.

A stochastic process is called centered if $\mathbb{E}[X_t] = 0$ for all $t \in \mathcal{T}$. Let $\{Y_t\}_{t \in \mathcal{T}}$ be an arbitrary stochastic process such that

$$Y_t = \mathbb{E}[X_t] + X_t$$

where $X_t = Y_t - \mathbb{E}[X_t]$. Without loss of generality, the attention is on centered stochastic processes.

Definition 3.4.1. The autocorrelation function of a stochastic process $\{X_t\}_{t \in \mathcal{T}}$ is given by $R_X : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}$ such that

$$R_X(s, t) = \mathbb{E}[X_s X_t], \quad s, t \in \mathcal{T}.$$

Lemma 3.4.2. A stochastic process $\{X_t\}_{t \in \mathcal{T}}$ is mean-square continuous if and only if its autocorrelation function R_X is continuous on $\mathcal{T} \times \mathcal{T}$.

3.5 Karhunen-Loève Expansion

It is assumed that $X : \mathcal{T} \times \Omega \rightarrow \mathbb{R}$ is a centered mean-square continuous stochastic process such that $X \in L^2(\mathcal{T} \times \Omega)$. It has been mentioned in section 3.2 that a compact positive self-adjoint operator $\Psi : L^2(\mathcal{T}) \rightarrow L^2(\mathcal{T})$ has a complete set of $\{e_i\}$ in $L^2(\mathcal{T})$ and real eigenvalues $\{\lambda_i\}$ such that:

$$\Psi e_i = \lambda_i e_i. \tag{3.13}$$

Moreover, since Ψ is positive, the eigenvalues λ_i are non-negative. The stochastic process X is assumed to be square integrable on $\mathcal{T} \times \Omega$ and the basis $\{e_i\}$ of $L^2(\mathcal{T})$

can be used to expand X_t as follows:

$$X_t = \sum_i^{\infty} x_i e_i(t), \quad x_i = \int_{\mathcal{T}} X_t e_i(t) dt \quad (3.14)$$

The above equation is to be understood in mean square sense. It can be noted that a realization \hat{X} of the stochastic process X admit the expansion

$$\hat{X} = \sum_i^{\infty} x_i e_i$$

where the convergence is in $L^2(\mathcal{T} \times \Omega)$. The above results lead to the *Karhunen-Loève Theorem*.

Theorem 3.5.1 (Karhunen-Loève). *Let $X : \mathcal{T} \times \Omega \rightarrow \mathbb{R}$ be a centered mean-square continuous stochastic process with $X \in L^2(\mathcal{T} \times \Omega)$. There exist a basis $\{e_i\}$ of $L^2(\mathcal{T})$ such that for all $t \in \mathcal{T}$,*

$$X_t = \sum_{i=1}^{\infty} x_i e_i(t), \quad \forall t \in \mathcal{T} \quad (3.15)$$

where the coefficients x_i are given by $x_i(\omega) = \int_{\mathcal{T}} X_t(\omega) e_i(t) dt$ and satisfy the following points:

1. $\mathbb{E}[x_i] = 0, \quad \forall i \in \mathbb{N};$
2. $\mathbb{E}[x_i x_j] = \delta_{ij} \lambda_j, \quad \forall i, j \in \mathbb{N};$
3. $\text{Var}[x_j] = \lambda_i, \quad \forall i \in \mathbb{N},$

$$\text{with } \delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

An important point is that since the random coefficient x_j of the *Karhunen-Loève Expansion* are uncorrelated, the variance of X_t is simply the sum of the variances of the individual eigenvalues (under the assumptions of the Beppo-Levi monotone convergence theorem):

$$\begin{aligned} \text{Var}[X_t] &= \text{Var} \left[\sum_{i=1}^{\infty} x_i e_i(t) \right] \\ &= \sum_{i=1}^{\infty} e_i^2(t) \text{Var}[x_i] \\ &= \sum_{i=1}^{\infty} \lambda_i e_i^2(t). \end{aligned}$$

Integrating the above result over \mathcal{T} and using the orthonormality of $\{e_i\}$, the total variance of the process is:

$$\int_D \text{Var}[X_t] dt = \sum_{i=1}^{\infty} \lambda_i. \quad (3.16)$$

In particular, the total variance of the N -truncated approximation which is $\sum_{i=1}^K \lambda_i$ explains $\frac{\sum_{i=1}^K \lambda_i}{\sum_{i=1}^{\infty} \lambda_i}$ of the total variance of the stochastic process X_t . The optimal number of eigenfunctions is the smallest value $K \in \mathbb{N}$ such that $\frac{\sum_{i=1}^K \lambda_i}{\sum_{i=1}^{\infty} \lambda_i} \geq \alpha$, where $0 \leq \alpha \leq 1$.

3.6 Closing Comments

This Chapter introduced some key results from Functional Analysis and more specifically results related to Hilbert Space and L^2 Space, and established the *Karhunen-Loève* Theorem. The *Karhunen-Loève* Theorem, also known as the *Kosambi-Karhunen-Loève* Theorem explained the reason why any stochastic process can be represented as an infinite linear combination of eigenfunctions which are elements of the L^2 Space on a bounded interval. Interested readers can consult Gohberg et al. (1990) for further indications on the topic. Furthermore, the infinite linear combination of eigenfunctions can be written as a finite sum of basis functions. In most cases, Statisticians replace the eigenfunctions and eigenvalues by basis functions $\phi_k(t)$ and coefficients $c_k \forall k = 1, 2, \dots, K$ (respectively) as seen in equation (2.1).

Now that the foundations and tools of Functional Data Analysis have been established, the next Chapter will introduce Functional Linear Regression Modeling which is an extension of Multivariate Regression Modeling.

Chapter 4

Functional Linear Regression Modeling (FLRM)

This chapter will review some key concepts related to the *Functional Linear Regression* model. Like in *Multivariate Analysis*, *Functional Linear Regression* model has appeared to be extremely useful in a broad range of applications including Bioscience and Time Series. A typical *Functional Linear Regression* model intends to explore the variability of a scalar continuous (functional) response while considering how much of its variation is explainable by other variables.

Linear regression models can be functional in one or both of two ways:

- The dependent or response variable is functional;
- One or more of the independent variables or covariates are functional.

Clearly, the functional-response case is an extension of the multivariate-response case with vectors converted into functions. The main change is that the regression coefficients now become regression functions with values $\beta_j(t)$ or $\beta_j(t, s)$ depending on the nature of the problem. Although the main focus of this chapter is on functional response predicted by one or more functional covariates, a preliminary look is done for all cases where the response variables are *scalar* and *multivariate*.

It should be noted that all inferential tools for *Functional Linear Regression* models have been developed under the assumption that the covariate/response pairs are independent.

4.1 Preliminary Cases

The aim of this section is to predict a scalar/multivariate response from one or more functional covariates. Since *Functional Linear Regression Modeling* has its roots from *multivariate multiple regression modelling*, the final result of all derivations have the form:

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\epsilon} \quad (4.1)$$

4.1.1 Scalar response and Functional Independent Variables

Let $\{Y_i, i = 1, \dots, N\}$ be an N -vector of scalar responses and $\{X_{im}(t), m = 1, \dots, M\}$ are M functional predictors. Using the definitions from Chapter 2, the functions $X_{im}(t)$ can be obtained using the smoothing techniques. The regression model that evaluates the relationship between the vector of scalar responses and the functional covariates is given by

$$Y_i = \beta_0 + \sum_{m=1}^M \int_{\mathcal{T}_m} X_{im}(t) \beta_m(t) dt + \epsilon_i, \quad \forall i, m \quad (4.2)$$

where β_0 is the usual intercept term that adjusts for the origin, $\beta_m(t)$ are the coefficient functions and ϵ_i are the error terms which are independently and normally distributed with mean 0 and variance σ_i^2 . Using the expansion in (2.2) to reduce the degrees of freedom in the model further using basis functions, the functional predictors $X_{im}(t)$ are expressed as

$$X_{im}(t) = \sum_{k=1}^{K_m^x} c_{imk} \phi_{mk}(t) = \mathbf{c}_{im}^T \boldsymbol{\phi}_m(t), \quad \forall t \in \mathcal{T}_m \quad (4.3)$$

In certain cases, $\boldsymbol{\phi}_m(t)$ may differ depending on how different the functional predictors are among $m = 1, \dots, M$. Furthermore, the coefficient functions are represented by linear combinations of K_m^β basis functions $\{\psi_{m1}(t), \dots, \psi_{mK_m^\beta}(t)\}$, with the following form

$$\beta_m(t) = \sum_{l=1}^{K_m^\beta} b_{ml} \psi_{ml}(t) = \mathbf{b}_m^T \boldsymbol{\psi}_m(t), \quad \forall t \in \mathcal{T}_m \quad (4.4)$$

Replacing equations (4.3) and (4.4) in equation (4.2) yields

$$\begin{aligned} Y_i &= \beta_0 + \sum_{m=1}^M \int_{\mathcal{T}_m} \mathbf{c}_{im}^T \phi_m(t) \psi'_m(t) \mathbf{b}_m dt + \epsilon_i \\ &= \beta_0 + \sum_{m=1}^M \mathbf{c}_{im}^T \mathbf{J}_{\phi\psi}^{(m)} \mathbf{b}_m + \epsilon_i, \end{aligned} \quad (4.5)$$

where $\mathbf{J}_{\phi\psi}^{(m)} = \int_{\mathcal{T}_m} \phi_m(t) \psi_m^T(t) dt$ is the $K_m^x \times K_m^\beta$ cross-product matrix. Taking equation (4.5) one step further, it can be rewritten as

$$\mathbf{Y} = \mathbf{Z}\mathbf{B} + \boldsymbol{\epsilon} \quad (4.6)$$

where

$$\mathbf{Z} = \begin{pmatrix} \mathbf{z}_1^T \\ \vdots \\ \mathbf{z}_N^T \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{c}_{11}^T \mathbf{J}_{\phi\psi}^{(1)} & \cdots & \mathbf{c}_{1M}^T \mathbf{J}_{\phi\psi}^{(M)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \mathbf{c}_{N1}^T \mathbf{J}_{\phi\psi}^{(1)} & \cdots & \mathbf{c}_{NM}^T \mathbf{J}_{\phi\psi}^{(M)} \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} \beta_0 \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_M \end{pmatrix},$$

\mathbf{Y} is the N -vector of scalar responses, \mathbf{Z} is the $N \times \left(\sum_{m=1}^M K_m^x + 1\right)$ matrix of functional covariates, \mathbf{B} is the $\left(\sum_{m=1}^M K_m^\beta + 1\right) \times 1$ vector of functional coefficients, and $\boldsymbol{\epsilon}$ is the N -vector error terms.

4.1.2 Multivariate Scalar Response and Functional Independent Variables

Expanding from section 4.1.1, consider the $N \times L$ matrix \mathbf{Y} to be a matrix multivariate scalar responses. The regression model that evaluates the relationship between the matrix of scalar responses and the functional covariates is given by

$$Y_{ij} = \beta_{0j} + \sum_{m=1}^M \int_{\mathcal{T}_m} X_{im}(t) \beta_{mj}(t) dt + \epsilon_{ij}, \quad \forall i, j \quad (4.7)$$

where β_{0j} are the intercepts, $\beta_{mj}(t)$ are the coefficient functions and $\boldsymbol{\epsilon}_{ij} = (\epsilon_{i1}, \dots, \epsilon_{iL})'$ are independently and normally distributed with mean vector $\mathbf{0}$ and variance-covariance matrix Σ . As always, the idea is to reduce the degrees of freedom in the model using basis functions. Therefore, functional predictors $X_{im}(t)$ are expressed as

$$X_{im}(t) = \sum_{k=1}^{K_m^x} c_{imk} \phi_{mk}(t) = \mathbf{c}_{im}^T \boldsymbol{\phi}_m(t), \quad \forall t \in \mathcal{T}_m \quad (4.8)$$

The coefficient functions are represented by linear combinations of K_m^β basis functions $\{\psi_{m1}(t), \dots, \psi_{mK_m^\beta}(t)\}$, with the following form

$$\beta_{mj}(t) = \sum_{l=1}^{K_m^\beta} b_{mlj} \psi_{ml}(t) = \mathbf{b}_{mj}^T \boldsymbol{\psi}_m(t), \quad \forall t \in \mathcal{T}_m \quad (4.9)$$

Replacing equations (4.8) and (4.9) in equation (4.7) yields

$$\begin{aligned} Y_{ij} &= \beta_{0j} + \sum_{m=1}^M \int_{\mathcal{T}_m} \mathbf{c}_{im}^T \boldsymbol{\phi}_m(t) \boldsymbol{\psi}_m'(t) \mathbf{b}_{mj} dt + \epsilon_{ij} \\ &= \beta_{0j} + \sum_{m=1}^M \mathbf{c}_{im}^T \mathbf{J}_{\phi\psi}^{(m)} \mathbf{b}_{mj} + \epsilon_{ij}, \end{aligned} \quad (4.10)$$

where $\mathbf{J}_{\phi\psi}^{(m)} = \int_{\mathcal{T}_m} \boldsymbol{\phi}_m(t) \boldsymbol{\psi}_m'(t) dt$ are the $K_m^x \times K_m^\beta$ cross-product matrices. Taking equation (4.10) one step further, it can be rewritten as:

$$\mathbf{y} = \mathbf{Z}\mathbf{B} + \mathbf{E} \quad (4.11)$$

where

$$\mathbf{Z} = \begin{pmatrix} \mathbf{z}'_1 \\ \vdots \\ \mathbf{z}'_N \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{c}_{1m}^T \mathbf{J}_{\phi\psi}^{(1)} & \cdots & \mathbf{c}_{1M}^T \mathbf{J}_{\phi\psi}^{(M)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \mathbf{c}_{Nm}^T \mathbf{J}_{\phi\psi}^{(1)} & \cdots & \mathbf{c}_{NM}^T \mathbf{J}_{\phi\psi}^{(M)} \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}'_{(0)} \\ \vdots \\ \mathbf{b}'_{(L)} \end{pmatrix}' = \begin{pmatrix} \beta_{01} & \cdots & \beta_{0L} \\ \beta_{11} & \cdots & \beta_{1L} \\ \vdots & \ddots & \vdots \\ \beta_{M1} & \cdots & \beta_{ML} \end{pmatrix}.$$

\mathbf{Y} is the $N \times L$ matrix of scalar responses, \mathbf{Z} the $N \times \left(\sum_{m=1}^M K_m^\beta + 1\right)$ matrix of functional covariates, \mathbf{B} the $\left(\sum_{m=1}^M K_m^\beta + 1\right) \times L$ matrix of functional covariates, and \mathbf{E} is the $N \times L$ matrix error terms.

4.2 Functional Response and Functional Independent Variables

In the previous section, the scenario involved scalar responses and functional covariates. In this section, the linear model is a fully *Functional Linear Regression* model in which both the response and covariates are functions. This is given below:

$$Y_i(t) = \beta_0(t) + \sum_{m=1}^M \int_{\mathcal{T}_m} X_{im}(s) \beta_m(s, t) ds + \epsilon_i(t), \quad \forall s \in \mathcal{T}_m \text{ \& } \forall t \in \mathcal{T} \quad (4.12)$$

The function $\beta_0(t)$ is a parameter function acting as the constant term in the standard regression setup, and allows for different functional origins for the functional response. The function $\beta_m(s, t)$ are bivariate coefficient functions which impose varying weights on $X_{im}(s)$ at arbitrary time $t \in \mathcal{T}_m$, and $\epsilon_i(t)$ are the error functions. Using the expansion in (2.2), the functional predictors $X_{im}(t)$ are expressed as

$$X_{im}(s) = \sum_{j=1}^{K_m^x} \tilde{c}_{imj} \phi_{mj}(s) = \tilde{\mathbf{c}}_{im}^T \boldsymbol{\phi}_m(s), \quad \forall s \in \mathcal{T}_m, \quad (4.13)$$

the functional responses $Y_i(t)$ are given by

$$Y_i(t) = \sum_{k=1}^{K_y} \tilde{d}_{ik} \psi_k(t) = \tilde{\mathbf{d}}_i^T \boldsymbol{\psi}(t), \quad \forall t \in \mathcal{T}_m. \quad (4.14)$$

The expression of β as a double expansion seems to be appropriate due to its double effect on both the predictors and response variables. The coefficient functions $\beta_m(s, t)$ are expressed as follows

$$\beta_m(s, t) = \sum_{j,k} b_{mjk} \phi_{mj}(s) \psi_k(t) = \boldsymbol{\phi}_m^T(s) \mathbf{B}_m \boldsymbol{\psi}(t), \quad (4.15)$$

where \mathbf{B}_m is a $K_m^x \times K_y$ coefficient matrices. By centering the *Functional Linear Regression* model (4.12) in the following way

$$\begin{aligned} X_{im}^*(s) &= X_{im}(s) - \bar{X}_{im}(s) \\ &= \tilde{\mathbf{c}}_{im}^T \boldsymbol{\phi}(s) - \bar{\mathbf{c}}_{im}^T \boldsymbol{\phi}(s) \\ &= \mathbf{c}_{im}^T \boldsymbol{\phi}(s), \end{aligned} \quad (4.16)$$

$$\begin{aligned} Y_i^*(t) &= Y_i(t) - \bar{Y}_i(t) \\ &= \tilde{\mathbf{d}}_i^T \boldsymbol{\psi}(t) - \bar{\mathbf{d}}_i^T \boldsymbol{\psi}(t) \\ &= \mathbf{d}_i^T \boldsymbol{\psi}(t), \end{aligned} \quad (4.17)$$

equation (4.12) now become

$$Y_i^*(t) = \sum_{m=1}^M \int_{\mathcal{T}_m} X_{im}^*(s) \beta_m(s, t) ds + \epsilon_i^*(t). \quad (4.18)$$

From equations (4.15), (4.16) and (4.17), equation (4.18) have the following form:

$$\begin{aligned} \mathbf{d}_i^T \boldsymbol{\psi}(t) &= \sum_{m=1}^M \int_{\mathcal{T}_m} \mathbf{c}_{im}^T \boldsymbol{\phi}(s) \boldsymbol{\phi}_m^T(s) \mathbf{B}_m \boldsymbol{\psi}(t) ds + \epsilon_i^*(t) \\ &= \sum_{m=1}^M \mathbf{c}_{im}^T \mathbf{J}_{\phi_m} \mathbf{B}_m \boldsymbol{\psi}(t) + \epsilon_i^*(t) \\ &= \mathbf{z}_i^T \mathbf{B} \boldsymbol{\psi}(t) + \epsilon_i^*(t) \end{aligned} \quad (4.19)$$

where $\mathbf{z}_i = (\mathbf{c}_{i1}^T \mathbf{J}_{\phi_1}, \dots, \mathbf{c}_{iM}^T \mathbf{J}_{\phi_M})^T$ is a vector of length $\left[\sum_{m=1}^M K_m^x \right]$, $\mathbf{J}_{\phi_m} = \int_{\mathcal{T}_m} \boldsymbol{\phi}(s) \boldsymbol{\phi}^T(s) ds$ which is $K_m^x \times K_m^x$ matrix, and $\mathbf{B} = (\mathbf{B}_1, \dots, \mathbf{B}_M)^T$ is a $\left(\sum_{m=1}^M K_m^x \times K_y \right)$ matrix. Combining all the information above, the *Functional Linear Regression* model for all the observations is

$$\mathbf{D} \boldsymbol{\psi}(t) = \mathbf{Z} \mathbf{B} \boldsymbol{\psi}(t) + \boldsymbol{\mathcal{E}}(t) \quad (4.20)$$

where \mathbf{D} is a $N \times K_y$ matrix and \mathbf{Z} is a matrix with dimensions $N \times \left(\sum_{m=1}^M K_m^x \right)$

4.3 Model Estimation

The main focus is now to estimate the parameter matrix \mathbf{B} in the *Functional Linear Regression* model (4.20). The methods considered are the followings

- Least Square method (in the FLRM context);
- *Maximum Likelihood* method;
- *Penalized Maximum Likelihood* method.

