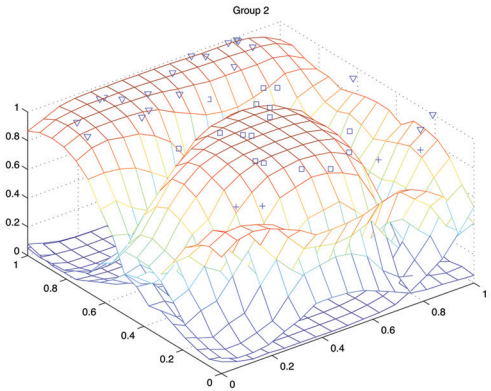
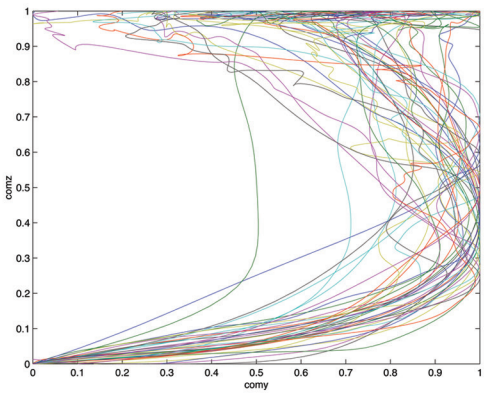
Gaussian Process Regression Analysis for Functional Data





Jian Qing Shi
Taeryon Choi

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Gaussian Process Regression Analysis for Functional Data

This page intentionally left blank

Gaussian Process Regression Analysis for Functional Data

Jian Qing Shi
Taeryon Choi


CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Version Date: 20110502

International Standard Book Number-13: 978-1-4398-3774-0 (eBook - PDF)

Contents

This page intentionally left blank

List of Figures

This page intentionally left blank

List of Tables

This page intentionally left blank

Preface

Over the last decade or so, we have seen a rapid development of functional data analysis, along with applications in a wide range of areas in both sciences and social sciences. Among the vast literature, one of the most influential works is undoubtedly the series of monographs published by Prof. Ramsay, Prof. Silverman and their colleagues (Ramsay and Silverman, 2002, 2005; Ramsay et al., 2009). Functional regression analysis is one of the most important topics in functional data analysis, and this book will discuss nonparametric statistical methods for functional regression analysis, specifically the methods based on a Gaussian process prior in a functional space. We will focus on problems involving functional response variables and mixed covariates of functional and scalar variables. The exposition of nonparametric functional regression analysis involving a scalar response variable can be found in Ferraty and Vieu (2006).

The concept of Gaussian process has been exploited by many researchers in various ways for decades. Gaussian process regression (GPR) and related methods have also been used in numerous applications for many years, for example, in spatial statistics under the name of "kriging," and also in machine learning as one type of "kernel machines." However, as pointed out by Neal (1999), "*Despite this past usage, and despite the fundamental simplicity of the idea, Gaussian process models appear to have been little appreciated by most Bayesians. I speculate that this could be partly due to a confusion between the properties one expects of the true function being modelled and those of the best predictor for this unknown function*." Based on our experience of several years of research, we think that many flexible models built based on Gaussian processes provide efficient ways of interpreting model structure, of model learning, and of carrying out inference—particularly in dealing with large dimensional functional data. The developments in this area over the last decade have built up a good theoretical and practical framework. This book attempts to introduce these commendable features of Gaussian process regression models and apply them to functional regression analysis for functional data. This book is therefore naturally formed into two parts: the first part (Chapters 2 to 4) will cover basics of GPR while the second part (Chapters 5 to 8) will cover various advanced topics on functional regression analysis.

After giving a brief description of various statistical concepts related to

functional regression analysis in Chapter 1, we focus on the problems of Gaussian process regression in the following three chapters. The basic idea as well as the technical details on Gaussian process regression have been discussed in many references; among them the monograph of Rasmussen and Williams (2006) has provided an excellent overview from the machine learning context. In this book, we highlight functional data analysis using Gaussian process regression, and present further aspects of Gaussian process regression methods that have not been well covered by other monographs. Specifically, we explore theoretical aspects of Gaussian process regression based on its asymptotic properties in terms of consistency theory in Chapter 2. We also provide new methodological developments of Gaussian process regression particularly for high dimensional data and variable selection in Chapters 3 and 4. The remainder of the book is devoted to the presentation of new nonparametric statistical methods for functional regression analysis, including Gaussian process functional regression (GPFR) models, mixture GPFR models, and generalized GPFR models. These methods ranged over various topics in functional data analysis, including curve prediction, curve clustering, functional ANOVA, and functional regression analysis for batch data or repeated curves and for non-Gaussian data.

The philosophy of the methods discussed in this book is mainly based on Bayesian thinking; the basic idea of the Gaussian process regression model is to assume a nonlinear model as a function in a functional space and a Gaussian process prior is assumed for the function. However, some likelihood-based techniques have also been adopted. For example, the idea behind empirical Bayesian learning is to select values of hyper-parameters by maximizing a marginal likelihood. These kinds of methods are sometimes implemented more efficiently than a fully Bayesian approach. This book aims to provide useful solutions for practitioners as well as to provide useful discussions of theory for researchers and postgraduate students—we usually provide both versions of the methods. Most methods are implemented in MATLAB and C languages and some of them are available on the website *http://www.staff.ncl.ac.uk/j.q.shi/*.

This book has greatly benefited from discussions and collaborations with many colleagues in different disciplines. We would like to use this opportunity to express our gratitude to various people, including Tao Chen, Roman Kamnik, Sik-Yum Lee, Jialiang Li, Julian Morris, Roderick Murray-Smith, Mark J. Schervish, Javier Serradilla, D. Mike Titterington, Bo Wang, Robert M. West, Gang Yi, and Jie Zhang. JQS also wishes to acknowledge the program *Statistical Theory and Methods for Complex, High-Dimensional Data* held in the Newton Institute at Cambridge in early 2008. He has benefited from various inspiring discussions with other participants, in particular, D. L. Bank, R. D. Cook, J. T. Kent, N. D. Lawrence, B. Nan, M. Pontil, M. W. Seeger, D. M. Titterington, and H. Zhou. TC is most grateful to his PhD advisor, Prof. Mark J. Schervish, for his profound guidance and insights.

TC's work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF, 2010-0015629) funded by the Ministry of Education, Science and Technology. We thank Dr. R. Kamnik and Prof. A. Kralj for allowing us to use Paraplegia data and Prof. R. M. West and Dr. E. J. Will for allowing us to use Renal data in this book, and thanks also goes to S. Shi for making great efforts in proofreading the manuscript. Finally, we would like to show our deep appreciation to our families for their support, help, and understanding while the book was being written, and Shan, Neil, Junhwan, and Eunseo for the joy they bring to their Daddies' lives.

Jian Qing Shi and Taeryon Choi
Newcastle Upon Tyne, United Kingdom and Seoul, Korea
April 2011

This page intentionally left blank

List of Abbreviations and Symbols

CV	cross-validation
$D_{\boldsymbol{\theta}}$	first derivative in terms of $\boldsymbol{\theta}$ (differentiation operator)
$D^2_{\boldsymbol{\theta}}$	second derivative matrix in terms of $\boldsymbol{\theta}$ (Hessian matrix)
$\|\|f\|\|_{\boldsymbol{k}}$	RKHS norm; see (2.17)
GCV	generalized cross-validation
GP	Gaussian process, denoted by $GP(\mu, k(\boldsymbol{\theta}))$; see (2.1)
GPBR	GP binary regression
GPFR	Gaussian process functional regression, denoted by $GPFR[\mu, k(\boldsymbol{\theta})\|\boldsymbol{x}(t), \boldsymbol{u}]$; see (5.8)
GPLV	GP latent variable
GPR	Gaussian process regression, denoted by $GPR[\mu, k(\boldsymbol{\theta})\|\boldsymbol{x}(t)]$; see (1.11)
i.i.d.	independent and identically distributed
$k(\cdot,\cdot\|\boldsymbol{\theta})$	covariance function/kernel with hyper-parameter $\boldsymbol{\theta}$
KL	Kullback-Leibler divergence, denoted by $D[p\|\|q]$; see (2.16)
LFR	linear functional regression
$\text{logit}(\pi)$	logit function: $\text{logit}(\pi) = \log(\pi/(1-\pi))$
MAP	maximum a posteriori probability
PCA	principal component analysis
$\varphi_n(\cdot;\boldsymbol{\mu},\boldsymbol{\Sigma})$	density function of n-dimensional multivariate normal distribution $N(\boldsymbol{\mu},\boldsymbol{\Sigma})$
$p(\boldsymbol{\theta})$	prior density function of $\boldsymbol{\theta}$
$p(\boldsymbol{\theta}\|\mathcal{D})$	posterior density function of $\boldsymbol{\theta}$ given data $\mathcal{D}$
Π	a prior
$\Pi(\cdot\|Z_1,\ldots,Z_n)$	a sequence of posterior distributions
RKHS	reproducing kernel Hilbert space
RMSE	root mean of squared errors; see (4.22)

This page intentionally left blank

Chapter 1

Introduction

The main goal of this chapter is to introduce what a functional variable is, and how a functional regression model can be defined for the problems connected with a functional response variable. We give a brief overview of different functional regression models although the focus is on introducing the main idea of the Gaussian process regression model. Several examples of functional data with their associated statistical problems are also provided in this chapter.

1.1 Functional regression models

1.1.1 Functional data

We begin the description of *functional data* by discussing an interesting example in biomechanical research (Kamnik et al., 1999, 2005). The actual application in this example concerns data collected during standing-up maneuvers of paraplegic patients. The outputs (*response variables*) are the trajectories of the body center of mass (COM), required for a simulator control system; for details see Section 1.3.1. Let $y(t)$ be the vertical trajectory of the body COM as output, which is recorded along time during the whole period of a standing-up maneuver. It is usually observed at different time points, denoted by $\{y(t_i), i = 1,\ldots,n\}$ or $\{y_i, i = 1,\ldots,n\}$. The $y(t)$ is a curve or a continuous *functional variable* (it will be defined later) as it is a continuous measurement along time t. Figure 1.1 depicts 40 curves of $y(t)$, the vertical trajectories of the body COM, denoted by *comz*. Each curve represents a standing-up maneuver. Specifically, eight patients participated in the experiment, and each of them repeated the experiment five times.

However, it is very difficult to measure body position unless expensive equipment is used and set up in a laboratory environment. Thus, one of the aims of the example is to develop a model for reconstructing the trajectory of the body COM by using some easily measured quantities, such as the forces and torques under the patient's feet, under the arm support, and under the seat while the body is in contact with it. More than 30 such *input variables* are observed, which are denoted by $x_q(t)$ for $q = 1,\ldots,Q$. Each of them takes up

different values along time during the period of standing-up maneuvers, which is also a functional variable. Bear in mind, the response variable $y(t)$ may also depend on some scalar variables such as the height and weight of the subject.