4.3.1 Least Square method

Ramsay and Silverman (2005) estimated \mathbf{B} in the model (4.20) by minimizing the integrated residual sum of squares, the result is now

$$\begin{aligned}
 & \sum_{i=1}^N \int_{\mathcal{T}} \left[Y_i^*(t) - \sum_{m=1}^M \int_{\mathcal{T}_m} X_{im}^*(s) \beta_m(s, t) ds \right]^2 dt \\
 &= \int_{\mathcal{T}} \text{tr} \left\{ (D\psi(t) - Z\mathbf{B}\psi(t)) (D\psi(t) - Z\mathbf{B}\psi(t))^T \right\} dt \\
 &= \int_{\mathcal{T}} \text{tr} \left\{ (D - Z\mathbf{B}) \psi(t) \psi^T(t) (D - Z\mathbf{B})^T \right\} dt \\
 &= \text{tr} \left\{ (D - Z\mathbf{B}) \mathbf{J}_{\psi} (D - Z\mathbf{B})^T \right\} \\
 &= \text{tr} \left\{ D\mathbf{J}_{\psi} D^T - D\mathbf{J}_{\psi} \mathbf{B}^T Z^T - Z\mathbf{B} \mathbf{J}_{\psi} D^T + Z\mathbf{B} \mathbf{J}_{\psi} \mathbf{B}^T Z^T \right\} \\
 &= \text{tr} (D\mathbf{J}_{\psi} D^T) - 2\text{tr} (\mathbf{B} \mathbf{J}_{\psi} D^T Z) + \text{tr} (Z^T Z \mathbf{B} \mathbf{J}_{\psi} \mathbf{B}^T) \tag{4.21}
 \end{aligned}$$

where $\mathbf{J}_{\psi} = \int_{\mathcal{T}} \psi(t) \psi^T(t) dt$ is a $K_y \times K_y$ matrix of basis functions. Computing the derivative of (4.21) with respect to \mathbf{B} and set the result to zero gives

$$\begin{aligned}
 & -2 (Z^T D \mathbf{J}_{\psi}) + 2 (Z^T Z \mathbf{B} \mathbf{J}_{\psi}) = \mathbf{0} \\
 & \implies Z^T D \mathbf{J}_{\psi} = Z^T Z \mathbf{B} \mathbf{J}_{\psi} \\
 & \implies \text{vec} (Z^T Z \mathbf{B} \mathbf{J}_{\psi}) = \text{vec} (Z^T D \mathbf{J}_{\psi}) \\
 & \implies (\mathbf{J}_{\psi} \otimes Z^T Z) \text{vec} (\mathbf{B}) = \text{vec} (Z^T D \mathbf{J}_{\psi}) \\
 & \implies \text{vec} (\hat{\mathbf{B}}) = (\mathbf{J}_{\psi} \otimes Z^T Z)^{-1} \text{vec} (Z^T D \mathbf{J}_{\psi}) \tag{4.22}
 \end{aligned}$$

where $\text{vec} (\mathbf{B})$ is a column vector of length $(\sum_{m=1}^M K_m^x) \times K_y$ of \mathbf{B} .

4.3.2 Maximum Likelihood method

Suppose the error function from equation (4.18) $\epsilon_i^*(t)$ are represented by linear combinations of basis functions $\psi_k(t)$, the same as the functional response $Y_i^*(t)$, that is,

$$\epsilon_i^*(t) = \sum_{k=1}^{K_y} e_{ik} \psi_k(t) = \mathbf{e}_i^T \boldsymbol{\psi}(t). \quad (4.23)$$

Therefore, the above result in equation (4.19) gives the following

$$\mathbf{d}_i^T \boldsymbol{\psi}(t) = \mathbf{z}_i^T \mathbf{B} \boldsymbol{\psi}(t) + \mathbf{e}_i^T \boldsymbol{\psi}(t) \quad (4.24)$$

where $\mathbf{e}_i = (e_{i1}, \dots, e_{iK_y})^T$ is a K_y -dimensional vector.

It is assumed that $\mathbf{e}_i \stackrel{i.i.d}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma}$ be the $K_y \times K_y$ variance-covariance matrix. By multiplying both sides of equation (4.24) from the right by $\boldsymbol{\psi}^T(t)$ and integrating the whole equation over the space \mathcal{T} leads to

$$\begin{aligned} \mathbf{d}_i^T \boldsymbol{\psi}(t) \boldsymbol{\psi}^T(t) &= \mathbf{z}_i^T \mathbf{B} \boldsymbol{\psi}(t) \boldsymbol{\psi}^T(t) + \mathbf{e}_i^T \boldsymbol{\psi}(t) \boldsymbol{\psi}^T(t) \\ \implies \int_{\mathcal{T}} \mathbf{d}_i^T \boldsymbol{\psi}(t) \boldsymbol{\psi}^T(t) dt &= \int_{\mathcal{T}} \mathbf{z}_i^T \mathbf{B} \boldsymbol{\psi}(t) \boldsymbol{\psi}^T(t) dt + \int_{\mathcal{T}} \mathbf{e}_i^T \boldsymbol{\psi}(t) \boldsymbol{\psi}^T(t) dt \\ \implies \mathbf{d}_i^T \mathbf{J}_{\psi} &= \mathbf{z}_i^T \mathbf{B} \mathbf{J}_{\psi} + \mathbf{e}_i^T \mathbf{J}_{\psi}. \end{aligned} \quad (4.25)$$

The matrix \mathbf{J}_{ψ} is nonsingular, therefore the simplified result from the above equation is:

$$\mathbf{d}_i^T = \mathbf{z}_i^T \mathbf{B} + \mathbf{e}_i^T, \quad i = 1, 2, \dots, N. \quad (4.26)$$

The above equation can be rewritten as

$$\mathbf{D} = \mathbf{Z} \mathbf{B} + \mathbf{E}, \quad (4.27)$$

which has the same form as a multivariate regression model defined in equation (4.11). It can be noted that equation (4.26) can be rewritten by transposing the whole equation as follows

$$\mathbf{d}_i = \mathbf{B}^T \mathbf{z}_i + \mathbf{e}_i, \quad i = 1, 2, \dots, N. \quad (4.28)$$

The probability density for a functional response Y_i given a functional predictor is

$$f(\mathbf{Y}_i | \boldsymbol{\theta}) = \frac{1}{(2\pi)^{K_y/2} |\boldsymbol{\Sigma}|^{1/2}} \times \exp \left\{ -\frac{1}{2} (\mathbf{d}_i - \mathbf{B}^T \mathbf{z}_i)^T \boldsymbol{\Sigma}^{-1} (\mathbf{d}_i - \mathbf{B}^T \mathbf{z}_i) \right\}, \quad (4.29)$$

where $\boldsymbol{\theta} = \{\mathbf{B}, \boldsymbol{\Sigma}\}$. The log-likelihood function is

$$\begin{aligned}
l(\mathbf{Y}|\boldsymbol{\theta}) &= -\frac{N}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{i=1}^N \left\{ (\mathbf{d}_i - \mathbf{B}^T \mathbf{z}_i)^T \boldsymbol{\Sigma}^{-1} (\mathbf{d}_i - \mathbf{B}^T \mathbf{z}_i) \right\} - \frac{NK_y}{2}\log(2\pi) \\
&\propto -\frac{N}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{i=1}^N \text{tr} \left\{ \boldsymbol{\Sigma}^{-1} (\mathbf{d}_i - \mathbf{B}^T \mathbf{z}_i) (\mathbf{d}_i - \mathbf{B}^T \mathbf{z}_i)^T \right\} \\
&= -\frac{N}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\text{tr} \left\{ \boldsymbol{\Sigma}^{-1} \sum_{i=1}^N (\mathbf{d}_i - \mathbf{B}^T \mathbf{z}_i) (\mathbf{d}_i - \mathbf{B}^T \mathbf{z}_i)^T \right\} \\
&= -\frac{N}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\text{tr} \left\{ \boldsymbol{\Sigma}^{-1} (\mathbf{D} - \mathbf{ZB})^T (\mathbf{D} - \mathbf{ZB}) \right\} \tag{4.30}
\end{aligned}$$

with $l(\mathbf{Y}|\boldsymbol{\theta}) = \sum_{i=1}^N \log f(\mathbf{Y}_i|\boldsymbol{\theta})$. Taking the derivatives of the above equation with respect to $\boldsymbol{\Sigma}^{-1}$ and \mathbf{B} gives:

$$\begin{aligned}
\frac{\partial l(\mathbf{Y}|\boldsymbol{\theta})}{\partial \mathbf{B}} &= \mathbf{Z}^T \mathbf{D} \boldsymbol{\Sigma}^{-1} - \mathbf{Z}^T \mathbf{Z} \mathbf{B} \boldsymbol{\Sigma}^{-1} \\
\frac{\partial l(\mathbf{Y}|\boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}^{-1}} &= \frac{N}{2} \boldsymbol{\Sigma} - \frac{1}{2} (\mathbf{D} - \mathbf{ZB})^T (\mathbf{D} - \mathbf{ZB})
\end{aligned}$$

Therefore, Equating the above results to $\mathbf{0}$

$$\hat{\mathbf{B}} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{D}, \text{ and } \hat{\boldsymbol{\Sigma}} = \frac{1}{N} (\mathbf{D} - \mathbf{Z}\hat{\mathbf{B}})^T (\mathbf{D} - \mathbf{Z}\hat{\mathbf{B}}) \tag{4.31}$$

4.3.3 Penalized Maximum Likelihood method

Dealing with *Functional Linear Regression* implies an infinite number of independent variables to predict a $(N \times \infty)$ -matrix of response variables. The solution is to model the weighting information to be sufficiently smooth, this implies that the penalty term involves the coefficient functions.

Using a similar approach as in equation (2.29), the penalized log-likelihood function is given by

$$l_{\Lambda}(\boldsymbol{\theta}) = l(\mathbf{Y}|\boldsymbol{\theta}) - \frac{N}{2} \text{tr} \left\{ \mathbf{B}^T (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \mathbf{B} \right\} \tag{4.32}$$

where $\boldsymbol{\Lambda}_M$ is a $\left(\sum_{m=1}^M K_m^x\right) \times \left(\sum_{m=1}^M K_m^x\right)$ matrix of regularization parameters $\lambda_1, \dots, \lambda_M$, that is $\boldsymbol{\Lambda}_M = \boldsymbol{\lambda}_M \boldsymbol{\lambda}_M^T$ with $\boldsymbol{\lambda}_M = \left(\sqrt{\lambda_1} \mathbf{1}_{K_1^x}^T, \dots, \sqrt{\lambda_M} \mathbf{1}_{K_M^x}^T\right)^T$.

$\mathbf{\Omega}$ is a $\left(\sum_{m=1}^M K_m^x\right) \times \left(\sum_{m=1}^M K_m^x\right)$ positive semi-definite matrix that has the form:

$$\mathbf{\Omega} = \begin{pmatrix} \mathbf{\Omega}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{\Omega}_M \end{pmatrix},$$

with $\mathbf{\Omega}_m$ ($m = 1, \dots, M$) being $K_m^x \times K_m^x$ positive semi-definite matrices.

Typically, $\mathbf{\Omega}_m = \mathbf{\Delta}_s^T \mathbf{\Delta}_s$ where $\mathbf{\Delta}_s$ is an $(K_m^x - s) \times K_m^x$ matrix that represents the s^{th} difference operator (see section 2.3.2). The function (4.32) can be rewritten as follows:

$$l_{\Lambda}(\boldsymbol{\theta}) \propto -\frac{N}{2} \log|\boldsymbol{\Sigma}| - \frac{1}{2} \text{tr} \left\{ \boldsymbol{\Sigma}^{-1} (\mathbf{D} - \mathbf{Z}\mathbf{B})^T (\mathbf{D} - \mathbf{Z}\mathbf{B}) \right\} - \frac{N}{2} \text{tr} \left\{ \mathbf{B}^T (\mathbf{\Lambda}_M \odot \mathbf{\Omega}) \mathbf{B} \right\} \quad (4.33)$$

Maximizing equation (4.33) with respect to \mathbf{B} and $\boldsymbol{\Sigma}^{-1}$ is done as follows:

Maximizing with respect to \mathbf{B}

$$\begin{aligned} l_{\Lambda}(\boldsymbol{\theta}) &\propto -\frac{1}{2} \text{tr} \left\{ \boldsymbol{\Sigma}^{-1} (\mathbf{D} - \mathbf{Z}\mathbf{B})^T (\mathbf{D} - \mathbf{Z}\mathbf{B}) \right\} - \frac{N}{2} \text{tr} \left\{ \mathbf{B}^T (\mathbf{\Lambda}_M \odot \mathbf{\Omega}) \mathbf{B} \right\} - \frac{N}{2} \log|\boldsymbol{\Sigma}| \\ &= -\frac{1}{2} \text{tr} \left\{ \boldsymbol{\Sigma}^{-1} (\mathbf{D}^T \mathbf{D} - \mathbf{D}^T \mathbf{Z}\mathbf{B} - \mathbf{B}^T \mathbf{Z}^T \mathbf{D} + \mathbf{B}^T \mathbf{Z}^T \mathbf{Z}\mathbf{B}) \right\} - \frac{N}{2} \text{tr} \left\{ \mathbf{B}^T (\mathbf{\Lambda}_M \odot \mathbf{\Omega}) \mathbf{B} \right\} \\ &\quad - \frac{N}{2} \log|\boldsymbol{\Sigma}| \\ &= -\frac{1}{2} \text{tr} (\mathbf{D}\boldsymbol{\Sigma}^{-1} \mathbf{D}^T) + \text{tr} (\mathbf{B}\boldsymbol{\Sigma}^{-1} \mathbf{D}^T \mathbf{Z}) - \frac{1}{2} \text{tr} (\mathbf{Z}^T \mathbf{Z}\mathbf{B}\boldsymbol{\Sigma}^{-1} \mathbf{B}^T) - \frac{N}{2} \text{tr} \left\{ \mathbf{B}^T (\mathbf{\Lambda}_M \odot \mathbf{\Omega}) \mathbf{B} \right\} \\ &\quad - \frac{N}{2} \log|\boldsymbol{\Sigma}| \end{aligned} \quad (4.34)$$

The first derivative of equation (4.34) with respect to \mathbf{B} is given by

$$\frac{\partial l_{\Lambda}(\boldsymbol{\theta})}{\partial \mathbf{B}} = (\mathbf{Z}^T \mathbf{D}\boldsymbol{\Sigma}^{-1}) - (\mathbf{Z}^T \mathbf{Z}\mathbf{B}\boldsymbol{\Sigma}^{-1}) - N (\mathbf{\Lambda}_M \odot \mathbf{\Omega}) \mathbf{B}$$