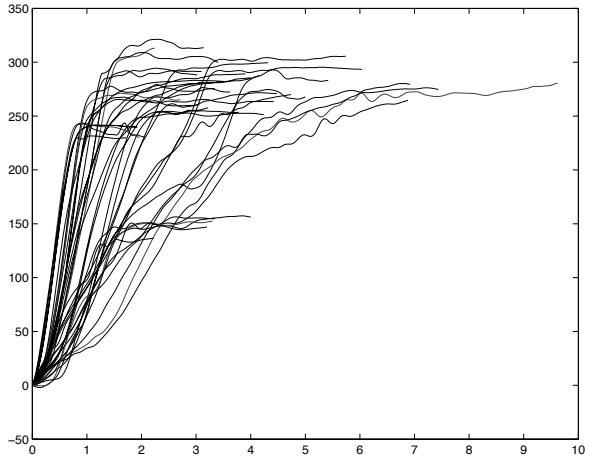


Figure 1.1 *Paraplegia data for eight patients: vertical trajectory of the body COM comz coordinate (Y-axis, in mm) against time (X-axis, in seconds). Each curve corresponds to one standing-up.*

The aim of functional regression analysis is to find a *model* f to explain and predict $y(t)$ given functional covariates $\boldsymbol{x}(t) = (x_1(t), \ldots, x_Q(t))^T$ and scalar variables $\boldsymbol{u} = (u_1, \ldots, u_p)^T$:

$$y(t) = f(t, \boldsymbol{x}(t), \boldsymbol{u}) + \varepsilon(t). \tag{1.1}$$

Throughout the remainder of the book, we need to distinguish between a functional variable and a scalar variable.

A *functional variable* $y(t)$—also called a *curve* (the term we usually used when $y(t)$ is continuous)—such as the vertical trajectory of the body COM discussed above, denotes a variable associated with t, i.e., a function of t (t could be time or some other temporal or spatial variable and it could therefore be multidimensional, although, in most cases, we will limit our discussion to a univariate t). More formally, a variable is called a functional variable if it takes values in an infinite dimensional space (Ferraty and Vieu, 2006). Observations of functional variables are called *functional data*, such as $\{y(t_i), i = 1, \ldots, n\}$ or $\{y_i, i = 1, \ldots, n\}$. Functional input variables are usually observed at the same time points as the response variable $y(t)$, denoted by $\{x_q(t_i), i = 1, \ldots, n, q = 1, \ldots, Q\}$ or $\{x_{qi}, i = 1, \ldots, n, q = 1, \ldots, Q\}$, although, in theory, each of these functional variables can be observed at different time points; see more discussion in Section 1.4.

A *scalar variable* denotes an ordinary variable, for example, the height of a subject, which takes a scalar value during the whole period of the standing-up maneuver.

1.1.2 Functional regression analysis

Classified according to the type of response variable, there are mainly two types of functional regression models. One involves the functional response variable and the other involves the scalar response variable; both include mixed covariates—scalar and functional variables. The former is discussed in this book, and the latter is considered in Ferraty and Vieu (2006). Further reading and notes will be given in Section 1.4.

We focus now on regression problems involving functional responses with mixed scalar input variables and functional input variables. For this purpose we first look at the basic structure and the notation of the functional data which can be used for our description of functional regression analysis. In the previous section, we used $y(t)$ to represent a single response variable. Let's now assume that we are given repeated data with M replications. We then denote $y_m(t)$ as the response variable corresponding to the m-th replication for $m = 1,\ldots,M$, and the related functional input variables are denoted by $\{x_{mq}(t), q = 1,\ldots,Q\}$. We denote all the scalar input variables by a p-dimensional vector $\boldsymbol{u}$. The data collected from the M replications or the M subjects are called *batch data* — which is the terminology popular in the engineering community. In the m-th batch, suppose that n_m observations are recorded, then the data collected in the m-th batch are:

$$\mathcal{D}_m = \{(y_{mi}, t_{mi}, x_{m,1i}, \ldots, x_{m,Qi}) \text{ for } i = 1,\ldots,n_m; \text{ and } (u_{m1},\ldots,u_{mp})\} \quad (1.2)$$

where t_{mi} is the time point at which we recorded the data, $y_{mi} = y(t_{mi})$ is the output recorded at t_{mi}, and $x_{m,qi} = x_q(t_{mi})$ is the measurement of the q-th input variable for $q = 1,\ldots,Q$. The elements of $\boldsymbol{u}_m = (u_{m1},\ldots,u_{mp})^T$ are the observations of scalar variables. Note that they are not functional variables. These variables are not dependent on t but they offer information for each batch. In the Paraplegia example, the experiment has been repeated 40 times, and thus we are given a set of batch data with 40 batches. The $\mathcal{D}_m$ denotes all the data recorded in the m-th standing-up.

There is a wider range of ways to model a functional response variable. We first consider how to explain $y(t)$ by using the scalar input (explanatory) variables $\boldsymbol{u}$. For instance, a linear functional regression model (Ramsay and Silverman, 2005) is defined as follows:

$$y_m(t) = \mu_m(t) + \varepsilon_m(t) \text{ and } \mu_m(t) = \boldsymbol{u}_m^T \boldsymbol{\beta}(t) \text{ for } m = 1,\ldots,M, \quad (1.3)$$

where $\boldsymbol{\beta}(t)$ is a vector of functional coefficients, which can be estimated by

suitable statistical methods such as the B-spline smoothing. This will be discussed in Chapter 5. Note that in model (1.3), $\mu_m(t)$ explains the mean structure of $y_m(t)$, and the random errors $\varepsilon_m(t)$ are assumed to be independent. There is a large collection of literature on fitting the mean structure $\mu_m(t)$ by a nonparametric approach. See more discussion and references in Sections 1.2 and 1.4.

Since the functional responses at different data points in the same batch are often dependent, we may need to further consider a covariance structure. Rice and Silverman (1991) defined the following stochastic process model which can model both mean and covariance structures:

$$y_m(t) = \mu_m(t) + \tau_m(t) + \varepsilon_m(t), \quad \text{for } m = 1, \ldots, M, \tag{1.4}$$

where $\varepsilon_m(t)$ are still independent random errors, and the mean structure $\mu_m(t) = E(y_m(t))$ can be modeled by a linear functional regression model as in (1.3) or other parametric or nonparametric models. The term $\tau_m(t)$ is a stochastic process with zero mean and covariance function $k(t,t') = \mathrm{Cov}(\tau(t), \tau(t'))$. In model (1.4), $\mu_m(t)$ can be treated as a common structure throughout all the M batches, which is estimated by the data collected from all batches. On the other hand, the dependence structure or the covariance structure is considered by incorporating it with an appropriate stochastic process. This part can be treated as a special structure for each batch and it needs to be estimated based on the data collected from the batch. This is an important feature of the model (1.4), compared to (1.3). Based on this model, if we have collected enough data from different batches, then under certain regularity conditions, the estimation of $\mu_m(t)$ is a consistent estimation of the true common mean structure; if we could collect enough data from a particular batch, the estimation or prediction based on both parts in (1.4) would also be consistent in estimating the true curve that explains $y_m(t)$. These consistent results are not only theoretically important but also very useful in applications. One example is to use it to construct an individual dose-response curve, hence the *patient-specific treatment regime* in biomedical research. The details of these results are discussed in Chapter 5.

Note that the stochastic process described in (1.4) depends on a one-dimensional functional covariate t. Shi et al. (2007) extended the idea to deal with a functional regression problem involving multidimensional functional covariates $\boldsymbol{x}(t) = (x_1(t), \ldots, x_Q(t))^T$ (Q is usually quite large):

$$y_m(\boldsymbol{x}) = \mu_m(t) + \tau_m(\boldsymbol{x}) + \varepsilon_m(t). \tag{1.5}$$

In this case, the mean structure can also be modeled by a linear functional regression model as defined in (1.3) or other models, but the covariance structure is modeled by a Gaussian process regression model,

$$\mu_m(t) = \boldsymbol{u}_m^T \beta(t) \ \text{ and } \ \tau_m(\boldsymbol{x}) \sim GPR[0, k(\boldsymbol{\theta})|\boldsymbol{x}]. \tag{1.6}$$

In other words, the mean structure $\mu_m(t)$ depends on scalar covariates $\boldsymbol{u}_m = (u_{m1},\ldots,u_{mp})^T$, while the regression relationship between the functional response $y(t)$ and the functional covariates $\boldsymbol{x}(t)$ is modeled by a nonparametric Gaussian process regression model $GPR[0,k(\boldsymbol{\theta})|\boldsymbol{x}]$, which is defined in (1.11). The basic idea and the definition of the Gaussian process regression model are discussed in the next section; basic theory and method for the implementation are discussed in Chapters 2 and 3. The Gaussian process functional regression model (1.5) is discussed in Chapter 5.

In model (1.5), we focus on the functional data (sometimes also called *multivariate time series data*) as defined in (1.2). We use a notation of either $y(t)$ or $y(\boldsymbol{x}(t))$ or $y(\boldsymbol{x})$ for the response variable. However, if we are given a set of spatial data

$$\mathcal{D}_m = \{(y_{mi},\boldsymbol{x}_{mi}), i = 1,\ldots,n_m, \boldsymbol{x} \in \mathcal{R}^Q\},$$

model (1.5) is re-expressed as

$$y_m(\boldsymbol{x}) = \mu_m(\boldsymbol{x}) + \tau_m(\boldsymbol{x}) + \varepsilon_m(\boldsymbol{x}). \tag{1.7}$$

Since inferences for two models are almost the same, we will ignore the slight difference between models (1.5) and (1.7) in the remainder of the book.

1.2 Gaussian process regression

For a functional response (a curve) $y(\boldsymbol{x})$ and a set of functional covariates $\boldsymbol{x}(t) = (x_1(t),\ldots,x_Q(t))^T$, consider a nonlinear functional regression model as follows:

$$y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon. \tag{1.8}$$

The unknown nonlinear function is a mapping $f(\cdot) : \mathcal{R}^Q \to \mathcal{R}$. When Q is small, say $Q = 1$, many nonparametric regression techniques can be used to estimate the unknown function f. A local polynomial model is one example, which is defined as

$$f(x) = \sum_{k=1}^{K} \alpha_k(x) x^k.$$

The coefficient $\alpha_k(x)$ can be estimated by weighted least squares theory via a kernel function (see, e.g., Wand and Jones, 1995; Fan and Gijbels, 1996). Another example is the model using spline smoothing. It is defined as (see, e.g., Wahba, 1990; Green and Silverman, 1994; Gu, 2002):

$$f(x) = \sum_{k=1}^{K} \alpha_k \Phi_k(x), \tag{1.9}$$

where $\{\Phi_k, k = 1,\ldots,...\}$ are the spline basis functions. However, most of these nonparametric regression models suffer from the *curse of dimensionality* when

they are applied to multidimensional covariates. A variety of alternative approaches have been proposed to overcome this problem. Examples include the additive model (Hastie and Tibshirani, 1990), the projection pursuit regression (Friedman and Stuetzle, 1981), the sliced inverse regression (Duan and Li, 1991; Li, 1991), the neural network model (see, e.g., Cheng and Titterington, 1994), and the varying-coefficient model (see, e.g., Fan and Zhang, 2000; Fan et al., 2003); more discussion can be found in Green and Silverman (1994), Wand and Jones (1995), Banks et al. (1999), and Ruppert et al. (2003).

Most of the methods mentioned above are developed from a frequentist perspective. On the other hand, the Bayesian version of their counterparts and other novel Bayesian approaches have also been studied; see, for example, Denison et al. (1998) and references therein. Another approach for Bayesian nonparametric regression analysis is based on a Gaussian process prior. This approach treats the unknown function $f(\cdot)$ as a random function, and then defines a Gaussian process, with mean function $\mu(\cdot)$ and covariance function $k(\cdot,\cdot)$, as a prior distribution for $f(\cdot)$. The covariance function or covariance kernel, $k(\boldsymbol{x},\boldsymbol{x}')$, depends on the Q-dimensional input variables $\boldsymbol{x}$ and $\boldsymbol{x}'$. Specifically, for any data points $\boldsymbol{x},\boldsymbol{x}' \in \mathcal{R}^Q$, we assume that $(f(\boldsymbol{x}), f(\boldsymbol{x}'))$ have a normal distribution with mean $(\mu(\boldsymbol{x}),\mu(\boldsymbol{x}'))$. The covariance is defined by the kernel function as

$$\mathrm{Cov}(f(\boldsymbol{x}), f(\boldsymbol{x}')) = k(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}), \tag{1.10}$$

where $\boldsymbol{\theta}$ are hyper-parameters. The details of dealing with hyper-parameters are discussed in Chapter 4. We call this nonparametric regression approach a Gaussian process regression (GPR) model, denoted by

$$f(\boldsymbol{x}) \sim GPR[\mu(\cdot), k(\boldsymbol{\theta})|\boldsymbol{x}]. \tag{1.11}$$

For convenience, sometimes we refer to both (1.8) and (1.11) as a Gaussian process regression model.

The Gaussian process $f(\boldsymbol{x})$ can be decomposed, according to the Karhunen-Loève orthogonal expansion (see, e.g., Wahba, 1990), as

$$f(\boldsymbol{x}) = \sum_{i=1}^{\infty} \phi_i(\boldsymbol{x})\xi_i, \tag{1.12}$$

where ξ_i's are independent distributed random variables having $N(0,\lambda_i)$, and the covariance kernel $k(\boldsymbol{x},\boldsymbol{x}')$ can be expanded, also known as Mercer's theorem (see, e.g., Adler and Taylor, 2007), into

$$k(\boldsymbol{x},\boldsymbol{x}') = \sum_{i=1}^{\infty} \lambda_i \phi_i(\boldsymbol{x})\phi_i(\boldsymbol{x}'), \tag{1.13}$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ denote the eigenvalues and $\phi_1, \phi_2, \ldots$ are the related

eigenfunctions of the operator whose kernel is $k(\boldsymbol{x},\boldsymbol{x}')$, so that

$$\int k(\boldsymbol{x}',\boldsymbol{x})\phi_i(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \lambda_i\phi_i(\boldsymbol{x}'), \tag{1.14}$$

where $p(\boldsymbol{x})$ is the density function of the input vector $\boldsymbol{x}$. The eigenfunctions are p-orthogonal, i.e.,

$$\int \phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \delta_{ij},$$

where δ_{ij} is the Kronecker delta, which is 1 if $i = j$ and 0 otherwise. In (1.12) ξ_i is given by

$$\xi_i = \int \phi_i(\boldsymbol{x})y(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}. \tag{1.15}$$

Based on (1.12), the response curve $y(t)$ can be expanded in an infinite-dimensional space through a Gaussian process prior as defined in (1.11).

Comparing (1.12) with (1.9), we see they are quite similar. The difference is that the former has random coefficients while the latter has fixed coefficients.

In practice, however, it is not always possible to obtain the explicit form of the eigenvalue decomposition of the covariance function, except for certain covariance kernels such as a Brownian motion, since it involves solving an integral equation based on Mercer's theorem (Adler and Taylor, 2007). There are usually no analytic solutions existing.

In addition, when the covariance function is completely unknown, and thus needs to be estimated, the problems become more complicated. The general theory has yet to develop. In the special case of a one-dimensional functional covariate (i.e., $Q = 1$), Yao et al. (2005a,b) used weighted least squares with local quadratic and linear components to obtain an estimate of $k(\cdot,\cdot)$ in (1.10); they then estimated the eigenfunctions and eigenvalues in (1.14). The nonlinear regression function $f(\boldsymbol{x})$ can therefore be estimated by (1.12). However, it is difficult to extend their method to the regression problems with multi-dimensional covariates.

Alternatively, in many applications, one uses a particular form of covariance function (like squared exponential) in order to model desired forms of dependence between functional responses. Thus, the first step of Gaussian process regression analysis (O'Hagan, 1978; Rasmussen, 1996) is to select a covariance kernel $k(\cdot,\cdot;\boldsymbol{\theta})$, where we assume that the covariance function depends on $\boldsymbol{\theta}$—a set of hyper-parameters. If we have obtained a finite number of observations, denoted by

$$\mathcal{D} = \{\boldsymbol{y},\boldsymbol{X}\} = \{(y_i,x_{i1},\ldots,x_{iQ}), i = 1,2,\ldots,n\}. \tag{1.16}$$

We can then use this set of data to estimate the nonlinear function $f(\cdot)$ in the GPR model (1.8) and (1.11) (the details of how to estimate will be discussed in Chapters 2 and 3).

Some advantages of this approach are:

(1) The prior specification of covariance function enables us to accommodate a wide class of nonlinear regression functions of $f(\cdot)$, and prior knowledge about the regression function can be incorporated. The hyper-parameters involved in the covariance function play a role similar to the smoothing parameters in spline models. They can be fixed (thus similar to imposing a fixed roughness penalty) or can be estimated based on the data by using, for example, a generalized cross-validation or an empirical Bayesian approach. The detailed discussion will be given in Chapters 3 and 4.

(2) The GPR model can be easily applied to address regression problems with multidimensional functional covariates.

(3) The GPR model provides a natural framework on modeling a nonlinear regression function and a covariance structure simultaneously. The latter is particularly useful when it is incorporated with a mean structure into a combined model to analyze batch data; see, for example, the Gaussian process functional regression model that is discussed in Chapter 5.

However, as pointed out by Williams and Seeger (2001), estimating $f(\cdot)$ using a data set of size n is equivalent to approximating $f(\cdot)$ in (1.12) using the first n eigenfunctions. So, essentially, a GPR model approximates the model structure defined in an infinite-dimensional space by a function defined in a finite-dimensional space. The approximation will become better when the sample size tends to infinity.

Gaussian process regression models have been used in many applications, and essentially the same model has long been used in spatial statistics under the name of *kriging* (see, e.g., Matheron, 1973; Journel and Huijbregts, 1978; Daley, 1991; Laslett, 1994; Stein, 1999; Diggle et al., 1998, 2003). Gaussian processes have been used successfully for regression and classification, particularly in machine learning. Neal (1996) has shown that many Bayesian regression models based on neural networks converge to Gaussian processes in the limit as the number of nodes becomes infinite. This has motivated many to look into the use of Gaussian process models for the high-dimensional applications to which neural networks are typically applied (Rasmussen, 1996). Readers can refer to the Gaussian processes website via the link *http://www.gaussianprocess.org/* for an overview of resources concerned with Gaussian processes modeling, including several good references for Gaussian process regression modeling. In particular, the monograph by Rasmussen and Williams (2006) would be useful for grasping the basic idea as well as the technical details on Gaussian process regression from the machine learning context.

Compared to those references in Gaussian process regression, in this book we highlight functional data analysis using Gaussian process regression models, and present further aspects of Gaussian process regression methods that have not been well discussed in the literature. Specifically, we explore theo-

retical aspects of Gaussian process regression based on its asymptotic properties in terms of the consistency theory in Chapter 2. We also provide new methodological developments of Gaussian process regression particularly for high dimensional data and variable selection in Chapters 3 and 4 before we move toward the later part of the book to discuss Gaussian process functional regression analysis for batch data or repeated curves.

1.3 Some data sets and associated statistical problems

In this section, we introduce data sets and the statistical problems associated with them. In particular, two data sets are discussed in detail in this section; other data sets will be introduced in later chapters as they are needed. Basically, these data sets are used to illustrate what functional data are, what statistical problems are in relation to the functional data, and what we aim to achieve in functional regression analysis.

1.3.1 Modeling of standing-up maneuvers

The first data set is the one that was introduced in Section 1.1. The application concerns Functional Electrical Stimulation (FES)-assisted standing-up maneuvers performed by paraplegic patients. In this biomechanical application, patients are supposed to stand up with the help of an arm support along with electrical stimulation of their paralyzed lower extremities. The FES artificially invokes muscle contractions and thus creates torques in the body joints. In the case of standing up, the knee joint extensor muscles and the quadriceps group are stimulated by two surface electrodes on each leg. In the experiments, the stimulation level was constant and was triggered by the user via push buttons; for more details see Kamnik et al. (1999). The stimulation sequences were determined on the basis of known subject body positions and arm reactions. Using Goniometers, accelerometers, other sensors, and the related algorithms, one can arrange for the body position and other information to be fed back to the simulator control system. However, the equipment is very expensive and it is a tedious job to set the sensors. This method can therefore only be used in a simulation or laboratory environment; it is not suitable for implementation in home or clinical praxis. For this reason, the supportive forces acting at the interaction points with the paraplegic's environment are considered as an alternative feedback source; for more details see Kamnik et al. (2005). To use the supportive force feedback information, one will need a model that relates the supportive forces to the output trajectory.

The output (response variable) $y_m(t)$, for the m-th replication (or batch), includes the horizontal (comy) or vertical (comz) trajectories of the body's COM (center of mass). The input variables (covariates), $\boldsymbol{x}_m(t)$, include the forces and torques under the patient's feet, under the arm support handle, and under the

seat while the body is in contact with it, and other variables. There are in total 33 functional input variables. In one standing-up, the outputs and inputs are recorded for a few hundred time points. The experiment was repeated five times for one patient, and there are a total of eight patients involved in this project.

Figure 1.1 presents the functional response variable $y_m(t)$ for $m = 1, \ldots, 40$, i.e., 40 replications. Each corresponds to one standing-up. The curves of one input variable are shown in Figure 1.2. Bear in mind that the data for most of the input variables are quite noisy since they are recorded from real sensors during a standing-up maneuver; robust statistical methods are therefore needed.

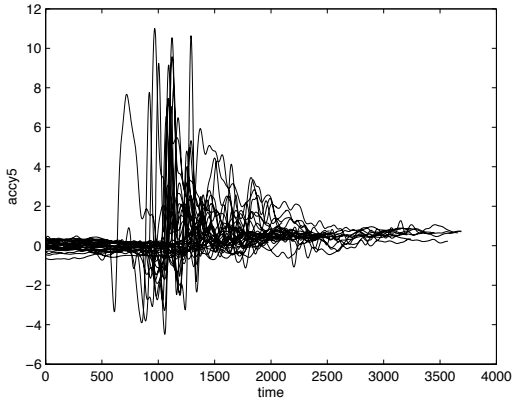


Figure 1.2 *Paraplegia data for eight patients: one of the functional input variables against time (X-axis, in seconds). Each curve corresponds to one standing-up.*

The main goal of these experiments is to find a regression model,

$$y_m(t) = f(\boldsymbol{x}_m(t), \boldsymbol{u}_m) + \varepsilon_m(t),$$

where $\boldsymbol{u}_m$ is a vector of scalar variables, offering information such as weight and height of the patient. We then apply the regression model to predict the values of the functional response variable $y(t)$ using the new recorded data of input variables in a new standing-up maneuver for the same patient or for a new patient. Since we have little information about the physical relationship between the output and input variables, it is therefore not realistic to use any parametric models for these data. As we will explain later in this book, the Gaussian process functional regression model, as a nonparametric approach, is an appropriate choice for fitting these data.