Equating the above equation to $\mathbf{0}$ implies the followings:

$$\begin{aligned}
& (\mathbf{Z}^T \mathbf{D} \boldsymbol{\Sigma}^{-1}) - (\mathbf{Z}^T \mathbf{Z} \boldsymbol{\beta} \boldsymbol{\Sigma}^{-1}) - N (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \boldsymbol{\beta} = \mathbf{0} \\
\implies & \mathbf{Z}^T \mathbf{D} \boldsymbol{\Sigma}^{-1} = \mathbf{Z}^T \mathbf{Z} \boldsymbol{\beta} \boldsymbol{\Sigma}^{-1} + N (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \boldsymbol{\beta} \\
\implies & \text{vec} (\mathbf{Z}^T \mathbf{D} \boldsymbol{\Sigma}^{-1}) = \text{vec} [\mathbf{Z}^T \mathbf{Z} \boldsymbol{\beta} \boldsymbol{\Sigma}^{-1} + N (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \boldsymbol{\beta} \mathbf{I}_{K_y}] \\
\implies & (\boldsymbol{\Sigma}^{-1} \otimes \mathbf{Z}^T) \text{vec} (\mathbf{D}) = [\boldsymbol{\Sigma}^{-1} \otimes \mathbf{Z}^T \mathbf{Z} + N \mathbf{I}_{K_y} \otimes (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega})] \text{vec} (\boldsymbol{\beta}) \\
\implies & \text{vec} (\hat{\boldsymbol{\beta}}) = \left[\hat{\boldsymbol{\Sigma}}^{-1} \otimes \mathbf{Z}^T \mathbf{Z} + N \mathbf{I}_{K_y} \otimes (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \right]^{-1} \left(\hat{\boldsymbol{\Sigma}}^{-1} \otimes \mathbf{Z}^T \right) \text{vec} (\mathbf{D}) \blacksquare
\end{aligned} \tag{4.35}$$

Maximizing with respect to $\boldsymbol{\Sigma}^{-1}$

The first derivative of equation (4.33) with respect to $\boldsymbol{\Sigma}^{-1}$ is given by

$$\frac{\partial l_{\Lambda}(\boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}^{-1}} = \frac{N}{2} \boldsymbol{\Sigma} - \frac{1}{2} (\mathbf{D} - \mathbf{Z} \boldsymbol{\beta})^T (\mathbf{D} - \mathbf{Z} \boldsymbol{\beta})$$

Equating the above equation to $\mathbf{0}$ gives the following:

$$\begin{aligned}
& \frac{N}{2} \boldsymbol{\Sigma} - \frac{1}{2} (\mathbf{D} - \mathbf{Z} \boldsymbol{\beta})^T (\mathbf{D} - \mathbf{Z} \boldsymbol{\beta}) = \mathbf{0} \\
\implies & \hat{\boldsymbol{\Sigma}} = \frac{1}{N} (\mathbf{D} - \mathbf{Z} \hat{\boldsymbol{\beta}})^T (\mathbf{D} - \mathbf{Z} \hat{\boldsymbol{\beta}})
\end{aligned} \tag{4.36}$$

The maximum penalized likelihood estimator of \mathbf{D} is therefore given by:

$$\begin{aligned}
\text{vec}(\hat{\mathbf{D}}) &= \text{vec} (\mathbf{Z} \hat{\boldsymbol{\beta}}) \\
&= \text{vec} (\mathbf{Z} \hat{\boldsymbol{\beta}} \mathbf{I}_{K_y}) \\
&= (\mathbf{I}_{K_y} \otimes \mathbf{Z}) \text{vec}(\hat{\boldsymbol{\beta}}) \\
&= (\mathbf{I}_{K_y} \otimes \mathbf{Z}) \left[\hat{\boldsymbol{\Sigma}}^{-1} \otimes \mathbf{Z}^T \mathbf{Z} + N \mathbf{I}_{K_y} \otimes (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \right]^{-1} \left(\hat{\boldsymbol{\Sigma}}^{-1} \otimes \mathbf{Z}^T \right) \text{vec} (\mathbf{D}) \\
&= \mathbf{S}_{\Lambda} \text{vec} (\mathbf{D})
\end{aligned} \tag{4.37}$$

where $\mathbf{S}_{\Lambda} = (\mathbf{I}_{K_y} \otimes \mathbf{Z}) \left[\hat{\boldsymbol{\Sigma}}^{-1} \otimes \mathbf{Z}^T \mathbf{Z} + N \mathbf{I}_{K_y} \otimes (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \right]^{-1} \left(\hat{\boldsymbol{\Sigma}}^{-1} \otimes \mathbf{Z}^T \right)$ is a hat matrix for $\text{vec} (\mathbf{D})$. Substituting the maximum likelihood estimator $\hat{\boldsymbol{\theta}} = \{\hat{\boldsymbol{\Sigma}}, \hat{\boldsymbol{\beta}}\}$ into (4.29), the result is

$$f(\mathbf{Y}_i | \boldsymbol{\theta}) = \frac{1}{(2\pi)^{K_y/2} |\hat{\boldsymbol{\Sigma}}|^{1/2}} \times \exp \left\{ -\frac{1}{2} \left(\mathbf{d}_i - \hat{\boldsymbol{\beta}}^T \mathbf{z}_i \right)^T \hat{\boldsymbol{\Sigma}}^{-1} \left(\mathbf{d}_i - \hat{\boldsymbol{\beta}}^T \mathbf{z}_i \right) \right\}. \tag{4.38}$$

Now that the penalized maximum likelihood estimator of $\hat{\mathbf{D}}$ is derived, the predicting values for the functional response $\hat{\mathbf{Y}}(t)$ are therefore:

$$\hat{\mathbf{Y}}(t) = \hat{\mathbf{D}}\boldsymbol{\psi}(t), \quad \forall t \in \mathcal{T} \quad (4.39)$$

4.4 Model Selection Criteria

When applying the regularization method to select the statistical model (i.e. equation 4.38), it makes sense to look for the selected set of model parameters that leads to the model that minimizes the value of these criteria. The following model criteria are derived from the ones discussed in Chapter 2 with the particularity of being improved to evaluate *Functional Linear Regression* models.

4.4.1 Generalized Cross-Validation

Using similar ideas as in equation (2.32), the *Generalized Cross-Validation* for *Functional Linear Regression* model 4.38 is defined as:

$$\text{GCV} = \frac{\text{tr} \left\{ \left(\mathbf{D} - \mathbf{Z}\hat{\mathbf{B}} \right)^T \left(\mathbf{D} - \mathbf{Z}\hat{\mathbf{B}} \right) \right\}}{NK_y (1 - \text{tr}(\mathbf{S}_{\Lambda}) / (NK_y))^2}, \quad (4.40)$$

where \mathbf{S}_{Λ} is the *hat* matrix given in equation (4.37).

4.4.2 Modified AIC

Using the result given in section 2.4.3, the mAIC for evaluating (4.38) is

$$\text{mAIC} = -2 \sum_{i=1}^N \log f(\mathbf{Y}_i | \hat{\boldsymbol{\theta}}) + 2\text{tr}(\mathbf{S}_{\Lambda}) \quad (4.41)$$

4.4.3 Generalized Information Criteria

Using the result that was derived in section 2.4.2, the GIC for model selection in the context of *Functional Linear Regression* modelling is given by

$$\text{GIC} = -2 \sum_{i=1}^N \log f(\mathbf{Y}_i | \hat{\boldsymbol{\theta}}) + 2\text{tr} \left\{ \mathbf{R}_{\Lambda}(\hat{\boldsymbol{\theta}})^{-1} \mathbf{Q}_{\Lambda}(\hat{\boldsymbol{\theta}}) \right\}, \quad (4.42)$$

where $\mathbf{R}_\Lambda(\hat{\boldsymbol{\theta}})$ and $\mathbf{Q}_\Lambda(\hat{\boldsymbol{\theta}})$ are given by

$$\mathbf{R}_\Lambda(\hat{\boldsymbol{\theta}}) = -\frac{1}{N} \sum_{i=1}^N \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \left\{ \log f(\mathbf{Y}_i | \hat{\boldsymbol{\theta}}) - \frac{1}{2} \text{tr} \{ \mathbf{B}^T (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \mathbf{B} \} \right\}$$

and

$$\mathbf{Q}_\Lambda(\hat{\boldsymbol{\theta}}) = \frac{1}{N} \sum_{i=1}^N \frac{\partial}{\partial \boldsymbol{\theta}} \left\{ \log f(\mathbf{Y}_i | \hat{\boldsymbol{\theta}}) - \frac{1}{2} \text{tr} \{ \mathbf{B}^T (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \mathbf{B} \} \right\} \frac{\partial}{\partial \boldsymbol{\theta}^T} \log f(\mathbf{Y}_i | \hat{\boldsymbol{\theta}}).$$

Note the elements in the four quadrants of $\mathbf{R}_\Lambda(\hat{\boldsymbol{\theta}})$ are:

$$\begin{aligned} R_{\Lambda}^{11}(\hat{\boldsymbol{\theta}}) &= \mathbf{Z}^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{Z} - N (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}); \\ R_{\Lambda}^{12}(\hat{\boldsymbol{\theta}}) &= N \mathbf{D}^T \mathbf{Z} + N \hat{\mathbf{B}}^T \mathbf{Z}^T \mathbf{Z}; \\ R_{\Lambda}^{21}(\hat{\boldsymbol{\theta}}) &= \mathbf{Z}^T \mathbf{D} + \mathbf{Z}^T \mathbf{Z} \hat{\mathbf{B}}; \\ R_{\Lambda}^{22}(\hat{\boldsymbol{\theta}}) &= \frac{N}{2} \mathbb{I}_{K_y}. \end{aligned}$$

Similarly, the elements in the four quadrants of $\mathbf{Q}_\Lambda(\hat{\boldsymbol{\theta}})$ are:

$$\begin{aligned} Q_{\Lambda}^{11}(\hat{\boldsymbol{\theta}}) &= \left[\mathbf{Z}^T \mathbf{D} \hat{\boldsymbol{\Sigma}}^{-1} - \mathbf{Z}^T \mathbf{Z} \hat{\mathbf{B}} \hat{\boldsymbol{\Sigma}}^{-1} - N (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \hat{\mathbf{B}} \right] \left[\mathbf{Z}^T \mathbf{D} \hat{\boldsymbol{\Sigma}}^{-1} - \mathbf{Z}^T \mathbf{Z} \hat{\mathbf{B}} \hat{\boldsymbol{\Sigma}}^{-1} \right]^T; \\ Q_{\Lambda}^{12}(\hat{\boldsymbol{\theta}}) &= \left[\mathbf{Z}^T \mathbf{D} \hat{\boldsymbol{\Sigma}}^{-1} - \mathbf{Z}^T \mathbf{Z} \hat{\mathbf{B}} \hat{\boldsymbol{\Sigma}}^{-1} - N (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \hat{\mathbf{B}} \right] \left[\frac{N}{2} \hat{\boldsymbol{\Sigma}} - \frac{1}{2} (\mathbf{D} - \mathbf{Z} \hat{\mathbf{B}})^T (\mathbf{D} - \mathbf{Z} \hat{\mathbf{B}}) \right]^T; \\ Q_{\Lambda}^{21}(\hat{\boldsymbol{\theta}}) &= \left[\frac{N}{2} \hat{\boldsymbol{\Sigma}} - \frac{1}{2} (\mathbf{D} - \mathbf{Z} \hat{\mathbf{B}})^T (\mathbf{D} - \mathbf{Z} \hat{\mathbf{B}}) \right] \left[\mathbf{Z}^T \mathbf{D} \hat{\boldsymbol{\Sigma}}^{-1} - \mathbf{Z}^T \mathbf{Z} \hat{\mathbf{B}} \hat{\boldsymbol{\Sigma}}^{-1} \right]^T; \\ Q_{\Lambda}^{22}(\hat{\boldsymbol{\theta}}) &= \left[\frac{N}{2} \hat{\boldsymbol{\Sigma}} - \frac{1}{2} (\mathbf{D} - \mathbf{Z} \hat{\mathbf{B}})^T (\mathbf{D} - \mathbf{Z} \hat{\mathbf{B}}) \right] \left[\frac{N}{2} \hat{\boldsymbol{\Sigma}} - \frac{1}{2} (\mathbf{D} - \mathbf{Z} \hat{\mathbf{B}})^T (\mathbf{D} - \mathbf{Z} \hat{\mathbf{B}}) \right]^T \end{aligned}$$

A more thorough derivation of the above results can be found in the Appendix B.

4.4.4 Generalized Bayesian Information Criterion

Based on the result from section 2.4.4, the GBIC for evaluating the model 4.38 fitted by the penalized maximum likelihood method is given by

$$\begin{aligned} \text{GBIC} &= -2 \sum_{i=1}^N \log f(\mathbf{Y}_i | \hat{\boldsymbol{\theta}}) + N \text{tr} \left\{ \hat{\mathbf{B}}^T (\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}) \hat{\mathbf{B}} \right\} \\ &\quad + (r + K_y q) \log N - (r + K_y q) \log(2\pi) \\ &\quad - K_y \log |\boldsymbol{\Lambda}_M \odot \boldsymbol{\Omega}|_+ + \log |\mathbf{R}_\Lambda(\hat{\boldsymbol{\theta}})| \end{aligned} \tag{4.43}$$

where $q = p - \text{rank}(\mathbf{\Omega})$, $p = \sum_m K_m^x$, $r = \frac{K_y(K_y + 1)}{2}$ and $\mathbf{R}_{\mathbf{\Lambda}}(\hat{\boldsymbol{\theta}})$ is as defined in section 4.4.3. For a detailed derivation of the above equation consult Matsui et al. (2009).

4.5 Closing Comments

This chapter reviewed some of the key concepts linked to Functional Linear Regression Model. Three different forms of Functional Linear models were discussed; on one hand *Functional Linear Regression* models when the response is multivariate and on the other hand *Functional Linear Regression* models when the response is functional. A particular emphasis was placed on the latter. The Functional Linear Regression models were estimated using *Least Square* method (in the FLRM context); *Maximum Likelihood* method and *Penalized Maximum Likelihood* method. A crucial problem in constructing Functional Linear Regression models using *Penalized Maximum Likelihood* method was the selection of the smoothing parameters. For that purpose, improved model criteria had to be derived from the ones mentioned in Chapter 2: *Generalized Cross-Validation* with $\mathbf{S}_{\mathbf{\Lambda}}$ as the *hat* matrix; *Generalized Information*; *modified Akaike Information Criteria* and *Generalized Bayesian Information Criteria*.

In the next chapter, an application of the *Penalized Maximum Likelihood* will be performed on the **Aemet** dataset from Febrero-Bande and Oviedo de la Fuente (2012).

Chapter 5

Applications: Functional Linear Regression Modeling

5.1 Introduction

Chapter 4 discussed the different types of functional linear regression models as well as the different criteria that should be applied in order to estimate these models. In this chapter, the interest will be on *Functional Linear Regression* models where both the response variable and the independent variables are functional as defined in section 4.2. The dataset that will be used to illustrate the modelling of functional variables is the **Aemet** data used by Febrero-Bande and Oviedo de la Fuente (2012) in their R-package **fda.usc**. This dataset has the following features:

- 73 weather stations selected over the time period 1980-2009;
- 365 points of averaged *temperature* from 1980 to 2009 evaluated at each station;
- 365 points of averaged *wind speed* from 1980 to 2009 evaluated at each station;
- 365 points of averaged *log-precipitation* from 1980 to 2009 evaluated at each station.

The exercise of this chapter will be to predict the functional behaviour of the Log-Precipitation knowing the functional behaviours of Temperature, and Wind Speed for each weather station. In this case the functional linear regression equation can be written as follows:

$$Y_i^*(t) = \int_{\mathcal{T}} X_{i1}^*(s)\beta_1(s,t)ds + \int_{\mathcal{T}} X_{i2}^*(s)\beta_2(s,t)ds + \epsilon_i(t), \quad \forall s, t \in \mathcal{T} \quad (5.1)$$

where $Y_i^*(t)$ is the centered functional variable for Log-Precipitation for the i^{th} station, $X_{i1}^*(s)$ and $X_{i2}^*(s)$ are the centered functional variables for Temperature and Wind Speed respectively for i^{th} station, and $\mathcal{T} = \{0.5, 1.5, \dots, 364.5\}$. Since *Least Squares* and *Maximum Likelihood* methods often result in unstable estimators, the *regularization* method is used to estimate the functional linear model (Matsui et al., 2009).

5.2 Methodology

The computation of functional variables require a clear outline of the steps involved in modeling the functional behaviour of the Temperature and Wind Speed, and consequently the values of the Log-Precipitation at any given time point. Computing the *Functional Linear Regression* model (5.1) are done as follows:

- Step 1 Center the data by subtracting the mean across all stations;
- Step 2 Compute the GCV, GIC, GBIC and mAIC matrices for each station to find optimal values for K and λ ;
- Step 3 Add all the values obtained from **Step 2** for each criterion to compute \hat{K} and $\hat{\lambda}$ that work for all stations;
- Step 4 Compute the \mathbf{C} matrix of coefficients for each independent variable and the \mathbf{D} matrix for the response variable using equation (2.30);
- Step 5 Compute the matrices \mathbf{J}_ϕ leading to compute \mathbf{Z} as explained at the end of section 4.2;
- Step 6 Compute the GCV, GIC, GBIC and mAIC matrices for each station (response variable) to find optimal value for λ for a fixed value of K computed in **Step 2** as explained in section 4.4;
- Step 7 Add all the values obtained from **Step 6** for each criterion to compute the matrices $\hat{\Sigma}$ and $\hat{\mathbf{B}}$ using equations (4.35) and (4.36);
- Step 8 Compute $\hat{\mathbf{D}}$ using equation (4.37).

The above steps are executed for each of the three basis functions, namely: *Gaussian*, *Fourier* and *B-splines*. Note that when the basis functions are orthonormal (e.g. *Fourier*, *B-Splines*), $\mathbf{J}_\phi = \mathbb{I}_{K_m^x}$. Another relevant point to note is that since the functions are observed at equally spaced timestamps, the *Gaussian* basis functions with *B-Splines* method is used to capture the functional behaviour of variables under *Gaussian basis functions*.

Choosing the set of values over which the optimal number of basis functions K is found is critical to the analysis. For large values of K (i.e. 365 daily observations over the year), the bias in estimating the smooth functions is small. But of course, the estimated functions are not smooth and therefore increase their variability. Reducing the variance implies looking for smaller values of K , but at the same time not too small to make the bias unacceptable. Ramsay and Silverman (2005) used $\hat{K} = 65$ to smooth the **Canadian Weather** data in order to economize on computer time when dealing with daily observations. In other words, the ideal smooth curve modelling the weather data is made of 65 basis functions which combines on average one basis function for 5 consecutive days of the year. For the case of the **Aemet** data, finding the optimal K -value depends on the model criterion chosen. A search of the optimal number of basis functions is computed for values of K ranging from 5 to 65. In any case, the influence of smoothing parameter helps in adjusting the smoothness of the curve computed.

Extensive R-codes had to be written to compute all the basis functions and all model criteria to fit all 73 weather stations. Because of the amount of information that had to be computed a large number of times, the use of parallel computing was necessary. Appendix A presents the functions that were written in order to make the long R scripts readable. Some of these R functions were written with the help of Dr. Shuichi KAWANO, they are listed below:

- `Gaussian.bsplines;`
- `Gaussian.kmeans;`
- `Pen_Max_Likelihood;`
- `gic_fun;`
- `mAIC_fun.`

The R scripts can be found in the supplementary materials for this dissertation.

Table 5.1 shows the values the optimal values for \hat{K} and $\hat{\lambda}$ that were found by computing each model criterion on the all 73 stations. From the Table, it is clear that the *Generalized Information Criterion* yields the lowest number of basis functions overall, and the *modified Akaike Information Criterion* yields the lowest value of the smoothing parameter 10^{-5} . Also, *Fourier Basis* tends to result in small values of K , especially for Wind Speed and Log-Precipitation.

TABLE 5.1: \hat{K} values and $\log_{10}(\hat{\lambda})$ values computed for all weather stations

Functional Variables	Types of Basis Functions	Model Criteria	\hat{K}	$\log_{10}\hat{\lambda}$
Temperature	Gaussian	GCV	63	-3.1
		GIC	5	-1.72
		mAIC	63	-3.1
		GBIC	63	-1.03
	Fourier	GCV	63	-5.17
		GIC	5	-1.72
		mAIC	65	-5.17
		GBIC	5	-5.86
	B-Splines	GCV	63	-1.03
		GIC	5	-1.72
		mAIC	63	-3.1
		GBIC	63	-1.03
Wind Speed	Gaussian	GCV	63	-2.41
		GIC	5	-1.72
		mAIC	63	-1.72
		GBIC	63	1.03
	Fourier	GCV	34	-4.48
		GIC	5	-1.03
		mAIC	34	-4.48
		GBIC	5	-5.17
	B-Splines	GCV	63	-2.41
		GIC	5	-1.72
		mAIC	63	-2.41
		GBIC	63	0.345
Log-Precipitation	Gaussian	GCV	51	-0.345
		GIC	5	-2.41
		mAIC	48	-0.345
		GBIC	63	0.345
	Fourier	GCV	7	-5.17
		GIC	7	-5.17
		mAIC	7	-5.17
		GBIC	5	-5.17
	B-Splines	GCV	63	-0.345
		GIC	5	-3.1
		mAIC	65	-0.345
		GBIC	63	0.345

5.3 Gaussian Basis Functions

The focus of this section is to model functional variables (independent variables and response variable) specified in equation (5.1) using *Gaussian Basis* functions with *Penalized Maximum Likelihood* estimate to compute the optimal model parameters $\hat{\Sigma}$ and $\hat{\mathcal{B}}$.

5.3.1 Temperature

Figure 5.1 depicts the fitted curves implemented on the observed Temperatures at A CORUÑA using the information from Table 5.1. For this specific scenario, the *Generalized Information Criterion* results in $\hat{K} = 5$ and $\hat{\lambda} = 10^{-1.72}$. The top right corner of Figure 5.1 shows a very smooth curve (blue line) on top the raw data, which is a different story for the other model criteria.

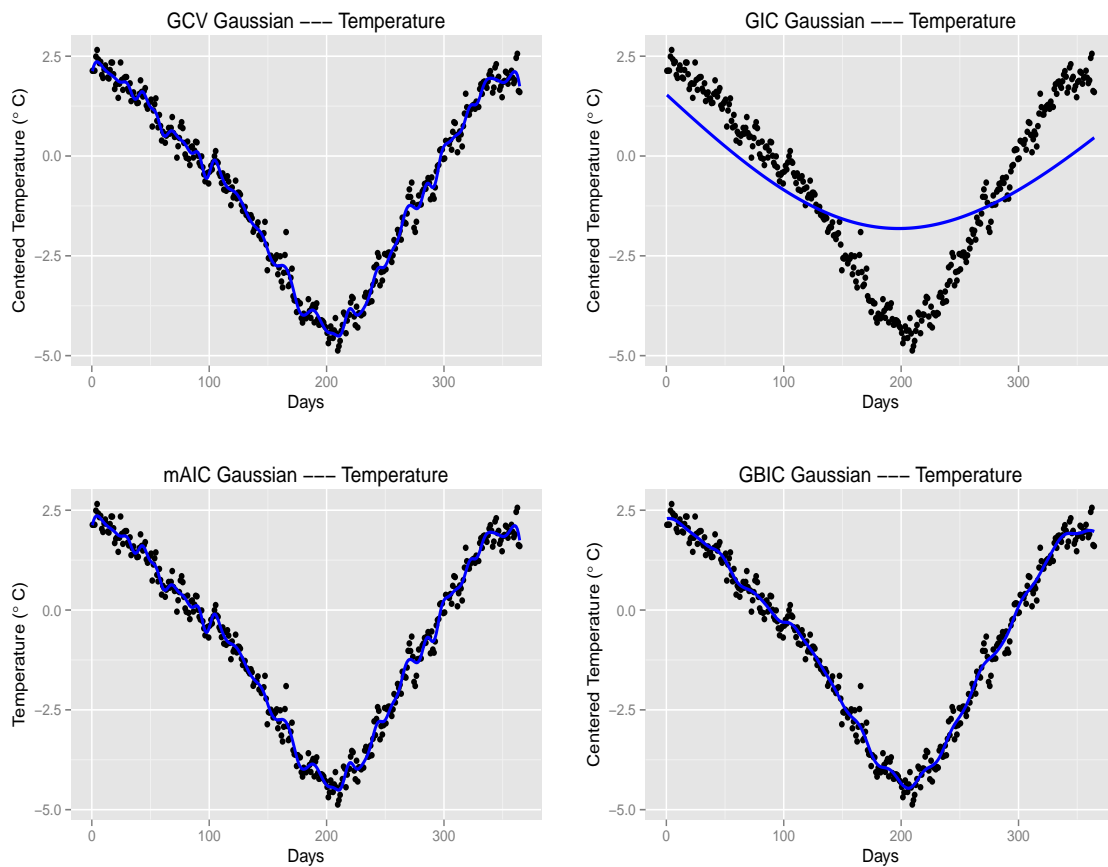


FIGURE 5.1: (a) fitted curve using GCV; (b) fitted curve using GIC; (c) fitted curve using mAIC; (d) fitted curve using GBIC on Temperatures in A CORUÑA

5.3.2 Wind Speed

Figure 5.2 depicts the fitted curves produced on the observed Wind Speed at A CORUÑA using the information from Table 5.1. For this specific scenario, the *Generalized Information Criterion* results in $\hat{K} = 5$ and $\hat{\lambda} = 10^{-1.72}$, same as Temperature. The top right corner of Figure 5.2 shows a very smooth curve (blue line) on top the raw data, which is a different story for the other model criteria.

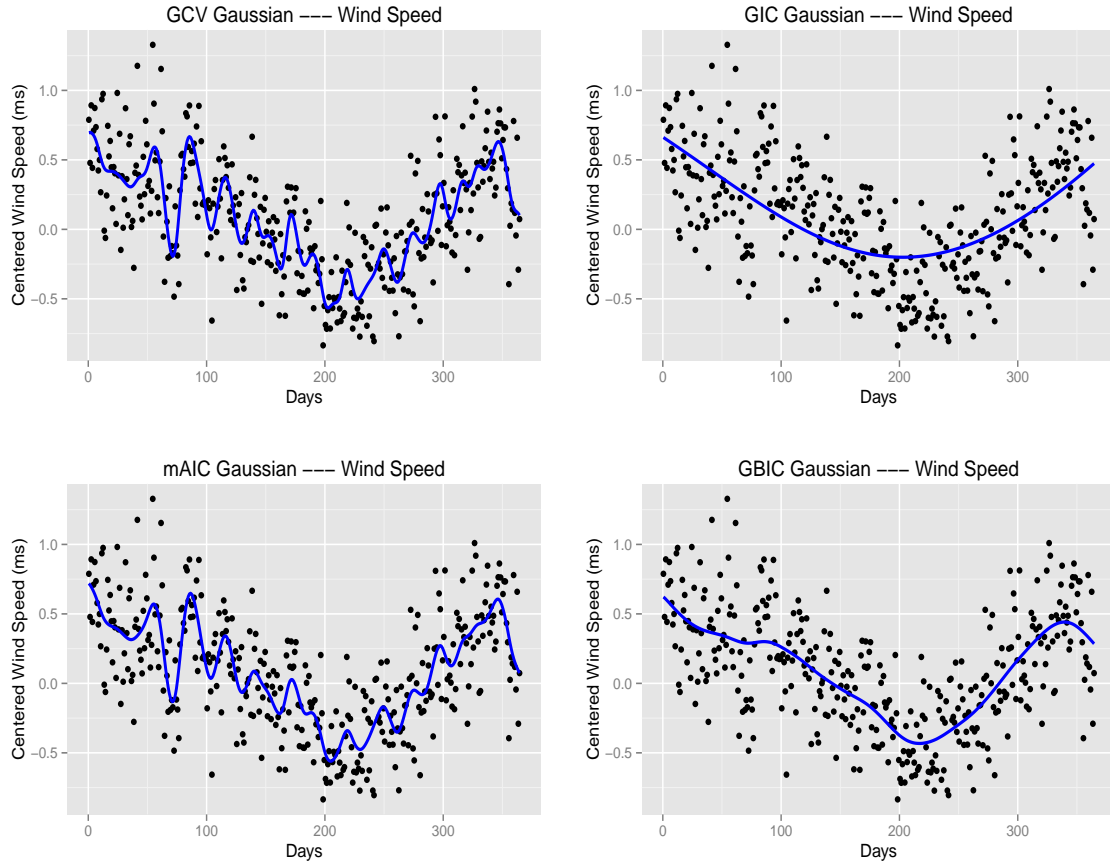


FIGURE 5.2: (a) fitted curve using GCV; (b) fitted curve using GIC; (c) fitted curve using mAIC; (d) fitted curve using GBIC on Wind Speed in A CORUÑA

5.3.3 Log-Precipitation

Computing the Log-Precipitation is done in two stages: (1) compute \mathbf{D} then (2) compute $\hat{\mathbf{D}}$. Recall from Chapter 4 that \mathbf{D} is a $N \times K_y$ matrix of coefficients with $N = 73$ and K_y is the optimal number of basis functions computed following **Steps 1-4** from section 5.2. Their corresponding smoothing parameters $\hat{\lambda}$ are computed as well. The results are listed in Table 5.1 of the Log-Precipitation using *Gaussian Basis* functions. With all that information, it is possible to compute the matrices $\hat{\Sigma}$ and $\hat{\mathcal{B}}$ which yields to $\hat{\mathbf{D}}$. Because of hardware limitations, it is assumed that $\hat{\mathbf{D}}$ and \mathbf{D} are of the same size and the optimization is only done on the smoothing parameters in order to get the optimal $\mathbf{\Lambda}$. The use of equation (4.37) helps in calculating $\hat{\mathbf{D}}$. Table 5.2 shows the optimal values $\log_{10}(\hat{\lambda}_1)$ and $\log_{10}(\hat{\lambda}_2)$ evaluated for all stations using each model criterion for a fixed the number of basis functions.

TABLE 5.2: Summary of the model selection on the Log-Precipitation using *Gaussian basis functions*

	GCV	GIC	mAIC	GBIC
$\log_{10}(\hat{\lambda}_1)$	-3.41	-1.42	-3.72	-2.33
$\log_{10}(\hat{\lambda}_2)$	-2.1	-1.92	-2.1	0.33
\hat{K}	51	5	48	63

Figure 5.3 depicts the fitted curves produced on the observed Log-Precipitation at A CORUÑA using the information from Table 5.2. The blue line represents the smooth curve computed using \mathbf{D} and the red line represents the smooth curve derived from $\hat{\mathbf{D}}$. It can be noted that the red lines have a similar shape as the blue lines for each model criterion. Although the predicted curve for $\hat{\mathbf{D}}_{GIC}$ exhibits a similar trend as the ones for \mathbf{D}_{GIC} , the fit is too smooth to be considered for further analysis. The top right corner plot confirms that statement.

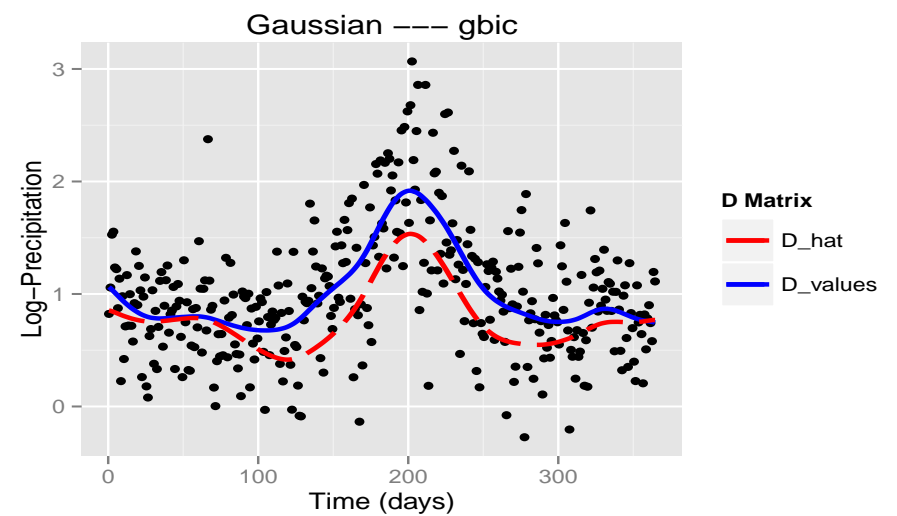
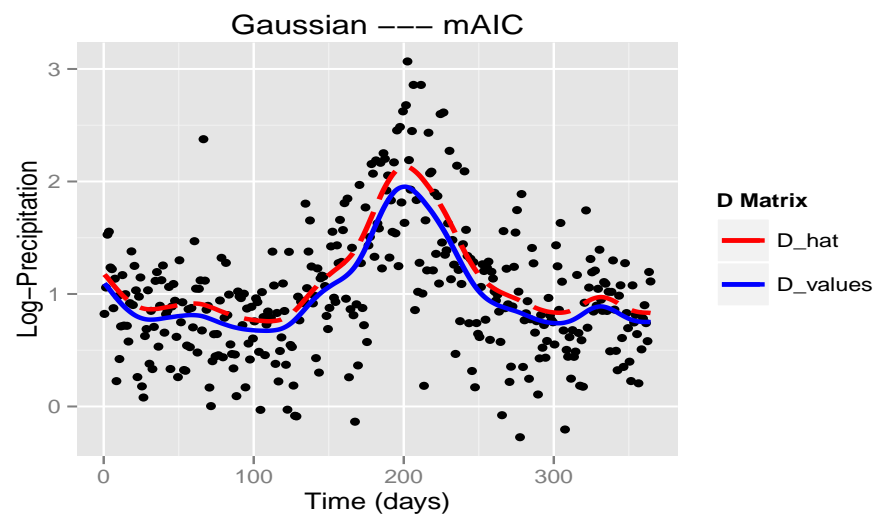
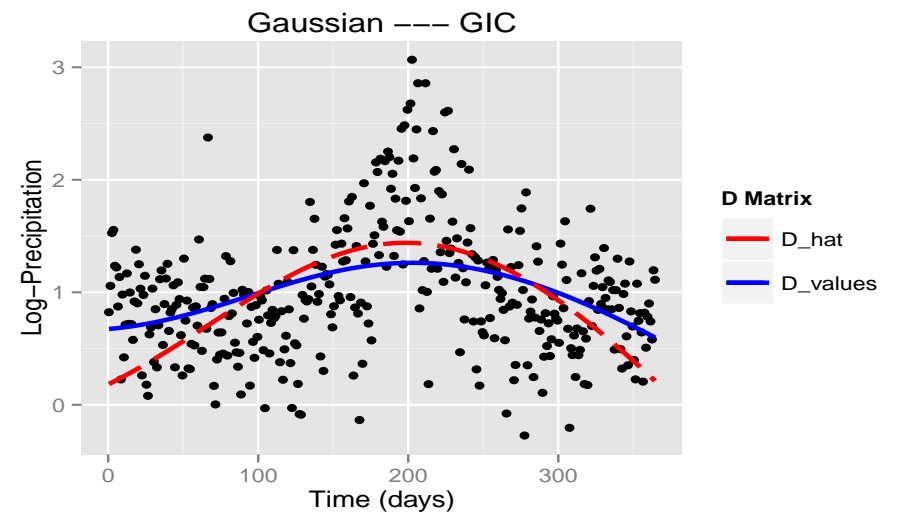
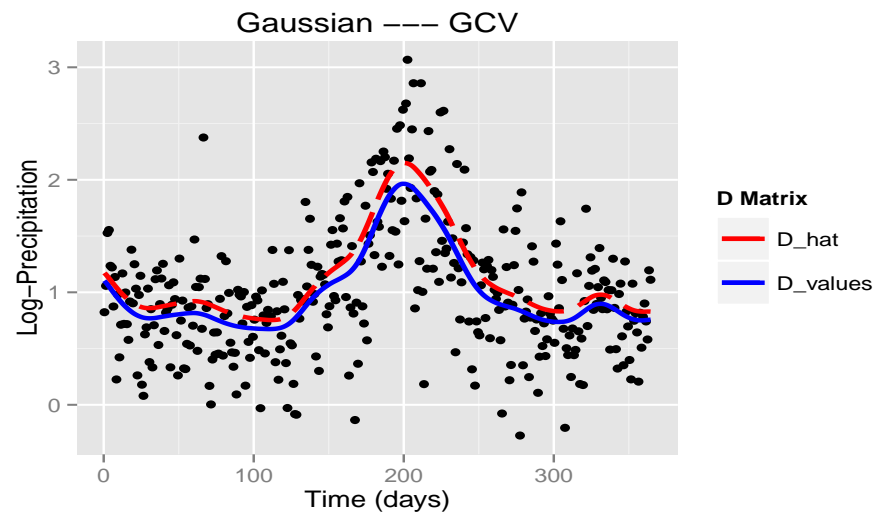


FIGURE 5.3: (a) fitted curve using GCV; (b) fitted curve using GIC; (c) fitted curve using mAIC; (d) fitted curve using GBIC on Log-Precipitation in A CORUÑA

5.4 Fourier Basis Functions

The focus of this section is to model functional variables (independent variables and response variable) specified in equation (5.1) using *Fourier Basis* functions with *Penalized Maximum Likelihood* estimate to compute the optimal model parameters $\hat{\Sigma}$ and $\hat{\mathcal{B}}$.