There are two major problems when the GPFR model is applied to this data set. One is the heterogeneity. The regression relationship between the func-

tional response variables and the input variables might be different for the different replication of the standing-up maneuver due to the differences in height, weight, and other factors for the different patients. The other problem is caused by the large number of input variables. Such an issue with high dimensional input variables often leads to computational instability, for example due to a singular Hessian matrix. Those problems are addressed and some solutions are provided in Chapters 4, 5, and 6.

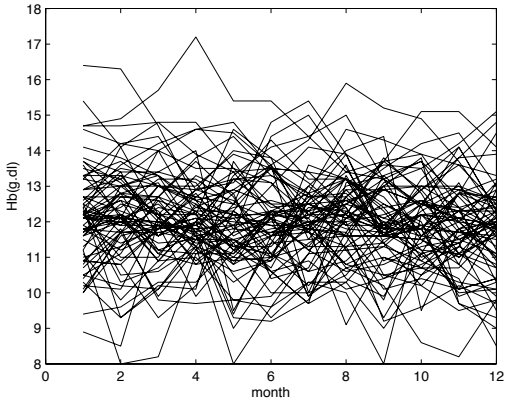
We will refer to this example as the *Paraplegia example* and the data as *Paraplegia data* in the remainder of this book.

1.3.2 Individual dose-response curve

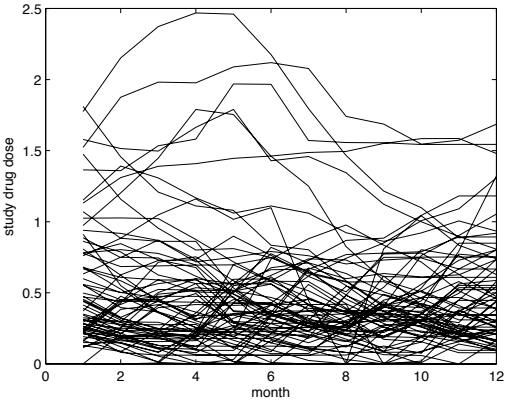
The second example is to assess the control of hemoglobin (Hb) levels in patients with kidney disease, each dosed with one of two epoetin agents. Patients with reduced kidney function not only require dialysis to remove waste products from their blood, but they also produce less erythropoetin (EPO)—the natural stimulus to the production of red cells in the bone marrow. As a consequence most dialysis patients suffer renal anemia to some degree. This can be treated subcutaneously or intravenously with either a synthetic EPO—for example, Erythropoetin Beta (EB)—or a modified epoetin such as Darbepoetin Alpha (DA). The dose of epoetin to be given to each patient is determined by monitoring the Hb concentration from a blood sample taken typically every two or four weeks (West et al., 2007; Will et al., 2007). Patients need to have their Hb levels controlled within relatively narrow limits around the level of 11.8 (Volkova and Arab, 2006). If Hb levels are too low then patients begin to show the symptoms of anemia, and if too high then there may be prothrombotic risks to their dialysis treatment and vascular tree. The primary therapeutic concern is how to maintain the Hb level by giving a suitable level of dose of epoetin for each patient.

In this example, the response is the Hb level for a patient, shown as curves in Figure 1.3(a). One of the covariates is the dose level of epoetin, which is also shown as a curve for each of the patients in Figure 1.3(b). The project studies the question of how to reveal the model structure from the data collected from all subjects and also how to capture the correlation structure from the data collected from each individual subject. This can be linked to the common mean structure and the covariance structure for each individual in (1.5). This is particularly useful in dose-response studies since we can then construct dose-response curves for each individual patient, enabling the planning of a *patient-specific treatment regime*. The details are discussed in Chapter 5.

We will refer to this example as *Renal example* and the data as *Renal data* in the remainder of the book.



(a)



(b)

Figure 1.3 *Renal data: (a) the Hb level, and (b) the dose level for all the patients. Each curve corresponds to one patient.*

1.4 Further reading and notes

Ramsay and Silverman (2005) is certainly a must-read book in functional data analysis, offering tools for exploring functional data with different models. Functional regression analysis has also been discussed in their book although it is mainly based on linear functional models.

As discussed in Section 1.1.2, there are mainly two types of functional regression models classified according to the category of response variables, although they can be subclassified according to the types of covariates. The func-

tional regression analysis for the scalar response variable with scalar and/or functional covariates has been discussed in Chapter 15 of Ramsay and Silverman (2005) using a functional linear model. Some related references can be found there. Ferraty and Vieu (2006) discussed this problem nonparametrically. Some other developments and the statistical properties can be found in, for example, Cardot et al. (1999), Cai and Hall (2006), Cardot et al. (2006), Delaigle et al. (2009), and the references therein.

This book focuses on functional regression models for functional response variables with mixed scalar and functional covariates. The discussion of linear functional regression models along with their relevant practical aspects can be found in Ramsay and Dalzell (1991), Faraway (1997), Ramsay and Silverman (2005), Yao et al. (2005a), Ramsay et al. (2009), and references therein. The relevant statistical properties can be found in Yao et al. (2005b). We focus our attention on a Gaussian process functional regression model, aiming to reveal both common mean structure and individual covariance structure for the regression problem. The latter will be modeled by a Gaussian process regression model as discussed in Section 1.2.

In functional data analysis, a preliminary step of exploratory data analysis to raw observations, may be essential. Some techniques such as curve registration or dynamic time warping can be used. The readers are referred to Ramsay and Li (1998) and Ramsay and Silverman (2005).

This page intentionally left blank

Chapter 2

Bayesian nonlinear regression with Gaussian process priors

In this chapter, we discuss the details of the idea on using Gaussian process priors for a Bayesian nonlinear regression model, i.e., the so-called Gaussian process regression model, and describe the basic theory of this model. First, in Section 2.1.1, we illustrate how the Gaussian process can be used as a prior distribution for the unknown regression function by associating the concept of the stochastic process with a random function. We then demonstrate how to implement Gaussian process regression when a covariance function is given in Section 2.1.2. Finally, we survey asymptotic results of Gaussian process regression methods, mainly in terms of consistency. Specifically, we provide basic concepts of posterior consistency in Section 2.2, and discuss several consistency results in the Gaussian process regression method in Section 2.3. In these regards, this chapter addresses some of the technical features in Gaussian process regression, in particular asymptotic properties in terms of consistency.

2.1 Gaussian process prior and posterior

2.1.1 Gaussian process prior

A Bayesian nonlinear regression model with Gaussian process prior, referred to as Gaussian process regression (GPR) model, can simply be thought of as an ordinary Bayesian regression with an infinite dimensional parameter space of unknown nonlinear regression functions. Thus, the unknown nonlinear regression function is regarded as the unknown parameter, i.e., the random function needs to be estimated, and its prior distribution also needs to be specified. That is, we need to consider specifying probability distributions for a random function and utilize a stochastic process, in particular the Gaussian process, for this purpose. We now first take a look at the basic concept of a stochastic process.

A stochastic process is a collection of random variables defined on the same probability space $(\Omega, \mathcal{F}, P)$. The fundamentals of stochastic processes can be found in standard probability theory books such as Billingsley (1995)

and Breiman (1968). If we denote the index of the collection by t from an index set T, the stochastic process can be written as $\{Y_t(\omega) : t \in T\}$. When we fix t, $Y_t(\omega)$ is nothing but a random variable, and the stochastic process can be thought of as a family of random variables indexed by $t \in T$. On the other hand, a stochastic process can be interpreted as a single random variable taking values in an infinite-dimensional function space by fixing ω. In this case, the random function $t \to Y_t(\omega)$ is a realization of stochastic process over $t \in T$. Therefore, it is natural to describe the prior probability distribution for the random regression function as a stochastic process.

Note that this concept of a stochastic process can be generalized into a multidimensional index set. For example, for spatial data, we need to define a stochastic process with respect to location in spatial space. This can be easily achieved by extending the above definition to $\{Y(\boldsymbol{x}), \boldsymbol{x} \in T \subset \mathcal{R}^Q\}$.

In particular, a Gaussian process is a stochastic process parametrized by its mean function

$$\mu(\cdot) : T \to \mathcal{R}, \ \mu(\boldsymbol{x}) = \mathrm{E}[Y(\boldsymbol{x})],$$

and its covariance function

$$k(\cdot,\cdot) : T^2 \to \mathcal{R}, \ k(\boldsymbol{x},\boldsymbol{x}') = \mathrm{Cov}(Y(\boldsymbol{x}), Y(\boldsymbol{x}'))$$

which we denote $GP(\mu, k)$. According to Kolmogorov's existence theorem (Billingsley, 1995), a stochastic process can be characterized by finite-dimensional distributions. To say that

$$Y(\cdot) \sim GP(\mu(\cdot), k(\cdot,\cdot)) \tag{2.1}$$

means that for any n and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in T$, the joint distribution of $\boldsymbol{Y}_n = (Y(\boldsymbol{x}_1), \ldots, Y(\boldsymbol{x}_n))^T$ is an n-variate normal distribution with mean vector $\boldsymbol{\mu}_n = (\mu(\boldsymbol{x}_1), \ldots, \mu(\boldsymbol{x}_n))^T$ and covariance matrix $\boldsymbol{\Psi}_n$ whose (i,j) entry is $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $i, j = 1, \ldots, n$. A Gaussian process can be viewed from two different points of view, one from the stochastic process and the other from the extension of multivariate normal random variables. From the second perspective, a Gaussian process can be taken as an infinite-dimensional generalization of multivariate normal random variables to infinite index sets. In other words, we can say that a stochastic process is Gaussian if every finite-dimensional joint distribution is multivariate normal and the covariance function of the Gaussian process is an extension of the covariance matrix of multivariate normal random variables. In this regard, as we put a multivariate normal distribution for a prior distribution on a finite set of parameters, so a natural extension will put a Gaussian process prior for a random function from infinite-dimensional parameter spaces. Accordingly, Gaussian processes are a natural way of defining prior distributions over spaces of functions, which are the parameter spaces for Bayesian nonlinear regression models.

One of the key aspects of Gaussian processes is the covariance function

$k(\cdot,\cdot)$, which is indispensable when it comes to studying Gaussian process characteristics, and it also determines their smoothness properties such as the sample path continuity (i.e., the continuity of $\{Y_t(\omega) : t \in T\}$) and its differentiability. The intrinsic property of the covariance function is that it is a non-negative definite function, defined as follows. For all n, every $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in T$ and $a_1, \ldots, a_n \in R$

$$\sum_{i,j=1}^{n} a_i a_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0. \tag{2.2}$$

In Gaussian process regression modeling, there are widely used covariance function choices and we introduce some of them in Section 4.1. In particular, we need to choose a Gaussian process with a suitable covariance function as a prior distribution for the regression function $f(\boldsymbol{x})$ to achieve posterior consistency. That is discussed in the next section. Appropriate smoothness assumptions on the covariance function are required for this purpose. In estimating the unknown regression function, the covariance function also plays a central role in determining the property of sample paths of the Gaussian process and controlling the smoothness of data. In addition, proving posterior consistency results in the next section, the choice of the covariance function plays an important part in guaranteeing the existence of sample paths for the Gaussian process with a suitable smoothness condition that we will assume. Analytical properties of the sample functions of Gaussian processes have been carefully studied and reviewed in Cramer and Leadbetter (1967), Adler (1990), and Abrahamsen (1997); see also the monographs by Rue and Held (2005), Adler and Taylor (2007), and references therein for further details about theory and applications of Gaussian processes.

In practice, one can use a particular form of covariance function (for example, the squared exponential covariance kernel) in order to model desired forms of dependence as mentioned in the previous chapter. Any form of covariance function can be used in principle, provided it is a non-negative definite function. Thus, it is a little hard to have an explicit rule for how to choose a covariance function in modeling, but in many cases, we may first select a covariance function based on the shape and the smoothness of the sample path. Once the covariance function is selected, we need to select the values of smoothing parameters involved in the covariance function. One method is to select the values of these smoothing parameters based on the observed data and the model. In such a case, the specific choice of a family of covariance functions is not so critical as long as the family is sufficiently flexible. The details of how to select a proper covariance function and values of smoothing parameters (they are also called hyper-parameters) are to be discussed in Section 4.1 and the following two chapters.

2.1.2 *Bayesian inference*

The problem of statistical inference is to draw meaningful conclusions about unknown parameters. The parameter is unobservable, and thus the inference must be based on the data or observations. Bayesian inference treats all unknown parameters as random variables and uses conditional probability to express all forms of uncertainty. Before the data are observed, this prior distribution quantifies our uncertainty about the unobservable. After we observe data, our opinion or belief is updated through a conditional distribution of the parameter given data, and this conditional distribution is called a posterior distribution. Therefore, based on Gaussian process priors we described in the previous section, we can obtain the posterior distribution of the unknown regression function based on observations.

A Gaussian process regression model has been defined in (1.8) and (1.11). Let us now assume that we have observed a set of data $\mathcal{D} = \{(y_i, \boldsymbol{x}_i), i = 1, \ldots, n,\ \boldsymbol{x}_i \in T \subset \mathcal{R}^Q\}$. A Gaussian process regression model for this data set can be specified as follows:

$$\begin{aligned} y_i &= f(\boldsymbol{x}_i) + \varepsilon_i,\ i = 1, \ldots, n, \\ \varepsilon_i &\sim i.i.d.\ N(0, \sigma^2),\ \sigma^2 \text{ known}, \\ f(\cdot) &\sim GP(\mu(\cdot), k(\cdot,\cdot)) \text{ and } \mathrm{Cov}(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) = k(\boldsymbol{x}_i, \boldsymbol{x}_j). \end{aligned} \tag{2.3}$$

We temporarily assume in this subsection that the noise variance σ^2 is known, and the covariance function $k(\cdot,\cdot)$ is predetermined with fixed hyper-parameters known in advance. It is common to assume a zero mean function in the Gaussian process prior, i.e., $\mu(\cdot) = 0$. Examples of covariance functions are to be discussed in Section 4.1, and we consider a specific example of covariance function in this subsection, *the squared exponential* covariance. This is one of the most commonly used covariance functions in practice, also sometimes called the *Gaussian kernel* or the *Gaussian radial basis function kernel* (see, e.g., Schölkopf and Smola, 2002; Rasmussen and Williams, 2006),

$$k(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}) = \mathrm{Cov}(f(\boldsymbol{x}), f(\boldsymbol{x}')) = v_0 \exp\left\{ -\frac{1}{2} w \sum_{q=1}^{Q} (x_q - x_q')^2 \right\}, \tag{2.4}$$

where $\boldsymbol{\theta} = (v_0, w)$ denotes the set of hyper-parameters assumed to be known in the remainder of this subsection. The problems regarding how to select the values of hyper-parameters will be discussed later.

From the model structure defined in (2.3), the estimation problem becomes a multivariate normal estimation problem, based on n observations in $\mathcal{D}$. Then, the posterior distribution of $\boldsymbol{f} = (f(x_1), f(x_2), \ldots, f(x_n))^T$ can be calculated in the following way. Let $\boldsymbol{K}$ be the $n \times n$ covariance matrix evaluated at all pairs of the n design points, $\boldsymbol{K}(i,j) = \mathrm{Cov}(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. That is, given the vector $\boldsymbol{f}$ and noise variance σ^2, the observed responses $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ have

a multivariate normal distribution with the mean vector, $\boldsymbol{f}$, and the covariance matrix, $\sigma^2\boldsymbol{I}$, where $\boldsymbol{I}$ is the $n \times n$ identity matrix. Also, $\boldsymbol{f}$ has a multivariate normal distribution with zero mean and the covariance matrix, $\boldsymbol{K}$, i.e.,

$$\begin{aligned}\left(y_1,\ldots,y_n|\boldsymbol{f},\sigma^2\right) &\sim N_n(\boldsymbol{f},\sigma^2\boldsymbol{I}),\\ \boldsymbol{f} &\sim N_n(\boldsymbol{0}_n,\boldsymbol{K}).\end{aligned}$$

Thus, the posterior distribution of $\boldsymbol{f}$, $p\left(\boldsymbol{f}|\mathcal{D},\sigma^2\right)$, is proportional to the product of two normal distributions

$$\begin{aligned}p(\boldsymbol{f}|\mathcal{D},\sigma^2) &= \frac{p(\boldsymbol{y}|\boldsymbol{f},\sigma^2)p(\boldsymbol{f})}{\int p(\boldsymbol{y}|\boldsymbol{f},\sigma^2)p(\boldsymbol{f})d\boldsymbol{f}}\\ &\propto \varphi_n(\boldsymbol{y}|\boldsymbol{f},\sigma^2\boldsymbol{I})\cdot\varphi_n(\boldsymbol{f}|\boldsymbol{0},\boldsymbol{K}),\end{aligned}$$

where $\varphi_n(\cdot|\boldsymbol{\mu},\boldsymbol{\Sigma})$ denotes the density function of n-variate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, and the notation "$\propto$" stands for "be proportional to."

Therefore, from well-known properties of multivariate normal distributions and Bayesian linear regression models (see, e.g., Lindley and Smith, 1972; Ravishanker and Dey, 2002), the posterior distribution, $p(\boldsymbol{f}|\mathcal{D},\sigma^2)$, is a multivariate normal distribution with

$$\begin{aligned}\mathrm{E}(\boldsymbol{f}|\mathcal{D},\sigma^2) &= \boldsymbol{K}(\boldsymbol{K}+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}\\ \mathrm{V}ar(\boldsymbol{f}|\mathcal{D},\sigma^2) &= \sigma^2\boldsymbol{K}(\boldsymbol{K}+\sigma^2\boldsymbol{I})^{-1}.\end{aligned}$$

Note that the mean vector of the Gaussian process prior is assumed to be zero.

The marginal distribution of $\boldsymbol{y}$, $p(\boldsymbol{y})$, is also given by a multivariate normal distribution,

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_n) \sim N_n(\boldsymbol{0}, \boldsymbol{\Psi}), \tag{2.5}$$

where $\boldsymbol{\Psi}$ is an $n \times n$ matrix, of which the (i,j)-th element is defined as

$$\boldsymbol{\Psi}(i,j) = \mathrm{Cov}(y_i, y_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sigma^2\delta_{ij}, \tag{2.6}$$

where δ_{ij} is the Kronecker delta. It is therefore straightforward to predict an output for the relevant test inputs (i.e., the new data points other than $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$; it is usually called *test data* in the engineering community and $\mathcal{D}$ is called the *training data*) based on the training data. Let $\boldsymbol{x}^*$ be a new input and let $f(\boldsymbol{x}^*)$ be the related nonlinear function. Then, the prediction of $f(\boldsymbol{x}^*)$ at the data point $\boldsymbol{x}^*$ can be obtained easily with the same principle because $f(\boldsymbol{x})$ can usually be assumed to be the same Gaussian process as the training data, and thus $(f(\boldsymbol{x}_1),\ldots,f(\boldsymbol{x}_n),f(\boldsymbol{x}^*))$ constitutes a $(n+1)$-variate normal vector. Consequently, the posterior distribution of $f(\boldsymbol{x}^*)$ given the training data $\mathcal{D}$ is also a Gaussian distribution, with mean and variance given by

$$\mathrm{E}(f(\boldsymbol{x}^*)|\mathcal{D}) = \boldsymbol{\psi}^T(\boldsymbol{x}^*)\boldsymbol{\Psi}^{-1}\boldsymbol{y}, \tag{2.7}$$

$$\mathrm{Var}(f(\boldsymbol{x}^*)|\mathcal{D}) = k(\boldsymbol{x}^*,\boldsymbol{x}^*) - \boldsymbol{\psi}^T(\boldsymbol{x}^*)\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}(\boldsymbol{x}^*), \tag{2.8}$$

where $\boldsymbol{\psi}(\boldsymbol{x}^*) = (k(\boldsymbol{x}^*, \boldsymbol{x}_1), \cdots, k(\boldsymbol{x}^*, \boldsymbol{x}_n))^T$ is the covariance between $f(\boldsymbol{x}^*)$ and $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))$, and $\boldsymbol{\Psi}$ is the covariance matrix of $(y_1, \cdots, y_n)$ given in (2.6). If y^* is the related output or response to $\boldsymbol{x}^*$, then its predictive distribution is also Gaussian, with the mean given by (2.7) and the variance

$$\hat{\sigma}^{*2} = \mathrm{Var}(f(\boldsymbol{x}^*)|\mathcal{D}) + \sigma^2. \tag{2.9}$$

Hence, when σ^2 and $\boldsymbol{\theta}$ are assumed to be known, the posterior distribution, marginal distribution, and predictive distributions are analytically computed via multivariate normal distributions as described above; obtaining the resulting posterior inference and numerical implementation can be quite straightforward. For example, the posterior mean in (2.7) is usually used to predict y^*; the prediction is denoted by $\hat{y}^*$. A 95% predictive interval of y^* can be calculated by $(\hat{y}^* - 1.96\hat{\sigma}^*,\ \hat{y}^* + 1.96\hat{\sigma}^*)$, where $\hat{\sigma}^{*2}$ is given by (2.9).