5.4.1 Temperature

Figure 5.4 depicts the fitted curves produced on the observed Temperature at A CORUÑA using the information from Table 5.1. For this case, once again the *Generalized Information Criterion* has elected small values for K and λ , smaller than the other model criteria. Clearly the small number of basis functions, has an important impact on the shape of the curve.

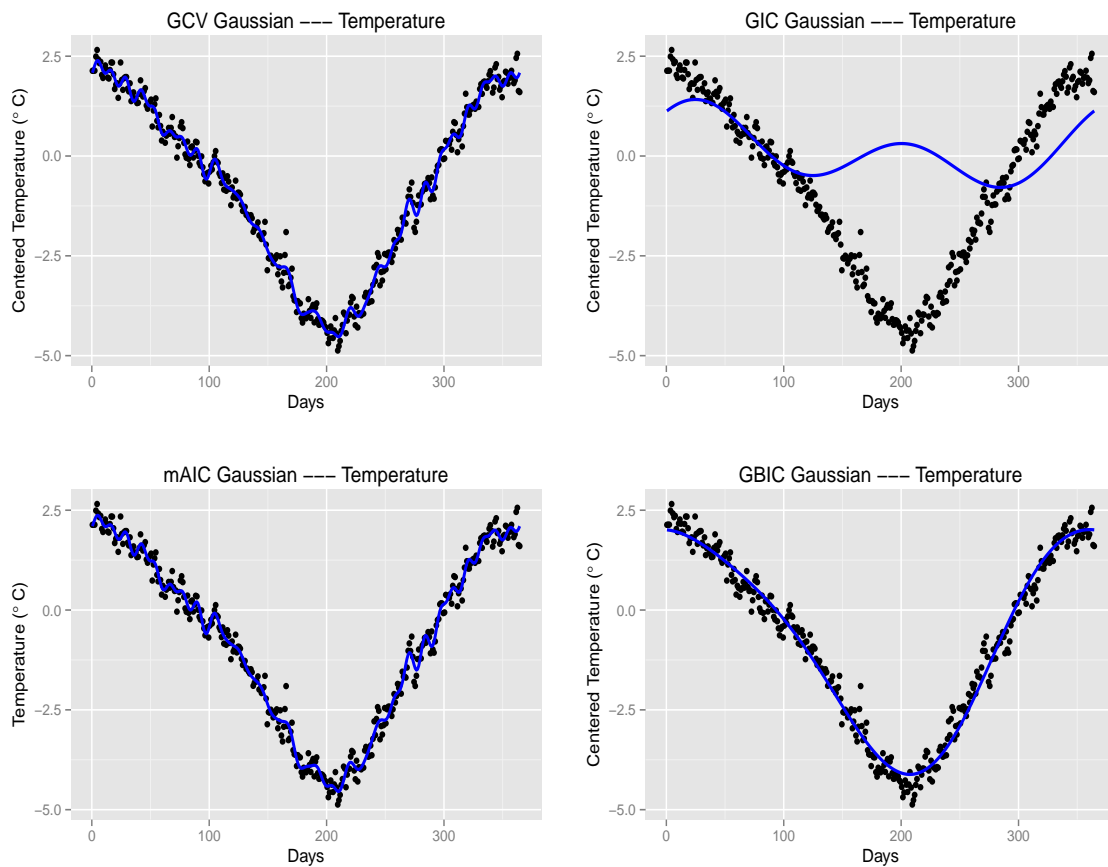


FIGURE 5.4: (a) fitted curve using GCV; (b) fitted curve using GIC; (c) fitted curve using mAIC; (d) fitted curve using GBIC on Temperatures in A CORUÑA

5.4.2 Wind Speed

Figure 5.5 depicts the fitted curves produced on the observed Wind Speed at A CORUÑA using the information from Table 5.1. For this case, the *Generalized Information Criterion* and the *Generalized Bayesian Information Criterion* result in small \hat{K} and small $\hat{\lambda}$. The top and bottom right hand side of Figure 5.5 show very smooth curves (blue line).

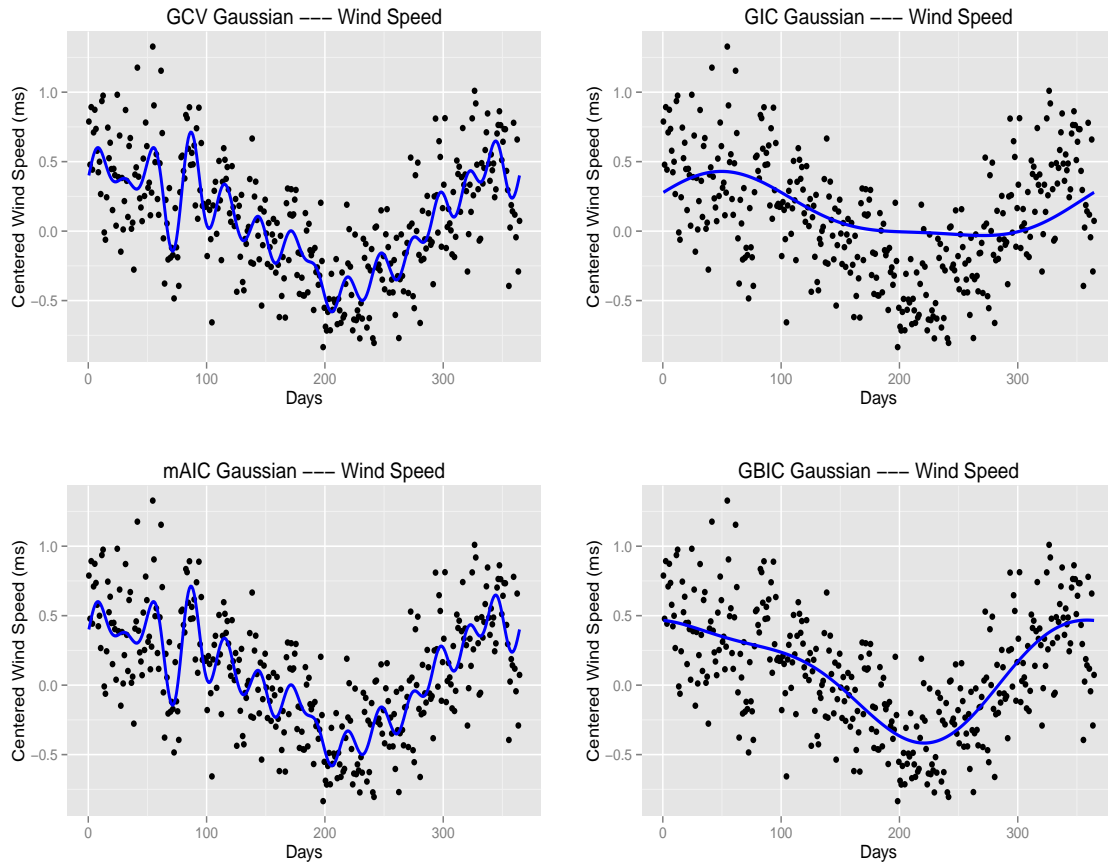


FIGURE 5.5: (a) fitted curve using GCV; (b) fitted curve using GIC; (c) fitted curve using mAIC; (d) fitted curve using GBIC on Wind Speed in A CORUÑA

5.4.3 Log-Precipitation

Table 5.4 shows the optimal values $\log_{10}(\hat{\lambda}_1)$ and $\log_{10}(\hat{\lambda}_2)$ evaluated for all stations using each model criterion for a fixed the number of basis functions. Note that the optimal number of basis functions is quite small for all basis functions and $\log_{10}(\hat{\lambda})$ values are all negatives.

TABLE 5.3: Summary of the model selection on the Log-Precipitation using *Fourier basis functions*

	GCV	GIC	mAIC	GBIC
$\log_{10}(\hat{\lambda}_1)$	-4.83	-1.52	-5.25	-5.17
$\log_{10}(\hat{\lambda}_2)$	-3.92	-1.25	-4.21	-5.17
\hat{K}	7	7	7	5

Figure 5.6 depicts the fitted curves produced on the observed Log-Precipitation at A CORUÑA using the information from Table 5.3. The blue line represents the smooth curve computed using \mathbf{D} and the red line represents the smooth curve derived from $\hat{\mathbf{D}}$.

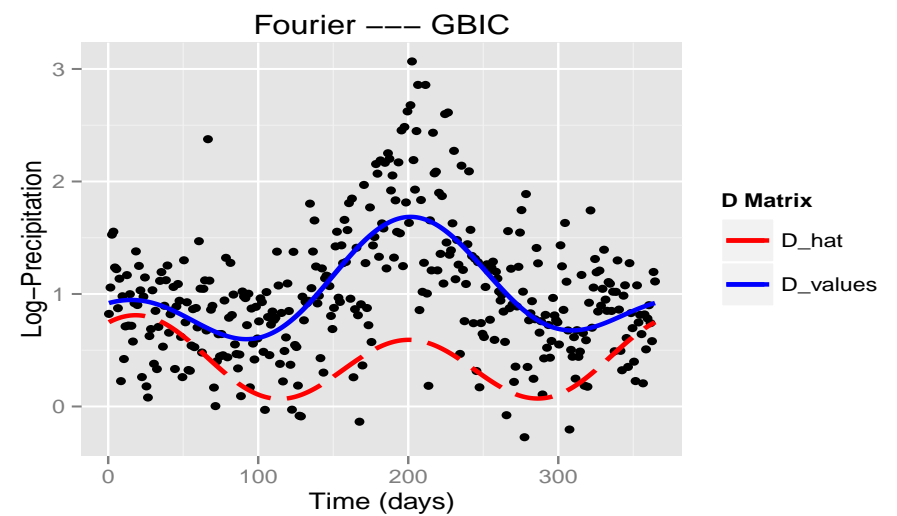
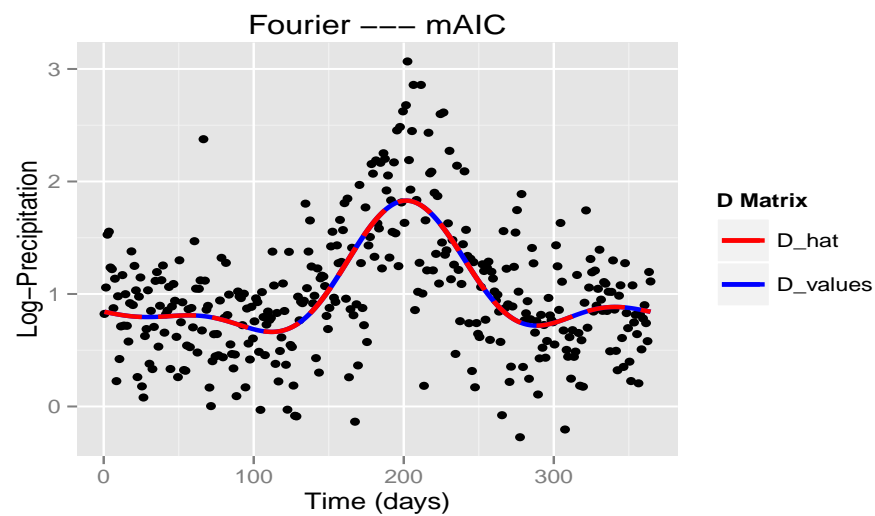
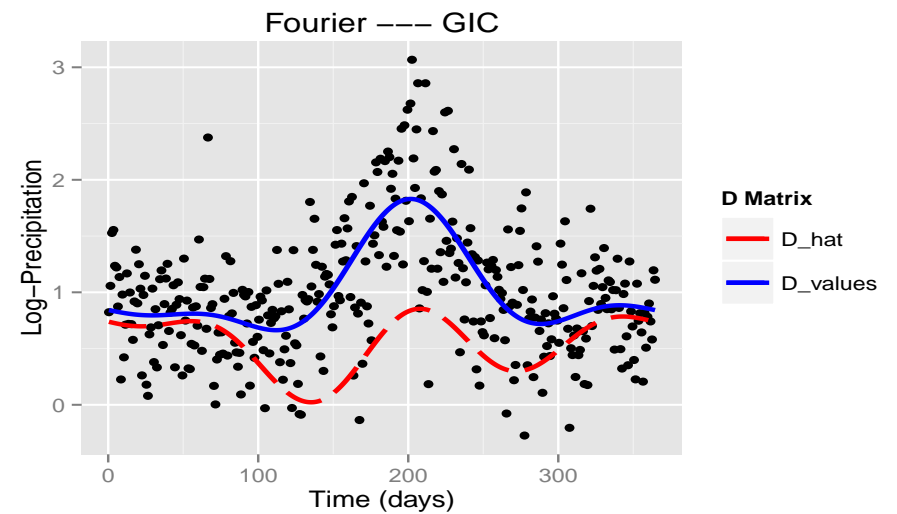
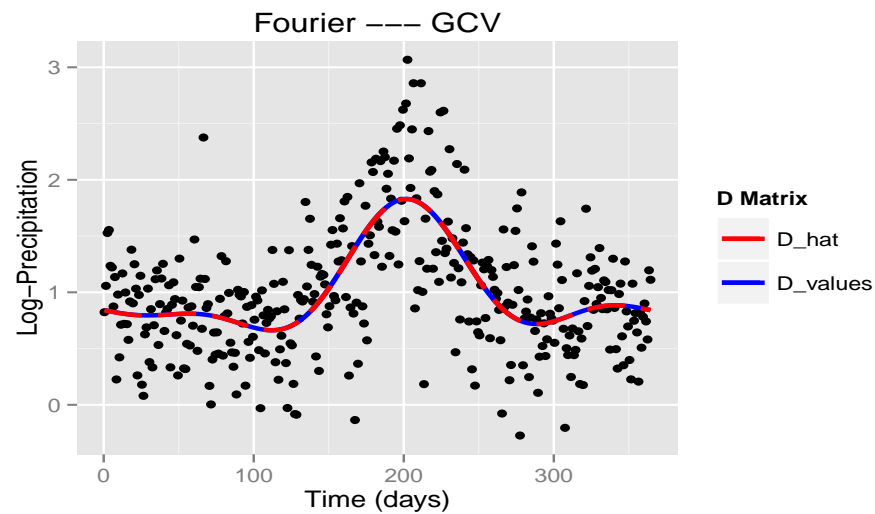


FIGURE 5.6: (a) fitted curve using GCV; (b) fitted curve using GIC; (c) fitted curve using mAIC; (d) fitted curve using GBIC on Log-Precipitation in A CORUÑA

5.5 B-Splines Basis Functions

In this section the modeling is done using *B-Splines basis functions* with *Penalized Maximum Likelihood* estimate to calculate the functional the linear regression model (see equation (5.1)).

5.5.1 Temperature

Figure 5.7 shows the fitted curves produced on the observed Temperature at A CORUÑA using the information from Table 5.1. For this case, once again the *Generalized Information Criterion* has elected small values for K and λ , smaller than the other model criteria. Clearly the small number of basis functions, has an important impact on the shape of the curve.