When σ^2 is assumed to be unknown but $\boldsymbol{\theta}$ is assumed to be known, the implementation also can be performed quite easily based on a Markov Chain Monte Carlo method, e.g., Gibbs sampling. We can calculate the full conditional posterior distributions of $\boldsymbol{f}_n$ and σ^2 by assigning a suitable prior to σ^2, for example, an inverse-gamma prior distribution. Thus, Monte Carlo simulation based on Gibbs sampling also can be applied in a straightforward manner. In addition, in the case of the unknown $\boldsymbol{\theta}$, several numerical schemes have been developed such as full Bayes methods based on Markov Chain Monte Carlo methods (see, e.g., Neal, 1997; MacKay, 1998a), empirical Bayes methods (see, e.g., Shi et al., 2005a; Shi and Wang, 2008), and other approximate Bayesian methods (see, e.g., Rue and Martino, 2007; Rue et al., 2009). Further details about these different computational schemes are discussed in Chapter 3.

2.2 Posterior consistency

In Bayesian inference, posterior distribution summarizes information regarding the unknown parameters, combined likelihood or probability model with prior distribution. As the sample size increases or we observe more and more data, we expect the posterior distribution to concentrate around the true distribution of the parameters. Broadly speaking, this summarizes what posterior consistency means.

Posterior consistency is a kind of frequentist validation of the updating method. If an oracle were to know the true value of the parameter, posterior consistency ensures that with enough observations one would get close to this true value. Posterior consistency also assures that as more and more observations accumulate, the data become more dominating over the role of the prior in inference. There are other interpretations of posterior consistency, related to merging opinions of Bayesian robustness with prior specification. A mathematical definition of posterior consistency and its systematic exposition can

be found in Ghosh and Ramamoorthi (2003), Ghosh et al. (2006), and references therein. In addition, it is also important to understand how to validate specific Bayesian methods in terms of consistency of posterior distributions. For a comprehensive survey of posterior consistency, the readers can refer to a constructive review by Choi and Ramamoorthi (2008). The review builds up some conceptual issues in consistency of posterior distributions; it also offers a critical comparison of various approaches to posterior consistency that have been investigated in the literature.

In order to define the formal concept of posterior consistency, we need to define several mathematical notations that will be used in the remainder of this chapter. Let θ be an unknown parameter and $Z_1, Z_2, \ldots, Z_n$ be n random variables whose joint distribution is $P_\theta^{(n)}$. In order to draw inference on θ, a prior distribution Π is assigned for θ, and it is updated to the posterior distribution given $Z_1, Z_2, \ldots, Z_n$, which we denote by $\Pi(\cdot|Z_1, Z_2, \ldots, Z_n)$. The sequence of posterior distributions $\{\Pi(\cdot|Z_1, Z_2, \ldots, Z_n)\}$ is said to be consistent at θ_0 if the posterior converges, in a suitable sense, to the degenerate measure at θ_0. Note that we have used a new flexible notation Π here. It should be understood as $\Pi(\theta) = p(\theta)$ and $\Pi(\theta \in A) = P(\theta \in A)$ for any set A.

The first posterior consistency result goes back to Laplace. In more recent times posterior consistency and asymptotic normality of the posterior were established for regular finite-dimensional models. Indeed, when the parameter space Ω is finite dimensional, posterior consistency can be achieved easily under fairly general conditions. Doob (1949) established an early and fundamental consistency result under weak conditions using the Martingale approach. Roughly speaking, Doob's theorem means that if there exists a consistent estimator, then the posterior distribution will tend to concentrate near the true value, with probability one under the joint distribution of the data and parameter. Doob's theorem guarantees that the posterior will be consistent for all θ except on a null set, a set of zero probability measure. However, if the prior misses the true value or the null set is huge, the conclusion to the theorem is not very attractive.

When the problem involves infinite-dimensional parameters, however, the consistency of posterior distributions is a much more challenging and complicated problem. That is, the intuition that the prior information will be dominated by the observables and thus posterior consistency will be achieved does not work necessarily in the nonparametric Bayesian procedures. In a seminal paper, Freedman (1963) gave a nonparametric example, where the posterior is inconsistent. Complementing this result, Diaconis and Freedman (1986) showed that in the nonparametric case inconsistency can occur, and suggested that instead of searching for priors that would be consistent at all unknown values of the parameter, it would be fruitful to study natural priors and identify points of consistency.

Nevertheless, there have been many positive results giving general condi-

tions under which features of posterior distributions are consistent in infinite-dimensional spaces. Soon after Freedman (1963) and Freedman (1965), a celebrated work by Schwartz (1965) provided conditions under which the posterior probability of a set A will go to 0. These conditions involved two parts, one on prior positivity of Kullback-Leibler neighborhoods and the other on the existence of certain test functions. Under the assumption of prior positivity of Kullback-Leibler neighborhoods, Barron et al. (1999) gave necessary and sufficient conditions for the posterior probability of A to go to 0. They provided other sufficient conditions to guarantee the posterior consistency based on the Hellinger neighborhood of the true distribution. These are support conditions that require prior positivity around the true parameter and a smoothness condition based on entropy-type bound on the roughness of densities. These results were then specialized to weak and L_1 neighborhoods by Barron et al. (1999), Ghosal et al. (1999), and Walker (2004).

However, those results on posterior consistency in the infinite-dimensional parameter space have focused mainly on density estimation, that is on estimating a density function for a random sample without assuming that the density belongs to a finite-dimensional parametric family. Some other research has paid attention to posterior consistency in general situations that include nonparametric and semiparametric regression problems. Since the Schwartz's theorem (Schwartz, 1965) was originally designed for independent and identically distributed random variables, its variants have been studied under a more general model setting. The theory of posterior consistency has been extended to independent but not-identically distributed observations as well as to some dependent cases including time series data; see, e.g., Amewou-Atisso et al. (2003), Choudhuri et al. (2004), Choi and Schervish (2007), and Wu and Ghosal (2008). Interested readers may refer to Choi and Ramamoorthi (2008) for in-depth information on posterior consistency and Hjort et al. (2010) for a comprehensive discussion on Bayesian nonparametrics.

2.2.1 Sufficient conditions for posterior consistency

As stated before, Π stands for a prior and $\{\Pi(\cdot|Z_1, Z_2, \ldots, Z_n)\}$ denotes a sequence of posterior distributions. The sequence of posteriors is said to be consistent at θ_0 if

$$\{\Pi(U|Z_1, Z_2, \ldots, Z_n)\} \to 1 \text{ a.s. } P_{\theta_0}^{\infty}$$

for all neighborhoods U of θ_0. In other words, for each neighborhood U of the true parameter value θ_0, we compute the posterior probability $\Pi(\theta \in U|Z_1, Z_2, \ldots, Z_n)$ as a function of the data. To say that the posterior distribution of θ is *almost surely* consistent means that, for every neighborhood U,

$$\lim_{n\to\infty} \Pi(\theta \in U|Z_1, Z_2, \ldots, Z_n) = 1, \tag{2.10}$$

almost surely with respect to the joint distribution of the infinite sequence of data values. Similarly, *in-probability* consistency means that for all U, $\Pi(\theta \in U|Z_1, Z_2, \ldots, Z_n)$ converges to 1 *in probability*.

Thus, posterior consistency involves examining a posterior probability of a set of parameter values as the sample size, n, goes to infinity. The set is any neighborhood of the true parameter θ_0 and it is necessary to define suitable topologies and corresponding neighborhood for the parameter of interest. Note that the parameter space for Gaussian process regression models involves a space of regression functions that needs to be specified with care. We will discuss different choices of topology of regression functions in the next subsection. After defining a suitable topology of the parameter and the neighborhood of the true value of the parameter, then we may verify the posterior distribution consistency mainly by checking sufficient conditions for a general posterior consistency theorem. An extension of Schwartz's theorem to independent but nonidentically distributed cases for almost sure consistency was discussed in Choi and Schervish (2007), and we make use of such an extension to verify posterior consistency of Gaussian process regression problems in the next subsection. The formal statement of the extension is provided in Theorem A.1 in the Appendix.

Simply speaking, in order to obtain the result in (2.10), the posterior probability $\Pi(\theta \in U^C|X_1, X_2, \ldots, X_n)$ is written as the ratio of two quantities:

$$\begin{aligned} \Pi(U^C|Z_1, Z_2, \ldots, Z_n) &= \frac{J_{U^C}(Z_1, Z_2, \ldots, Z_n)}{J(Z_1, Z_2, \ldots, Z_n)} \\ &= \frac{\int_{U^C} \prod_{i=1}^n \frac{f(z_i|\theta)}{f(z_i|\theta_0)} \Pi(d\theta)}{\int_\Theta \prod_{i=1}^n \frac{f(z_i|\theta)}{f(z_i|\theta_0)} \Pi(d\theta)}, \end{aligned} \tag{2.11}$$

where U^C is a complement of U and $J_A(\boldsymbol{Z})$ denotes the probability $P(\boldsymbol{Z} \in A)$. Then, we consider an upper bound of the numerator and a lower bound of the denominator of (2.11) separately; we do so by using two sufficient conditions, namely, the prior positivity and the existence of uniformly consistent tests as stated in Theorem A.1, so that the posterior probability eventually converges to zero. The asymptotic result in Theorem A.1 is known as *exponential consistency* (Choi and Ramamoorthi, 2008), i.e., the posterior probability goes to zero exponentially. Indeed, all the general consistency results in the literature establish exponential consistency. The formal definition of the term of exponential consistency along with further details is also given in the Appendix.

In the next section we provide the results of posterior consistency under the Gaussian process regression model (2.3) by verifying the sufficient conditions $(A1)$ and $(A2)$ in Theorem A.1.

2.3 Asymptotic properties of the Gaussian process regression models

2.3.1 Posterior consistency

Posterior consistency related to Gaussian process methods has been investigated in Ghosal and Roy (2006) and Choi (2007) for nonparametric binary regression, Tokdar and Ghosh (2007) for density estimation, and Choi (2005) and Choi and Schervish (2007) for the nonparametric GPR model. In this section, we illustrate *almost sure consistency* for posterior distributions in the GPR model using Theorem A.1 in Appendix A.1. Note that the posterior consistency is mainly discussed for the GPR model with the case of one-dimensional covariate, i.e., $Q = 1$ in (2.3), and the extension for multidimensional covariate is briefly described at the end of this subsection.

In order to establish posterior consistency, neighborhoods of the true parameter, i.e., neighborhoods of the true regression function, must be specified first. For this purpose, different topologies of functions can be considered, depending on the type of covariate in the regression model, whether it is a fixed design covariate or a randomly chosen covariate with a probability distribution. Then, consistency results based on these topologies, such as L_1 topology or a topology based on Hellinger distance, can be discussed (see, e.g., Ghosal and Roy, 2006; Choi and Schervish, 2007). In this subsection, we provide a consistency result with a specific metric based on an empirical probability measure for illustrating posterior consistency in Gaussian process regression.