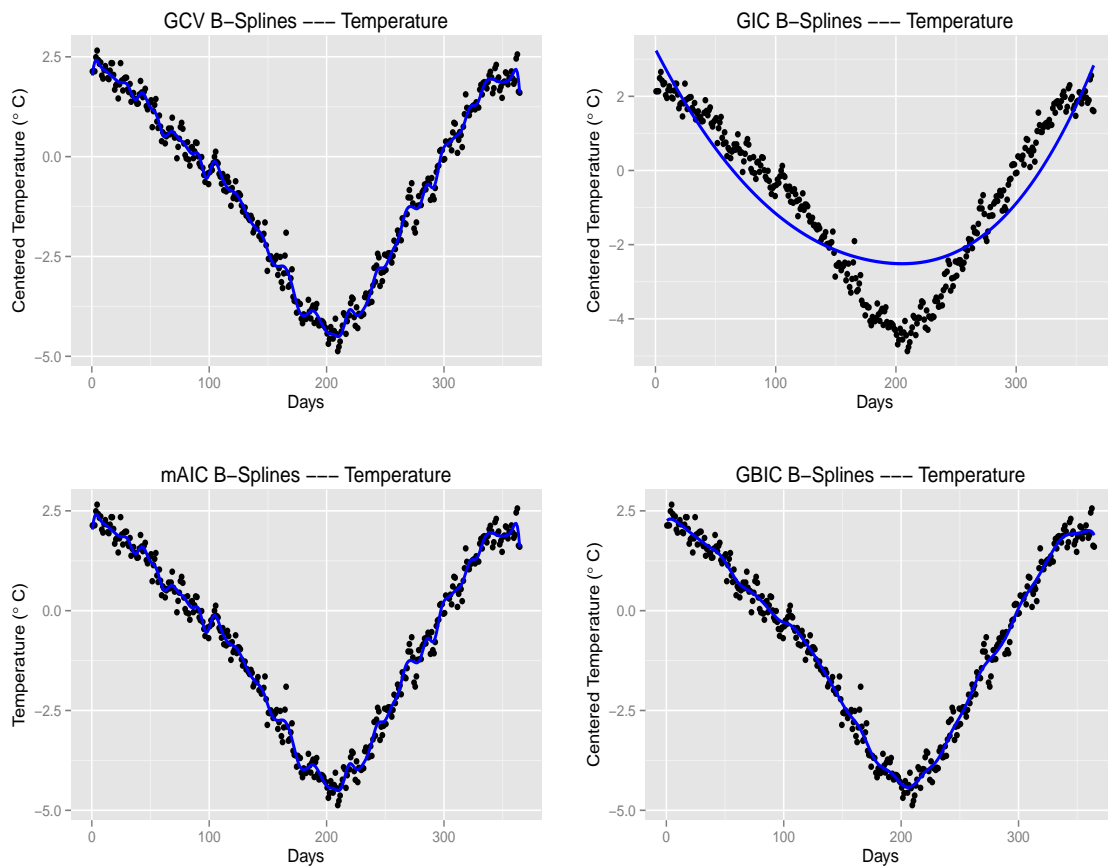


FIGURE 5.7: (a) fitted curve using GCV; (b) fitted curve using GIC; (c) fitted curve using mAIC; (d) fitted curve using GBIC on Temperatures in A CORUÑA

5.5.2 Wind Speed

Figure 5.8 depicts the fitted curves produced on the observed Wind Speed at A CORUÑA using the information from Table 5.1. For this case, the *Generalized Information Criterion* and the *Generalized Bayesian Information Criterion* result in small \hat{K} and small $\hat{\lambda}$. The top and bottom right hand side of Figure 5.5 show very smooth curves (blue line).

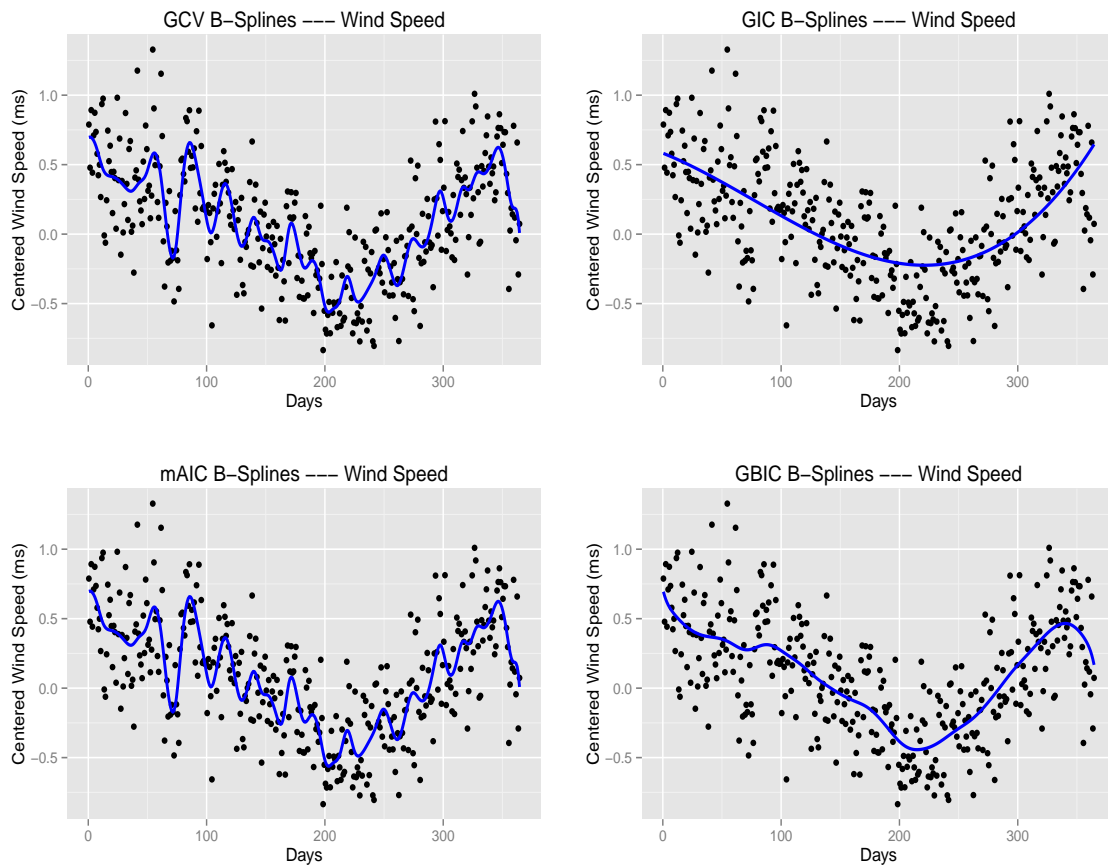


FIGURE 5.8: (a) fitted curve using GCV; (b) fitted curve using GIC; (c) fitted curve using mAIC; (d) fitted curve using GBIC on Wind Speed in A CORUÑA

5.5.3 Log-Precipitation

The computation $\hat{\mathbf{D}}$ and consequently the predicted functional *Log-Precipitation* implies optimizing the matrix of smoothing parameters $\mathbf{\Lambda}$ for fixed values of K given in Table 5.1.

Table 5.4 shows the optimal values $\log_{10}(\hat{\lambda}_1)$ and $\log_{10}(\hat{\lambda}_2)$ evaluated for all stations using each model criterion for a fixed the number of basis functions.

TABLE 5.4: Summary of the model selection on the Log-Precipitation using *B-Splines* basis functions

	GCV	GIC	mAIC	GBIC
$\log_{10}(\hat{\lambda}_1)$	-3.51	-2.15	-3.72	-1.33
$\log_{10}(\hat{\lambda}_2)$	-1.98	-1.42	-2.1	1.23
\hat{K}	65	5	65	63

Figure 5.9 depicts the fitted curves produced on the observed Log-Precipitation at A CORUÑA using the information from Table 5.4. The blue line represents the smooth curve computed using \mathbf{D} and the red line represents the smooth curve derived from $\hat{\mathbf{D}}$.

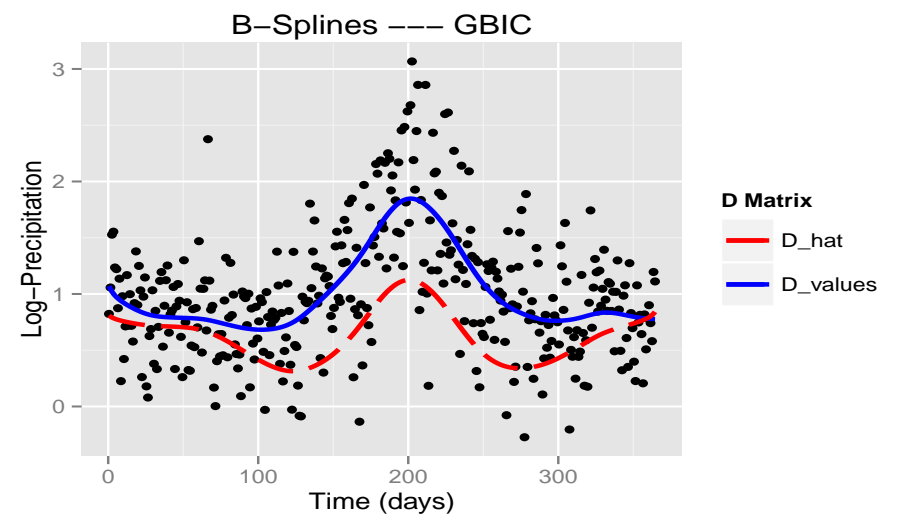
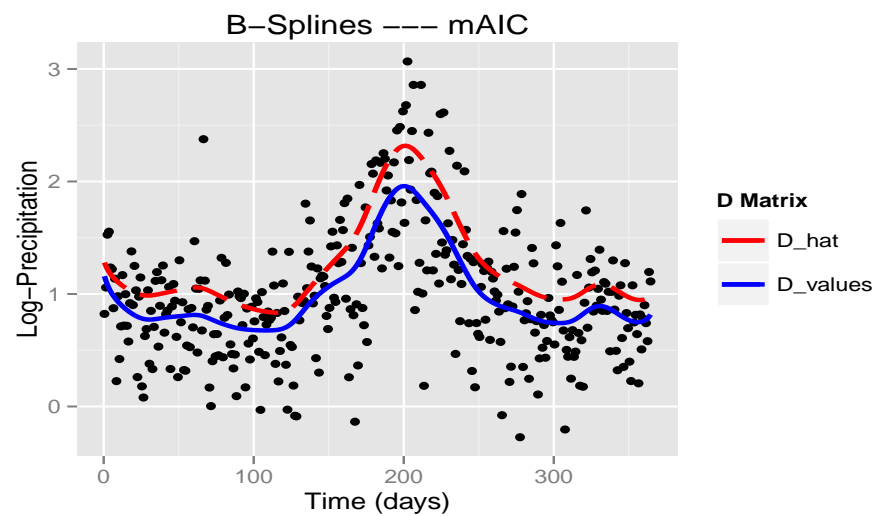
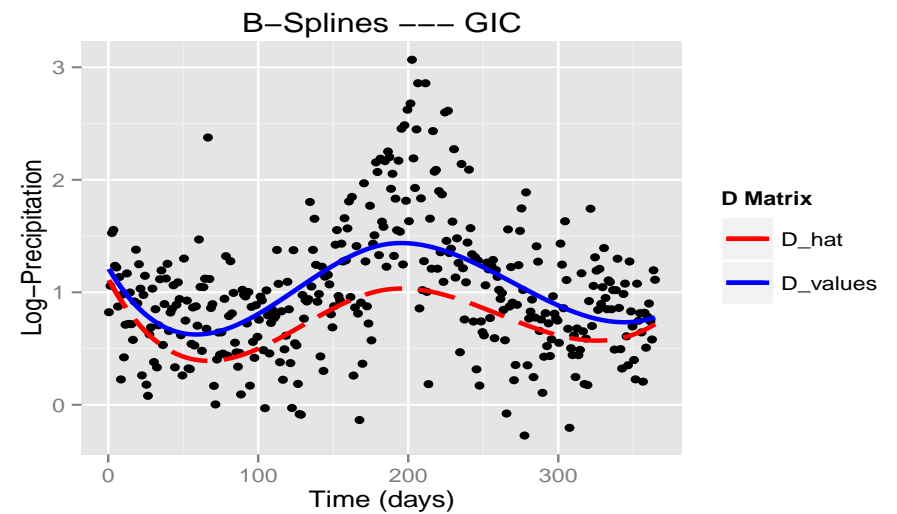
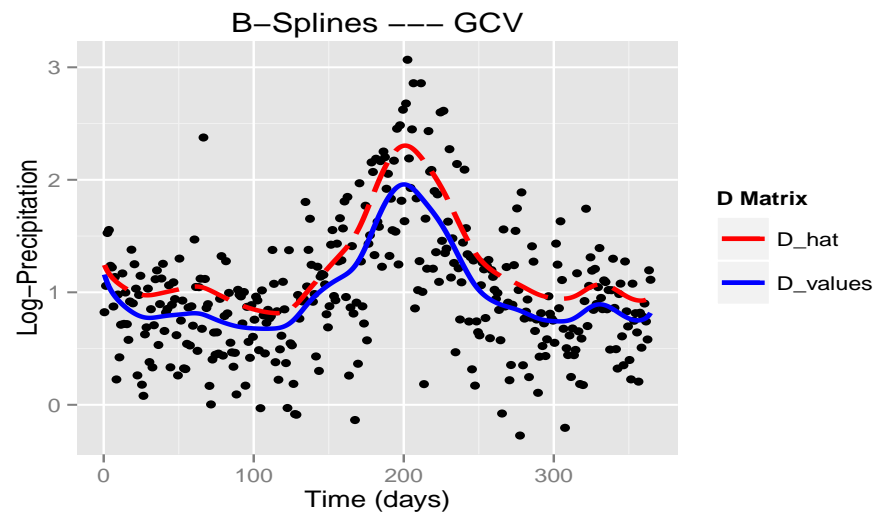


FIGURE 5.9: (a) fitted curve using GCV; (b) fitted curve using GIC; (c) fitted curve using mAIC; (d) fitted curve using GBIC on Log-Precipitation in A CORUÑA

5.6 Discussion of the Results

The objective of this chapter was to illustrate the implementation of a Functional Linear Regression model (5.1) when both covariates and the response variable are functional. The dataset used for illustration was the Spanish weather data. The aim was to model the functional behaviour of the Log-Precipitation when the functional behaviour of the Temperature and Wind Speed were known. The Functional Linear regression Model was computed using the *Penalized Maximum Likelihood* estimate and four model criteria were used to evaluate the model namely: the *Generalized Cross-Validation*; the *Generalized Information Criteria*; the *modified Akaike Information Criteria* and the *Generalized Bayesian Information Criteria*. Three kinds of basis functions were used in the analysis: *Gaussian Basis* function; *Fourier Basis* function and *B-Splines Basis* function. For each kind of basis function and each model criterion, the optimal number of basis functions \hat{K} was estimated as well as the optimal smoothing parameter $\hat{\lambda}$. It was found that, each variable modeled using each kind of basis functions evaluated using each kind model criterion resulted in more or less different values for \hat{K} and the smoothing parameter.

Table 5.5 shows the *Mean Square Error* averaged across all stations, for each model criterion and for each type of basis function. In this context, the *Mean Square Error* is defined as the difference between the actual observed data and the predicted: $\hat{\mathbf{D}}\Phi(t)$. Comparing these values across each model criteria is not a very objective way to reach a meaningful conclusion. In fact, it is sensible to compare the results across all types of basis functions. As given on Table 5.5, the lowest *Average Mean Square Error* (AMSE), considering all models criteria, is found to be at *Gaussian Basis* function. The next type of basis functions is *B-Splines Basis* followed by *Fourier Basis* which appeared to perform the worst out of all three.

TABLE 5.5: *Average Mean Square Error* for the predicted versus observed values of the functional Log-Precipitation

Model Criterion	<i>Gaussian Basis</i>	<i>B-Splines Basis</i>	<i>Fourier Basis</i>
GCV	0.759	0.808	0.772
GIC	1.680	1.750	2.6
mAIC	0.759	0.791	0.772
GBIC	1.24	1.403	1.674

Chapter 6

Conclusion

This Chapter will summarize the results of the research and its applications subject to the dissertation. The Chapter will begin with a discussion of the objectives mentioned in Chapter 1 in a concluding manner. An overview and a brief summary of the illustrations implemented in Chapter 5 are done. Some comments will be made on the hardware specifications used to run the scripts as well as their limitations in performing the analysis. Finally, some recommendations will be made to those interested in future research regarding Functional Linear Regression Modeling.

6.1 Concluding Remarks about Objectives

Below each of the objectives stated in Chapter 1 is discussed.

- The first objective stated in Chapter 1 was to define Functional Data Analysis and introduce important basis functions that were used throughout the dissertation. Chapter 2 offered an in-depth explanation of the different basis expansions used in FDA and provided a visualization aspect to them. The emphasis was on the relevant techniques and methods in this dissertation, namely: *Gaussian*, *Fourier* and *B-Splines*.
- Chapter 2 provided an in-depth understanding of four different model criteria (i.e. *Generalized Cross-Validation*, *Generalized Information Criterion*, *modified Akaike Information Criterion* and *Generalized Bayesian Information Criterion*) as well their computations in R through examples, in the FDA context. Chapter 4 and Chapter 5 provided the insight of the aforementioned model criteria in the FLRM context. Appendix A was fully dedicated in the implementation

of the most recurrent steps done when converting discrete observed data to a functional process.

- Chapter 3 main objective was to provide the mathematical foundations to have a better understanding of the link between Functional Analysis and Functional Data Analysis. Key definitions and theorems were given in order to equip the readers with the relevant background to understand the meaning of the *Kahrunen-Loeve* Theorem.
- In Chapter 4, the Functional Linear Regression was introduced. All the different model estimation methods with a deeper attention on the *Penalized Maximum Likelihood* estimate. The derivations of the maximum likelihood estimators, in every case, were computed. The model criteria in the FLRM context were defined and derived accordingly, see Appendix B for the proofs.
- Chapter 5 provided an application of FLRM using the **Aemet** data. The *Gaussian Basis*, *Fourier Basis* and *B-Splines Basis* were used to smooth the discretized observed data of Temperature, Wind Speed and Log-Precipitation for all 73 stations. For each basis, the four model criteria were used to compute the optimal model parameters. Once the functional data of Log-Precipitation were obtained, the Functional Linear Regression was implemented to predict the functional data of Log-Precipitation (see Figures 5.3, 5.6 and 5.9). The AMSE was computed to compare the different basis functions. Based on the results, it appeared that the *Gaussian Basis* outperformed all the other basis functions.