We first define notations corresponding to the ones used in Theorem A.1. The parameter θ in Theorem A.1 is (f, σ) with the true value of parameter, $\theta_0 = (f_0, \sigma_0)$. The parameter space Ω is a product space of a function space Ω_1 and $\mathcal{R}^+$. Let θ have prior Π, a product measure, $\Pi = \Pi_1 \times \Pi_2$. We then need to define joint neighborhoods of f and σ, which are similar to U_n in Theorem A.1. Those neighborhoods contain $\theta_0 = (f_0, \sigma_0)$. When the covariate values are fixed in advance, we consider the neighborhoods based on an empirical measure of the design points. Let Q_n be the empirical probability measure of the design points, $Q_n(x) = n^{-1} \sum_{i=1}^{n} I_{x_i}(x)$, where $I_{x_i}(x)$ is an indicator function taking value 1 when $x = x_i$ and 0 otherwise. Based on Q_n, we define the following neighborhood,

$$W_{\varepsilon,n} = \left\{ (f, \sigma) \ : \ \int |f(x) - f_0(x)| dQ_n(x) < \varepsilon, \ \left| \frac{\sigma}{\sigma_0} - 1 \right| < \varepsilon \right\}.$$

The marginal neighborhood of f or σ contains the joint neighborhood that we consider. Thus, if we prove that the posterior probability of the joint neighborhood converge is almost surely to 1, then it obviously follows that the posterior probability of marginal neighborhoods converges almost surely to 1. Note that in the simple GPR model in (2.3), the noise variance σ^2 is assumed to be known as σ_0^2. Hence, we merely discuss posterior consistency in terms of a marginal neighborhood of f in the remainder of this section.

Let $\{Y_1,\ldots,Y_n\}$ be the random variables associated with the observations $\{y_1,\ldots,y_n\}$. The following theorem provides a formal statement of posterior consistency for a GPR model.

Theorem 2.1. *Let P_0 denote the joint conditional distribution of $\{Y_n\}_{n=1}^{\infty}$ given the covariate assuming that f_0 is the true response function. Suppose that the values of the covariate in $[0,1]$ are fixed, i.e., known ahead of time. Then for every $\varepsilon > 0$,*

$$\Pi\left\{f \in W_{\varepsilon,n}^{C} | \mathcal{D}\right\} \to 0 \ a.s.\ [P_0]. \tag{2.12}$$

To prove the above theorem, we need to verify conditions (A1) and (A2) in Theorem A.1. We now first look at condition (A1). This condition is closely related to the Kullback-Leibler (KL) support condition as mentioned in Appendix A.1. Since the observed data, $\mathcal{D}$, from the GPR model in (2.3) are independent but nonidentically distributed, a stronger form of KL support is needed as described in condition $(A1)$. Note that we have $Y_i \sim N(f_0(x_i),\sigma_0^2)$, $i = 1,\ldots,n$ under the GPR model. By computing the mean and variance of the log-likelihood given in $(A1)$, we obtain

$$KL_i(\theta_0;\theta) = \mathrm{E}_{\theta_0} \log \frac{f_i(Y_i;\theta_0)}{f_i(Y_i;\theta)} = \frac{1}{2}\frac{[f_0(x_i)-f(x_i)]^2}{\sigma_0^2},$$

$$V_i(\theta_0;\theta) = \mathrm{Var}_{\theta_0} \log \frac{f_i(Y_i;\theta_0)}{f_i(Y_i;\theta)} = [f(x_i)-f_0(x_i)]^2.$$

Define for every $\varepsilon > 0$,

$$B = \left\{ f : \sup_{x\in[0,1]} \|f(x)-f_0(x)\|_\infty < \varepsilon \right\},$$

where $\|\cdot\|_\infty$ is the usual supremum norm. Then, it is sufficient to show that $\Pi(B) > 0$ to verify condition $(A1)$.

In the Gaussian process prior for f, as a predetermined covariance function, we consider the squared exponential covariance function (2.4) with $Q = 1$:

$$k(x,x';w) = \nu_0 \exp\left(-\frac{1}{2}w(x-x')^2\right).$$

Note that we do not treat w as fixed like we did in the previous subsection; instead we assign a prior for w here. We also assume that the support of w is $\mathcal{R}^+$ and, thus, w determines primarily the shape of the covariance function. It can be observed that as w becomes extreme (close to infinity), the shape of the covariance function becomes flat. Specifically, we assume that there exists δ, b_1, and b_2 ($0 < \delta < 1/2$ and $b_1,\ b_2 > 0$) such that

$$\kappa\left\{w > n^{\delta}\right\} < b_1 \exp(-b_2 n),\ \ \forall n \geq 1,$$

where κ is the prior distribution of w. A similar assumption on the prior distribution of w can be found in Ghosal and Roy (2006).

Note that the prior positivity of condition $(A1)$, i.e., $\Pi(B > 0)$, depends on the support of the Gaussian process prior; the squared exponential covariance function in (2.4) with $Q = 1$ ensures that the support of the Gaussian process prior contains every continuously differentiable function. In addition, without a smoothing parameter w in the covariance function of the Gaussian process, only functions in the reproducing kernel Hilbert Space (RKHS) of the fixed covariance function would be in the support of the Gaussian process prior, which could not suffice $\Pi(B) > 0$. Technical details can be found in Tokdar and Ghosh (2007) and Ghosal and Roy (2006). For the RKHS of Gaussian process priors, a constructive review and further discussions are given by van der Vaart and van Zanten (2008b).

In addition, if we consider the kernel of the squared exponential covariance function, it can be easily seen by Taylor's expansion that

$$\exp(-h^2) = 1 - h^2 + O(h^4), \text{ as } h \to 0, \tag{2.13}$$

which guarantees the existence of a continuous sample derivative $f'(x)$ with probability one, and that $f'(x)$ is also a Gaussian process (see, e.g., Theorem A.3 in Appendix A.5 and Cramer and Leadbetter (1967) for further details).

Note that many other covariance functions that are widely used in practice, for example, those listed in Table 4.1, also satisfy $\Pi(B > 0)$ as well as analytic conditions similar to (2.13). A formal discussion is given in Appendix A.2. Existence of the continuous sample derivative condition is also useful for the construction of uniformly consistent tests for the verification of $(A2)$. For detailed descriptions of analytical properties of covariance functions and the smoothness of sample paths of Gaussian processes, good references are Cramer and Leadbetter (1967), Adler (1981), Adler (1990), and Abrahamsen (1997).

We now describe briefly how to verify condition (A2), i.e., the existence of tests. When the prior under consideration satisfies certain properties as discussed before, the crucial condition for posterior consistency is to construct tests that make the true parameter be separated from the outside of the suitable neighborhoods of the parameter. Under appropriate conditions on the regression function, it can be shown that there exist tests, of which the type I error and the type II error probabilities are exponentially small for distinguishing the true parameter from the complements of the suitable neighborhoods of the parameter. Those test functions that are indeed uniformly consistent enable us to verify the condition $(A2)$ in Appendix A.1.

Previous results for independent but nonidentically distributed models, as in Amewou-Atisso et al. (2003), Choudhuri et al. (2004), Ghosal and Roy (2006), and Choi and Schervish (2007), tried to establish the existence of uniformly consistent tests as we will present in this section. Alternatively, Le

Cam (1986) showed that for independent nonidentically distributed variables, tests with exponentially small errors exist when we use the average squared Hellinger distance to separate densities and convex sets. That is, uniformly consistent tests are always obtained if the entropy with such a distance is controlled. Similar results to entropy condition in Ghosal et al. (1999) have been investigated by Ghosal and van der Vaart (2007), in the test construction for the convergence rates of posterior distributions for non i.i.d. observations. On the other hand, Walker's sufficient conditions (Walker, 2004) also can be adapted in this case (Choi and Ramamoorthi, 2008).

First, we construct test functions that distinguish the true parameter f_0 from $W_{\varepsilon,n}^C$ in the sense that the type I error and type II error probabilities of these test functions are exponentially small. However, since we have an infinite dimensional parameter space, it is not always feasible to construct such test functions. Thus, we consider a sieve, Ω_n, which grows eventually to the space of continuously differentiable function on $[0,1]$. For each element of the sieve, we construct a test, as required by condition $(A2)$. The specific construction of the sieve and uniformly consistent test is outlined in Appendix A.3.

The other condition for prior distributions which needs to be satisfied concerns the probability of Ω_n^C. The subcondition (iii) of $(A2)$ requires that there exist constants C_2 and c_2 such that $\Pi(\Omega_n^C) \leq C_2 e^{-c_2 n}$. To obtain a suitable probability bound for Ω_n^C using the Gaussian process prior with the squared exponential covariance function, we can show that there exist constants C_4 and c_4 such that

$$\Pi(\Omega_n^C) \leq C_4 \exp(-c_4 n). \tag{2.14}$$

The proof is based on the existence of continuous sample path derivative and the result that the sample path derivative is also a Gaussian process, which is summarized in Lemma A.1 in the Appendix. The above results are also true for Gaussian process prior with *Assumption P* as specified in Appendix A.2.

We have therefore finished the verification of two sufficient conditions $(A1)$ and $(A2)$ for Theorem A.1; this results in Theorem 2.1. Other types of posterior consistency with different neighborhoods can be similarly established. For further discussions, the readers may refer to Ghosal and Roy (2006) and Choi and Schervish (2007).

Up to this point we limited our discussion on posterior consistency to the case of one-dimensional covariate for the regression function based on the squared exponential covariance function; these results can be extended under more general model structures. There are several open issues that are worth further consideration in posterior consistency of Gaussian process regression methods. For example, we can think about the case in which the covariance function of the Gaussian process has (finitely many) parameters that need to be estimated, and the case of multidimensional covariates. We briefly give a remark on these two issues.

In a predetermined squared exponential covariance function in (2.4), the

parameter ν_0 can be treated as a scaling parameter of the covariance function, and it can regulate the variability of the Gaussian process. It is typically the case in applications that the various parameters that govern the smoothness of the Gaussian process prior are not sufficiently well understood to be chosen with certainty. In such cases, there are several ways to choose hyper-parameters, which we will discuss in detail in Chapter 3. One way is to assign a prior distribution for the additional parameter ν_0 and treat ν_0 as another random quantity to be estimated. The posterior consistency results presented previously are associated with a fixed ν_0. The question we naturally would ask now is how the posterior consistency is affected if we treat ν_0 as random. In dealing with these kinds of hyper-parameters, posterior consistency also can be established but under stronger conditions such as compactness and continuity of the covariance function with respect to the hyper-parameter (Choi, 2007) or additional conditions similar to those for w (Ghosal and Roy, 2006).

Although we assumed that the covariate is one-dimensional, much concern also lies in multidimensional covariates. In terms of the posterior consistency, Ghosal and Roy (2006) and Choi and Schervish (2007) considered the case of multidimensional covariates for regression problems under a certain topology of multidimensional probability functions. In general, the main difficulty in dealing with multidimensional or high-dimensional regression function lies in the so-called phenomenon, the "curse of dimensionality," which implies that the problem gets harder very quickly as the dimension of the observations increases. This problem also affects the posterior consistency and makes the results less promising. As the sample size increases, the posterior is also expected to be consistent for models with multidimensional covariates. However, in order to achieve the posterior consistency in the same fashion as the one-dimensional regression function, the sample size should be much larger than that in the one-dimensional case, which is not practical. As a result, stronger assumptions on design points or regression functions are required, or different topologies in the parameter space can be considered. For example, Ghosal and Roy (2006) treated L_1 consistency of a probability function in one dimension with an assumption about the fixed design points, while in higher dimensions, they considered the consistency with respect to the empirical measure of the design points. Choi and Schervish (2007) also used stronger assumptions than those for the case of a one-dimensional regression function such as uniformly boundedness on the partial derivatives of the regression function.