6.2 Limitations

One of the major issues with performing an analysis in Functional Linear Regression Modeling is the time taken to compute the functional variables. It is clear that when the number of data points J is very large, computing an expansion in $\mathcal{O}(J)$ operations is critical. Additionally, when the operations are performed over a number of sets of observed data (e.g. weather stations) then the computation takes even more time to be completed. Another computationally intensive process is the implementation of model criteria. *Generalized Cross-Validation*, for instance, is known to be computationally intensive and therefore requires a lot of space in memory. This is a particular issue when running codes using the Programming Language R. A typical error message displayed when trying to compute matrix operations would be: **Error:**

cannot allocate vector of size 78.1 Gb.

For this dissertation, the only focus was on Regression in Functional Data Analysis. The functional behaviour of the Log-Precipitation was predicted assuming the functional behaviours of Temperature and Wind Speed were known. This particularity of the **Aemet** data and of weather data in general is difficult to find. In other words, the availability of datasets that have similar features as the weather data is not given. Even when the dataset is available, the discretized observed points are sparse which makes the analysis in functional context strenuous.

6.3 Recommendations

As already mentioned, the analysis of Functional Linear Regression was done using the **Aemet** weather data only. More datasets with similar intrinsic features as the weather data from other fields of research should be investigated to involve more scientists in Functional Data Analysis. The R-package **fda** by Ramsay et al. (2009) has been around for a long time and is the most popular package for FDA. Unfortunately, the package is a bit restricted with other kinds of analysis involving: other types of basis functions (e.g. *Gaussian Basis*); other types of model evaluations methods (e.g. *Penalized Maximum Likelihood* method) and other types model criteria (e.g. GIC, mAIC, GBIC). More packages must be released in that regard.

When it comes to Functional Linear Regression models, one of the main assumption is that the chosen basis function is the one that smooth the predictor and covariates. Which is not always a correct assumption because different variables exhibit different stochastic paths. One of the challenges with violating that assumption is the computation of the matrices $\mathbf{J}_{\phi\psi}$ where $\phi(t)$ could be a *Fourier Basis* function and $\psi(t)$ *Gaussian Basis* function.

Thus there is still vast unexplored area of research in the field of Functional Data Analysis in general and specifically for Functional Regression Modeling. This dissertation provides a first step in that direction.

Appendix A

R-Functions

In order to make some of the R-codes readable, most of the repeated operations have been wrapped up into functions that are used throughout the dissertation.

A.1 Matrices of Basis Functions and Model Selection

Gaussian_bsplines	<i>Gaussian Basis</i> functions with <i>B-Splines</i>
-------------------	-------------------------------------------------------

Description

This function is used to compute a matrix of *Gaussian Basis* functions with *B-Splines*. Its arguments are:

- `tt` being the vector of values $\{t_1, \dots, t_J\} \in \mathcal{T}$;
- `m` represents the number of basis functions applied to the function.

R-Code

```
Gaussian_bsplines = function(tt,m){  
  
  range <- diff(range(tt))  
  kn <- seq(min(tt) - (range/(m-3))*3, max(tt) + (range/(m-3))*3, by = range/(m-3)  
  )  
  myu <- kn[3:(m+2)]  
  h <- diff(kn,lag = 2)/3
```

```

B <- matrix(0,length(tt),(m))
for (j in 1:m){
  B[,j] <- exp(-0.5*(tt-myu[j])^2/(h[1]^2))
}
return(B)
}

```

Gaussian_kmeans

Gaussian Basis functions with K-Means

Description

This function is used to compute a matrix of *Gaussian Basis* functions using *K-means*. Its arguments are:

- `tt` is used to specify the vector of values $\{t_1, \dots, t_J\} \in \mathcal{T}$;
- `m` is used to specify the number of basis functions applied to the function;
- `nyu` is used to specify the hyperparameter.

The clustering method used is the one developed by Hartigan and Wong (1979).

R-Code

```

Gaussian_kmeans = function(tt,m,nyu){

  k <- kmeans(tt, centers = m,algorithm = "Hartigan-Wong")
  myu <- as.vector(k$centers)
  h <- k$withinss/k$size

  B <- matrix(0,length(tt),(m))
  for (j in 1:m){
    B[,j] <- exp(-0.5*(tt-myu[j])^2/(h[j]*nyu))
  }
  return(B)
}

```

Bsplines_FDA	<i>B-Splines Basis</i> functions
--------------	----------------------------------

Description

This function is used to generate a matrix of *B-Splines Basis* functions. It uses the R-package `fda`. Its arguments are:

- `tt` is used to specify the vector of values $\{t_1, \dots, t_J\} \in \mathcal{T}$;
- `m` is used to specify the number of basis functions applied to the function.
- `norder` is used to specify the order of the *B-Splines*

R-Code

```
Bsplines_FDA <- function(tt,m,norder=4){
  require(fda)
  basis = create.bspline.basis(rangeval = range(tt),nbasis = m,norder)
  B <- eval.basis(evalarg = tt,basisobj = basis)
  return(B)
}
```

Fourier_FDA	<i>Fourier Basis</i> functions
-------------	--------------------------------

Description

This function is used to generate a matrix of *Fourier Basis* functions. It uses the R-package `fda`. Its arguments are:

- `tt` is used to specify the vector of values $\{t_1, \dots, t_J\} \in \mathcal{T}$;
- `m` is used to specify the number of basis functions applied to the function.

R-Code

```
Fourier_FDA <- function(tt,m){
  require(fda)
  if((m %% 2)==0) {m <- m + 1} else {m <- m}
  basis = create.fourier.basis(rangeval = range(tt),nbasis = m)
  B <- eval.basis(evalarg = tt,basisobj = basis)
  return(B)
}
```

Pen_Max_Likelihood

Penalized Maximum Likelihood estimate

Description

This function is used to compute the *Penalized Maximum Likelihood* estimate. Its arguments are:

- B is used to specify the matrix of basis functions;
- n is used to specify the number of basis functions;
- lambda is used to specify $\log_{10}(\lambda)$;
- y is used for the vector of observed values.

R-Code

```
Pen_Max_Likelihood <- function(B, n, lambda, y){
  D <- matrix(0,(n-2),n)
  D[1, ] <- c(1,-2,1,rep(0,(n-3)))
  for (i in 1:(n-4)) {
    D[(i+1), ] <- c(rep(0,i),1,-2,1,rep(0,(n-3)-i))
  }
  D[(n-2), ] <- c(rep(0,(n-3)),1,-2,1)
  K <- t(D)%*%D

  lamda <- 10^(lambda)
  sigma <- 2
  sigma1 <- 1

  while((sigma-sigma1)^2 > 1e-7){
    Binv <- solve(t(B)%*%B+ncol(train.temp)*(lamda)*(sigma)*K,diag(ncol(K)))
    w <- (Binv)%*%t(B)%*%y[1,]
    sigma1 <- sigma
    sigma1 <- as.vector(sigma1)
    sigma <- (1/ncol(train.temp))*t(y[1,]-B%*%w)%*%(y[1,]-B%*%w)
  }
```

```

    sigma <- as.vector(sigma)
  }
  list(lamda=lamda,sigma=sigma,K=K,w=w)
}

```

A.2 Model Criterion

<code>gcv_fun</code>	<i>Generalized Cross-Validation</i> criterion
----------------------	-----------------------------------------------

Description

This function is used to compute the *Generalized Cross-Validation* criterion for model evaluation. Its arguments are:

- `ob` is used to specify the object created from the `Pen_Max_Likelihood` function;
- `y` is used for the vector of observed values;
- `tt` is used to specify the vector of values $\{t_1, \dots, t_J\} \in \mathcal{T}$.

R-Code

```

gcv_fun <- function(tt, y, ob){
  Binv <- solve(t(B)%*%B+length(y)*(ob$lamda)*(ob$sigma)*ob$K,diag(n))
  H <- B%*%(Binv)%*%t(B)
  yhat <- H%*%y[1,]
  den = 1 - sum(diag(H))/length(tt) # load(matrixcalc)
  y.diff = yhat - y[1,]
  return(mean((y.diff/den)^2))
}

```

<code>gic_fun</code>	<i>Generalized Information Criterion</i>
----------------------	------------------------------------------

Description

This function is used to compute the *Generalized Information Criterion* for model evaluation. Its arguments are:

- `ob` is used to specify the object created from the `Pen_Max_Likelihood` function;
- `y` is used for the vector of observed values;
- `tt` is used to specify the vector of values $\{t_1, \dots, t_J\} \in \mathcal{T}$.

R-Code

```
gic_fun <- function(y,ob,n){
  gamma <- diag(as.vector(y[1,]-B%*%ob$w))
  one <- rep(1,length(y))

  R1 <- rbind(t(B)%*(B+length(y)*(ob$lamda)*(ob$sigma)*ob$K,t(one))%*gamma%*B/(ob$
    sigma))
  R2 <- rbind(t(B)%*gamma%*one/(ob$sigma),length(y)/(2*(ob$sigma)))
  R <- cbind(R1,R2)
  R <- R/(length(y)*(ob$sigma))
  if(det(R) < 10^(103)) {Rinv <- solve(R,diag(n+1))} else {Rinv <- NA}

  Q1 <- rbind(t(B)%*(gamma)^2%*B/(ob$sigma)-(ob$lamda)*ob$K%*ob$w%*t(one)%*
    gamma%*B,t(one)%*(gamma)^3%*B/(2*(ob$sigma)^2)-t(one)%*gamma%*B/(2*(ob$
    sigma)))
  Q2 <- rbind(t(B)%*(gamma)^3%*one/(2*(ob$sigma)^2)-t(B)%*gamma%*one/(2*(ob$
    sigma)),t(one)%*(gamma)^4%*one/(4*(ob$sigma)^3)-length(y)/(4*(ob$sigma)))
  Q <- cbind(Q1,Q2)
  Q <- Q/(length(y)*(ob$sigma))

  V <- ifelse(det(R) < 10^(103) & all(!is.na(Rinv)), length(y)*(log(2*pi)+1)+length(
    y)*log(ob$sigma)+2*sum(diag(Rinv%*%Q)), NA)
  return(V)
}
```

`mAIC_fun`

modified Akaike Information Criterion

Description

This function is used to compute the *modified AIC* method for model evaluation. Its arguments are:

- `ob` is used to specify the object created from the `Pen_Max_Likelihood` function;
- `y` is used for the vector of observed values;
- `tt` is used to specify the vector of values $\{t_1, \dots, t_J\} \in \mathcal{T}$.

R-Code

```
mAIC_fun <- function(ob,y,n){
  Binv <- solve(t(B)%*%B+length(y)*(ob$lamda)*(ob$sigma)*ob$K,diag(n))
  H <- B%*%(Binv)%*%t(B)
  return(length(y)*(log(2*pi)+1)+length(y)*log(ob$sigma)+2*sum(diag(H)))
}
```

gbic_fun	<i>Generalized Bayesian Information Criterion</i>
----------	---------------------------------------------------

Description

This function is used to compute the *Generalized Bayesian Information Criterion* for model evaluation. Its arguments are:

- `ob` is used to specify the object created from the `Pen_Max_Likelihood` function;
- `y` is used for the vector of observed values;
- `tt` is used to specify the vector of values $\{t_1, \dots, t_J\} \in \mathcal{T}$.

The R-code is as follows:

```
gbic <- function(y,ob,n){
  gamma <- as.vector(y[1,]-B%*%ob$w)

  Q1 <- rbind(t(B)%*%B+length(y)*(ob$lamda)*(ob$sigma)*ob$K,t(gamma)%*%B/(ob$sigma))
  Q2 <- rbind(t(B)%*%gamma/(ob$sigma),length(y)/(2*(ob$sigma)))
  Q <- cbind(Q1,Q2)
  Q <- Q/(length(y)*(ob$sigma))
  Q.det <- det(Q)
  vec <- eigen(ob$K)$values
  vec <- vec[vec >= 0]

  return((length(y)+n-1)*log(ob$sigma) + length(y)*(ob$lamda)*(ob$sigma)*t(ob$w)%*%
    ob$K%*%ob$w/(ob$sigma) + length(y) + (length(y)-3)*log(2*pi)+
    3*log(length(y)) + log(Q.det) - log(prod(vec)) - (n-1)*log((ob$lamda)*(ob$sigma)
    ))
}
```

GCV.Gauss_bs

*Generalized Cross-Validation*²

Description

This function computes the *Generalized Cross-Validation* criterion using the *Least Squares* method without a smoothing parameter based Gaussian basis function with *B-Splines*. Its arguments are:

- `dat` is used to specify the $N \times J$ matrix of observations;
- `tt` is used to specify the vector of values $\{t_1, \dots, t_J\} \in \mathcal{T}$;
- `m` is used to specify the number of basis functions applied to the function.

This function is used run for illustrative purpose (see section 2.6.2)

R-Code

```
S = NULL
GCV.Gauss_bs = function(dat,tt,m){

  range <- diff(range(tt))
  kn <- seq(min(tt) - (range/(m-3))*3, max(tt) + (range/(m-3))*3, by = range/(m-3))
  myu <- kn[3:(m+2)]
  h <- diff(kn,lag = 2)/3

  B <- matrix(0,length(tt),(m))
  for (j in 1:m){
    B[,j] <- exp(-0.5*(tt-myu[j])^2/(h[1]^2))
  }
  Binv <- solve(t(B)%*%B,diag(m))
  S <- B%*%Binv%*%t(B)
  xhat <- S%*%dat
  den <- 1 - sum(diag(S))/length(tt)
  x.diff <- xhat - dat
  return(mean((x.diff/den)^2)) # GCV value
}
```

Appendix B

Derivations and Proofs

B.1 Karhunen-Loeve proofs

The coefficients in theorem 3.5.1 satisfy the following:

1. $\mathbb{E}[x_i] = 0, \forall i \in \mathbb{N};$
2. $\mathbb{E}[x_i x_j] = \delta_{ij} \lambda_j, \forall i, j \in \mathbb{N};$
3. $\text{Var}[x_j] = \lambda_i,$

Proof

1. $\mathbb{E}[x_i] = 0, \forall i \in \mathbb{N}$

$$\begin{aligned}\mathbb{E}[x_i] &= \mathbb{E}\left[\int_D X_t e_i(t) dt\right] \\ &= \int_{\Omega} \int_D X_t(\omega) e_i(t) dt d\mathbb{P}(\omega) \\ &= \int_D \int_{\Omega} X_t(\omega) e_i(t) d\mathbb{P}(\omega) dt \text{ (Fubini)} \\ &= \int_D \mathbb{E}[X_t] e_i(t) dt = 0 \text{ (} X_t \text{ is a centered process)} \quad \blacksquare\end{aligned}$$

$$2. \mathbb{E}[x_i x_j] = \delta_{ij} \lambda_j, \quad \forall i, j \in \mathbb{N}$$

$$\begin{aligned}
\mathbb{E}[x_i x_j] &= \mathbb{E} \left[\left(\int_D X_s e_i(s) ds \right) \left(\int_D X_t e_j(t) dt \right) \right] \\
&= \mathbb{E} \left[\int_D \int_D X_s e_i(s) X_t e_j(t) ds dt \right] \\
&= \int_D \int_D \mathbb{E}[X_s X_t] e_i(s) e_j(t) ds dt \\
&= \int_D \left(\int_D \Psi(s, t) e_j(t) dt \right) e_i(s) ds \\
&= \int_D [\Psi e_j](s) e_i(s) ds \text{ from (3.9)} \\
&= \langle \Psi e_j, e_i \rangle \\
&= \langle \lambda_j e_j, e_i \rangle \\
&= \lambda_j \delta_{ij} \quad \blacksquare
\end{aligned}$$

$$3. \text{Var}[x_j] = \lambda_j, \quad \forall j \in \mathbb{N}$$

$$\begin{aligned}
\text{Var}[x_i] &= \mathbb{E}[(x_i - \mathbb{E}[x_i])^2] \\
&= \mathbb{E}[x_i^2] \quad (\text{because } \mathbb{E}[x_i] = 0) \\
&= \lambda_i \quad \blacksquare
\end{aligned}$$

Proof of theorem 3.5.1

Let Ψ be the Hilbert-Schmidt operator defined as in section 3.9. Ψ has a complete set of eigenvectors $\{e_i\}$ in $L^2(D)$ and non-negative eigenvalues $\{\lambda_i\}$. Consider the following equation:

$$\epsilon_n(t) := \mathbb{E} \left[\left(X_t - \sum_{i=1}^n x_i e_i(t) \right)^2 \right].$$

The rest of the proof results in showing that $\lim_{n \rightarrow \infty} \epsilon_n(t) = 0$ uniformly in D

$$\begin{aligned}
\epsilon_n(t) &= \mathbb{E} \left[\left(X_t - \sum_{i=1}^n x_i e_i(t) \right)^2 \right] \\
&= \mathbb{E}[X_t^2] - 2\mathbb{E} \left[X_t \sum_{i=1}^n x_i e_i(t) \right] + \mathbb{E} \left[\sum_{i,j=1}^n x_i x_j e_i(t) e_j(t) \right]
\end{aligned}$$

$\mathbb{E}[X_t^2] = \Psi(t, t)$, and

$$\begin{aligned}
 \mathbb{E} \left[X_t \sum_{i=1}^n x_i e_i(t) \right] &= \mathbb{E} \left[X_t \sum_{i=1}^n \left(\int_D X_s e_i(s) ds \right) e_i(t) \right] \\
 &= \sum_{i=1}^n \left(\int_D \mathbb{E}[X_t X_s] e_i(s) ds \right) e_i(t) \\
 &= \sum_{i=1}^n \left(\int_D \Psi(t, s) e_i(s) ds \right) e_i(t) \\
 &= \sum_{i=1}^n [\Psi e_i](t) e_i(t) \\
 &= \sum_{i=1}^n \lambda_i e_i(t)^2
 \end{aligned}$$

In a similar fashion, $\mathbb{E} \left[\sum_{i,j=1}^n x_i x_j e_i(t) e_j(t) \right] = \sum_{i=1}^n \lambda_i e_i(t)^2$. Therefore,

$$\epsilon_n(t) = \Psi(t, t) - \sum_{i=1}^n \lambda_i e_i(t) e_i(t)$$

By invoking the Mercer's Theorem,

$$\lim_{n \rightarrow \infty} \epsilon_n(t) = 0 \quad \blacksquare$$

B.2 Derivation of \mathbf{J} -matrix

$\mathbf{J}_{\phi_1 \phi_2}$ is a square matrix involving the cross-product of vectors of basis functions $\phi_1(t)$ and $\phi_2(t)$ with length m . It is defined as $\mathbf{J}_{\phi_1 \phi_2} = \int_{\mathcal{T}} \phi_1(t) \phi_2'(t) dt$. If the basis functions are orthogonal (e.g. Fourier basis, B-Splines basis, etc...) then $\mathbf{J}_{\phi_1 \phi_2} = \mathbb{I}_m$ with m being the length of vectors of basis functions. If the basis functions are not orthogonal (e.g. Gaussian basis), then $\mathbf{J}_{\phi_1 \phi_2}$ is evaluated analytically or numerically. Let $\phi_1(t; \mu_1, \sigma_1^2) = \exp \left(-\frac{(t - \mu_1)^2}{2\sigma_1^2} \right)$ and $\phi_2(t; \mu_2, \sigma_2^2) = \exp \left(-\frac{(t - \mu_2)^2}{2\sigma_2^2} \right)$ be two Gaussian basis functions. Then the ij^{th} element of the matrix $\mathbf{J}_{\phi_1 \phi_2}$ is expressed as follows:

$$\begin{aligned}
\int_{\mathcal{T}} \phi_1(t) \phi_2(t) dt &= \int_{\mathcal{T}} \exp\left(-\frac{(t-\mu_1)^2}{2\sigma_1^2}\right) \exp\left(-\frac{(t-\mu_2)^2}{2\sigma_2^2}\right) dt \\
&= \int_{\mathcal{T}} \exp\left(-\left[\frac{(t-\mu_1)^2}{2\sigma_1^2} + \frac{(t-\mu_2)^2}{2\sigma_2^2}\right]\right) dt \\
&= \int_{\mathcal{T}} \exp\left(-\left[\frac{(\sigma_1^2 + \sigma_2^2)t^2 - 2(\mu_1\sigma_2^2 + \mu_2\sigma_1^2)t + \mu_2^2\sigma_1^2 + \mu_1^2\sigma_2^2}{2\sigma_1^2\sigma_2^2}\right]\right) dt \\
&= \int_{\mathcal{T}} \exp\left(-\left[\frac{t^2 - 2\frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}t + \frac{\mu_2^2\sigma_1^2 + \mu_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}}\right]\right) dt
\end{aligned}$$

Let $\sigma_{12} = \sqrt{\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}}$, $\mu_{12} = \frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$ and $\zeta = \frac{t^2 - 2\frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}t + \frac{\mu_2^2\sigma_1^2 + \mu_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}}$.

Suppose that κ is the term required to complete the square in ζ i.e.

$$\kappa = \frac{\left(\frac{\mu_2^2\sigma_1^2 + \mu_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)^2 - \left(\frac{\mu_2^2\sigma_1^2 + \mu_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)^2}{\frac{2\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}} = 0$$

Adding this term to ζ gives

$$\begin{aligned}
\zeta &= \frac{t^2 - 2\frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}t + \left(\frac{\mu_2^2\sigma_1^2 + \mu_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)^2}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}} + \frac{\frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2} - \left(\frac{\mu_2^2\sigma_1^2 + \mu_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\right)^2}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}} \\
&= \frac{\left(t - \frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\right)^2}{2\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}} + \frac{(\mu_1 - \mu_2)^2}{2(\sigma_1^2 + \sigma_2^2)} \\
&= \frac{(t - \mu_{12})^2}{2\sigma_{12}^2} + \frac{(\mu_1 - \mu_2)^2}{2(\sigma_1^2 + \sigma_2^2)}
\end{aligned} \tag{B.1}$$

Therefore,

$$\begin{aligned}
\int_{\mathcal{T}} \phi_1(t) \phi_2(t) dt &= \int_{\mathcal{T}} \exp \left[\frac{(t - \mu_{12})^2}{2\sigma_{12}^2} \right] \exp \left[\frac{(\mu_1 - \mu_2)^2}{2(\sigma_1^2 + \sigma_2^2)} \right] dt \\
&= \exp \left[\frac{(\mu_1 - \mu_2)^2}{2(\sigma_1^2 + \sigma_2^2)} \right] \int_{\mathcal{T}} \exp \left[\frac{(t - \mu_{12})^2}{2\sigma_{12}^2} \right] dt \\
&= \exp \left[\frac{(\mu_1 - \mu_2)^2}{2(\sigma_1^2 + \sigma_2^2)} \right] \sqrt{2\pi\sigma_{12}^2} \int_{\mathcal{T}} \frac{1}{\sqrt{2\pi\sigma_{12}^2}} \exp \left[\frac{(t - \mu_{12})^2}{2\sigma_{12}^2} \right] dt \\
&= \exp \left[\frac{(\mu_1 - \mu_2)^2}{2(\sigma_1^2 + \sigma_2^2)} \right] \sqrt{2\pi\sigma_{12}^2} \\
&= \sqrt{2\pi} \exp \left[\frac{(\mu_1 - \mu_2)^2}{2(\sigma_1^2 + \sigma_2^2)} \right] \sqrt{\frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}} \quad \blacksquare
\end{aligned} \tag{B.2}$$

B.3 Derivation of $\mathbf{R}_{\Lambda}(\theta)$ matrix

In this section an assiduous derivation of $\mathbf{R}_{\Lambda}(\theta)$ is done. The first derivatives of the log-likelihood function defined in section 4.3.3 are:

$$\frac{\partial l_{\Lambda}(\theta)}{\partial \mathbf{B}} = (\mathbf{Z}^T \mathbf{D} \Sigma^{-1}) - (\mathbf{Z}^T \mathbf{Z} \mathbf{B} \Sigma^{-1}) - N (\Lambda_M \odot \Omega) \mathbf{B}$$

and

$$\frac{\partial l_{\Lambda}(\theta)}{\partial \Sigma^{-1}} = \frac{N}{2} \Sigma - \frac{1}{2} (\mathbf{D} - \mathbf{Z} \mathbf{B})^T (\mathbf{D} - \mathbf{Z} \mathbf{B}).$$

Hence, the second derivatives are with respect to $\{\mathbf{B}, \Sigma^{-1}\}$ are given by:

$$\begin{aligned}
\frac{\partial^2 l_{\Lambda}(\theta)}{\partial \mathbf{B} \partial \mathbf{B}^T} &= \mathbf{Z}^T \hat{\Sigma}^{-1} \mathbf{Z} - N (\Lambda_M \odot \Omega) : \mathbf{R}_{\Lambda \Lambda}^{11}(\theta) \\
\frac{\partial^2 l_{\Lambda}(\theta)}{(\partial \mathbf{B})(\partial \Sigma^{-1})^T} &= \mathbf{Z}^T \mathbf{D} + \mathbf{Z}^T \mathbf{Z} \hat{\mathbf{B}} : \mathbf{R}_{\Lambda \Lambda}^{21}(\theta) \\
\frac{\partial^2 l_{\Lambda}(\theta)}{(\partial \Sigma^{-1})(\partial \mathbf{B})^T} &= N \mathbf{D}^T \mathbf{Z} + N \hat{\mathbf{B}}^T \mathbf{Z}^T \mathbf{Z} : \mathbf{R}_{\Lambda \Lambda}^{12}(\theta) \\
\frac{\partial^2 l_{\Lambda}(\theta)}{(\partial \Sigma^{-1})(\partial \Sigma^{-1})^T} &= \frac{N}{2} \mathbb{I}_{K_y} : \mathbf{R}_{\Lambda \Lambda}^{22}(\theta)
\end{aligned}$$

B.4 Derivation of $\mathbf{Q}_\Lambda(\theta)$ matrix

This section helps to understand the derivation of $\mathbf{Q}_\Lambda(\theta)$. The first derivatives of the log-likelihood function $l_\Lambda(\theta)$ are:

$$\begin{aligned}\frac{\partial l_\Lambda(\theta)}{\partial \mathbf{B}} &= (\mathbf{Z}^T \mathbf{D} \Sigma^{-1}) - (\mathbf{Z}^T \mathbf{Z} \mathbf{B} \Sigma^{-1}) - N (\Lambda_M \odot \Omega) \mathbf{B} \\ \frac{\partial l_\Lambda(\theta)}{\partial \Sigma^{-1}} &= \frac{N}{2} \Sigma - \frac{1}{2} (\mathbf{D} - \mathbf{Z} \mathbf{B})^T (\mathbf{D} - \mathbf{Z} \mathbf{B}).\end{aligned}$$

The first derivatives of the log-likelihood $l(\mathbf{Y}|\theta)$ are:

$$\begin{aligned}\frac{\partial l(\mathbf{Y}|\theta)}{\partial \mathbf{B}} &= \mathbf{Z}^T \mathbf{D} \Sigma^{-1} - \mathbf{Z}^T \mathbf{Z} \mathbf{B} \Sigma^{-1} \\ \frac{\partial l(\mathbf{Y}|\theta)}{\partial \Sigma^{-1}} &= \frac{N}{2} \Sigma - \frac{1}{2} (\mathbf{D} - \mathbf{Z} \mathbf{B})^T (\mathbf{D} - \mathbf{Z} \mathbf{B})\end{aligned}$$

Hence, $\mathbf{Q}_\Lambda(\theta)$ is given by:

$$\begin{aligned}\mathbf{Q}_\Lambda^{11}(\theta) &= \left[\frac{\partial l_\Lambda(\theta)}{\partial \mathbf{B}} \right] \left[\frac{\partial l(\mathbf{Y}|\theta)}{\partial \mathbf{B}} \right]^T \\ \mathbf{Q}_\Lambda^{21}(\theta) &= \left[\frac{\partial l_\Lambda(\theta)}{\partial \mathbf{B}} \right] \left[\frac{\partial l(\mathbf{Y}|\theta)}{\partial \Sigma^{-1}} \right]^T \\ \mathbf{Q}_\Lambda^{12}(\theta) &= \left[\frac{\partial l_\Lambda(\theta)}{\partial \Sigma^{-1}} \right] \left[\frac{\partial l(\mathbf{Y}|\theta)}{\partial \mathbf{B}} \right]^T \\ \mathbf{Q}_\Lambda^{22}(\theta) &= \left[\frac{\partial l_\Lambda(\theta)}{\partial \Sigma^{-1}} \right] \left[\frac{\partial l(\mathbf{Y}|\theta)}{\partial \Sigma^{-1}} \right]^T\end{aligned}$$

Bibliography

- Ando, T., Konishi, S. and Imoto, S. (2008). Nonlinear regression modeling via regularized radial basis function networks, *Journal of Statistical Planning and Inference* **138**: 3616–3633.
- Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press, Inc., New York, NY, USA.
- Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots, *Journal of the American Statistical Association* **74**: 829–836.
- Craven, P. and Wahba, G. (1978). Smoothing noisy data with spline functions, *Numerische Mathematik* **31**: 377–403.
- De Boor, C. (2001). *A practical guide to splines; rev. ed.*, Applied mathematical sciences, Springer, Berlin.
- Fan, J. and Gijbels, I. (1995). Data-driven bandwidth selection in local polynomial fitting: Variable bandwidth and spatial adaptation, *Journal of the Royal Statistical Society, Series B, Methodological* **57**: 371–394.
- Febrero-Bande, M. and Oviedo de la Fuente, M. (2012). Statistical computing in functional data analysis: The R package fda.usc, *Journal of Statistical Software* **51**(4): 1–28.
URL: <http://www.jstatsoft.org/v51/i04/>
- Fujikoshi, Y. and Satoh, K. (1997). Modified aic and cp in multivariate linear regression, *Biometrika* **84**(3): pp. 707–716.
URL: <http://www.jstor.org/stable/2337590>
- Gasser, T. and Müller, H. (1979). *Kernel estimation of regression functions*, Springer-Verlag, New York, pp. 23–68.
- Gasser, T. and Müller, H. (1984). Estimating regression functions and their derivatives by the kernel method, *Scandinavian Journal of Statistics* **11**: 171–185.

- Gauss, C. F. (1809). *Theoria motus corporum coelestium*, Sumtibus F. Perthes et I.H. Besser Hamburgi.
- Gohberg, I., Goldberg, S. and Kaashoek, M. A. (1990). *Classes of linear operators. Vol. I*, Vol. 49 of *Operator Theory: Advances and Applications*, Birkhäuser Verlag, Basel.
URL: *Gohberg I., Goldberg S., Kaashoek M. Classes of Linear Operators, Vol.1.djvu*
- Good, I. J. and Gaskins, R. A. (1971). Nonparametric roughness penalties for probability densities, *Biometrika* **58**(2): pp. 255–277.
URL: <http://www.jstor.org/stable/2334515>
- Green, P. J. and Silverman, B. W. (1994). *Nonparametric regression and generalized linear models: a roughness penalty approach*, Chapman Hall.
- Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**(1): pp. 100–108.
URL: <http://www.jstor.org/stable/2346830>
- Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*, 2 edn, Springer.
URL: <http://www-stat.stanford.edu/~tibs/ElemStatLearn/>
- Härdle, W. (1994). *Applied Nonparametric Regression*, Econometric Society Monographs, Cambridge University Press.
- Kawano, S. and Konishi, S. (2007). Nonlinear regression modeling via regularized gaussian basis functions, *Bull. Inform. Cybern.* **39**: 83–96.
URL: <http://ci.nii.ac.jp/naid/120001944234/en/>
- Knight, M. N. M. (2012). *adlift: An adaptive lifting scheme algorithm*. R package version 1.3-2.
URL: <http://CRAN.R-project.org/package=adlift>
- Konishi, S., Ando, T. and Imoto, S. (2004). Bayesian information criteria and smoothing parameter selection in radial basis function networks, *Biometrika* **91**(1): pp. 27–43.
URL: <http://www.jstor.org/stable/20441077>
- Konishi, S. and Kitagawa, G. (1996). Generalised information criteria in model selection, *Biometrika* **83**(4): pp. 875–890.
URL: <http://www.jstor.org/stable/2337290>

Konishi, S. and Kitagawa, G. (2008). *Information criteria and statistical modeling*, Springer series in statistics, Springer, New York.

URL: <http://www.springer.com/statistics/statistical+theory+and+methods/book/978-0-387-71886-6>

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency, *Ann. Math. Statist.* **22**(1): 79–86.

URL: <http://dx.doi.org/10.1214/aoms/1177729694>

Legendre, A. M. (1805). *Nouvelles methodes pour la determination des orbites des cometes*, F. Didot Paris.

Matsui, H., Kawano, S. and Konishi, S. (2009). Regularized functional regression modeling for functional response and predictors, **1**(1): 17–25.

URL: <http://j-mi.org/articles/view/52>

Moody, J. and Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units, *Neural Comput.* **1**(2): 281–294.

URL: <http://dx.doi.org/10.1162/neco.1989.1.2.281>

Nadaraya, E. A. (1964). On estimating regression, *Theory of Probability and its Applications* **9**: 141–142.

Ramsay, J. O. and Dalzell, C. J. (1991). Some Tools for Functional Data Analysis (with discussion), *Journal of the Royal Statistical Society.* **53**(3): 539–572.

URL: <http://dx.doi.org/10.2307/2345586>

Ramsay, J. O., Hooker, G. and Graves, S. (2009). *Functional Data Analysis with R and MATLAB*, 1st edn, Springer Publishing Company, Incorporated.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis*, Springer Series in Statistics, 2nd edn, Springer.

Rao, C. R. and Wu, Y. (2001). *On model selection*, Vol. Volume 38 of *Lecture Notes–Monograph Series*, Institute of Mathematical Statistics, Beachwood, OH, pp. 1–57.

URL: <http://dx.doi.org/10.1214/lnms/1215540960>

Revolution Analytics and Steve Weston (2014a). *doMC: Foreach parallel adaptor for the multicore package*. R package version 1.3.3.

URL: <http://CRAN.R-project.org/package=doMC>

Revolution Analytics and Steve Weston (2014b). *doParallel: Foreach parallel adaptor for the parallel package*. R package version 1.0.8.

URL: <http://CRAN.R-project.org/package=doParallel>

Revolution Analytics and Steve Weston (2014c). *doSNOW: Foreach parallel adaptor for the snow package*. R package version 1.0.12.

URL: <http://CRAN.R-project.org/package=doSNOW>

Revolution Analytics and Steve Weston (2014d). *foreach: Foreach looping construct for R*. R package version 1.4.2.

URL: <http://CRAN.R-project.org/package=foreach>

Ruppert, D. and Wand, M. P. (1994). Multivariate locally weighted least squares regression, *The Annals of Statistics* **22**: 1346–1370.

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): pp. 1–52.

URL: <http://www.jstor.org/stable/2345542>

Walker, J. S. (2008). *A Primer on Wavelets and Their Scientific Applications, Second Edition*, Studies in Advanced Mathematics, Taylor & Francis.

URL: <http://books.google.co.za/books?id=68uwQgAACAAJ>

Watson, G. S. (1964). Smooth regression analysis, *Sankhyā Ser.* **26**: 359–372.