Once we establish posterior consistency, we are able to discuss the consistency of Bayes estimate of the regression function, or existence of consistent Bayes estimator from the predictive point of view. When we are interested in estimating the conditional density, $p(y|x)$, a commonly used Bayes estimator is the predictive density, given by

$$\hat{p}_n(y|x) = \int p(y|x,\theta) d\Pi(\theta|\mathcal{D}). \tag{2.15}$$

It is known that Bayes predictive estimates inherit the convergence property of the posterior (Ghosh and Ramamoorthi, 2003, Proposition 4.2.1). That is, if posterior is consistent, then the Bayes estimate (2.15) is also consistent. In addition, a Bayesian estimate of the conditional mean $f_0(x)$ can be defined in a similar way. Let

$$f_0(x) = \mathrm{E}_{p_0}[Y|X = x] = \int y p_0(y|x) dy$$

be the true regression function. Then,

$$\hat{f}_n(x) = \mathrm{E}_{\hat{p}_n}[Y|X = x]$$

is the predictive regression function, or a Bayes estimate of the regression function. Therefore, posterior consistency ensures that the predictive estimate of the regression function, $\hat{f}_n(x)$, is also consistent in some sense (see, e.g., Shen and Wasserman, 2001; Ge and Jiang, 2006; Jiang, 2006).

Figure 2.1 displays a curve fitting and prediction based on Gaussian process regression in (2.3) with fixed values of hyper-parameters. As the number of observations increases, it appears that the posterior mean curve and predictive intervals are more accurate. Thus, Figure 2.1 shows a good performance of prediction with an increasing sample size, and provides an empirical illustration of consistent Bayesian estimation of the regression function. This may be regarded as an intuitive reasoning on posterior consistency, i.e., the concept that as the sample size increases, the posterior distribution of the regression function concentrates around the true regression function, f_0.

Note that the idea of consistency related to predictive distribution is closely related to another concept of consistency, so-called *information consistency* (Seeger et al., 2008), which will be discussed in the next section.

2.3.2 Information consistency

As discussed in the previous section, asymptotic properties of Gaussian process regression in terms of posterior consistency have been well studied in various general settings, and results have been proved using abstract sufficient conditions such as metric entropy and uniformly consistent tests. Often, those sufficient conditions are a little hard to validate and not intuitive when applied to concrete models.

Alternatively, Seeger et al. (2008) focused on the concept of information consistency for Gaussian process regression models. They presented information consistency results for nonparametric sequential prediction with Gaussian processes, connected to nonparametric minimum description length (MDL) through the sequential approach (Grünwald, 2007). They also obtained information consistency rates for a wide range of covariance functions using kernel eigenvalues asymptotics. These results also depend strongly on the covariance function of the prior process similar to the case of posterior consistency,

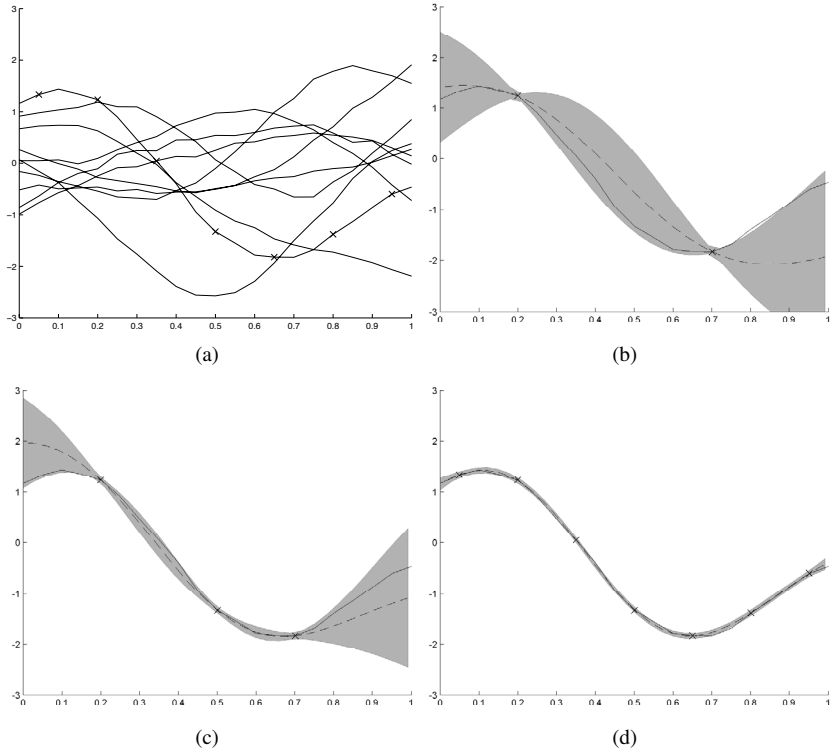


Figure 2.1 *Curve fitting based on the posterior distribution from a GPR model: (a) 10 curves drawn randomly from a GPR model, in which 7 data points marked "×" are used as training data; (b)-(d) estimated posterior mean and 95% predictive intervals (in the shade) by using 2, 3, or 7 observations, respectively, where dashed lines stand for the posterior mean and solid lines stand for the true curves.*

thereby giving a new interpretation to penalization with regards to reproducing kernel Hilbert space (RKHS) norms and to commonly used covariance function classes and their parameters.

The concept of *information consistency* or *consistency in information* was first introduced in Barron (1999), and it is an information criterion in terms of the Césaro KL risk. In other words, the information criterion is the Césaro average of the sequence of prediction errors (Grünwald, 2007). Let $KL(p,q) = \int \log(p/q)dP$ denote the Kullback-Leibler (KL) divergence between two probability densities p and q, equivalent to the relative entropy in information theory (Cover and Thomas, 2006), denoted by

$$D[p\|q] = \int (\log p - \log q)dP. \tag{2.16}$$

We can obtain a lower bound of $D[P(y_1,\ldots,y_n|f)\|P_{bs}(y_1,\ldots,y_n)]$, where $P_{bs}(y_1,\ldots,y_n)$ denotes a Bayesian Gaussian process prediction strategy based on n observations, and $P_{bs}(y_1,\ldots,y_n)=\int p_f(y^*)d\Pi(f|\mathcal{D})$, where y^* is a future observation. Thus,

$$D[P(y_1,\ldots,y_n|f)\|P_{bs}(y_1,\ldots,y_n)] \leq \frac{1}{2}\|f\|^2_{\boldsymbol{K}}+\frac{1}{2}\log|\boldsymbol{I}_n+c\boldsymbol{K}|, \tag{2.17}$$

where $\|f\|_{\boldsymbol{K}}$ is the RKHS norm of f, and c is a certain constant. Based on the lower bound in (2.17), it can be shown that the expected KL divergence divided by the sample size converges to zero as the sample size increases (see the details in Seeger et al., 2008). Specifically, using the eigen-expansion of the covariance kernel and Widom's theorem (Widom, 1963), a lower bound of the expected regret term can be identified (c.f. Grünwald, 2007), i.e., $\mathrm{E}[\log|\boldsymbol{I}_n+c\boldsymbol{K}|]$, for widely used covariance functions such as the squared exponential and the Matérn kernels in Table 4.1. That is, assuming $k(\cdot,\cdot)$ has an eigen-expansion $k(x,x')=\sum_{s\geq 0}\lambda_s\phi_s(x)\phi_s(x')$, where $\{(\lambda_s,\phi_s)|s\geq 0\}$ is a complete orthonormal eigen-system of $k(\cdot,\cdot)$, the lower bound of the regret term and the expected regret (Seeger et al., 2008, Lemma 1) are given, respectively, by

$$R=\log|\boldsymbol{I}_n+c\boldsymbol{K}| \leq \sum_{s\geq 0}\log\left(1+c\lambda_s\sum_{i=1}^{n}\phi_s(x_i)^2\right) \tag{2.18}$$

and

$$\mathrm{E}(R)\leq\sum_{s\geq 0}\log(1+c\lambda_s n).$$

Furthermore, depending on the distribution of covariate x as well as the covariance function, an information rate bound can also be obtained. For example, when the covariate distribution is chosen to be Gaussian, and the squared exponential covariance function is used, a tight bound on $\mathrm{E}(R)$ is given by

$$\mathrm{E}(R)=O\left((\log n)^2\right),$$

while the bound for the Matérn covariance function with bounded support covariates is given by

$$\mathrm{E}(R)=O\left(n^{1/(2\nu+1)}(\log n)^{2\nu/(2\nu+3)}\right).$$

In this way, information convergence rate bounds depend strongly on the specifics of the model in a fairly direct manner; see the details in Seeger et al. (2008). Similar ideas were also studied to bound $\mathrm{E}(R)$ for the squared exponential kernel (Opper and Vivarelli, 1999) and to understand Gaussian process regression in the context of the equivalent kernel (Sollich and Williams, 2005), which will be briefly discussed in the next section.

In an unpublished manuscript, van der Vaart and van Zanten (2010) have elaborated those results in Seeger et al. (2008) by considering the case that the true response function f_0 is contained in the reproducing kernel Hilbert space of the prior. They have also characterized the information rates in a more accurate way by studying small ball probabilities and entropy calculations in the Gaussian process prior. Their approach was based on the general method for studying posterior convergence rates and Gaussian process priors in Ghosal and van der Vaart (2007) and van der Vaart and van Zanten (2008a).

Formal statement of the main theorem and further discussions can be found in Seeger et al. (2008). The detailed proofs can also be found there. Further discussions on incorporating Gaussian process regression and other kernel methods to *minimum description length* principle can be found in Grünwald (2007).

2.4 Further reading and notes

Consistency is merely the basic asymptotic properties of Bayesian procedures. Issues such as rates of convergence and asymptotic normality have received much attention, and there have also been positive results in Bayesian nonparametric that includes Gaussian process methods. Specifically, when the posterior consistency is achieved, the next inherent task is to investigate the convergence rate of the posterior distribution. Based on the same framework with posterior consistency, sufficient conditions to determine the rate that posterior distribution converges to zero was identified mainly for i.i.d. observations (see, e.g., Ghosal et al., 2000; Shen and Wasserman, 2001). For non i.i.d. observations which include nonparametric regression cases, Ghosal and van der Vaart (2007) presented general results on convergence rates of posterior distribution. Similar sufficient conditions to those used to prove posterior consistency are required to study the rate of convergence. The test functions that we constructed in the previous section could also be used to analyze the rate that the posterior probability concentrates around the true parameter. The challenge is to find a rate that a prior probability shrinks as we consider a decreasing sequence of Kullback-Leibler neighborhoods. To be specific, this problem involves finding lower tail probabilities for Gaussian processes, known as a small ball problem or a small deviation problem. Historically, this problem has been studied mainly in the setting of Brownian motion or its variants. There have been several positive results obtaining convergence rates of posterior distributions, in the context of the RKHS of Gaussian process priors (c.f. van der Vaart and van Zanten, 2008b). For example, van der Vaart and van Zanten (2008a) obtained upper bounds for rates of convergence of posterior distribution with Gaussian process priors, and Castillo (2008) obtained the precise rate of convergence of posteriors for Gaussian process priors by studying the lower bounds for posterior rates. As mentioned in the previous section, van der Vaart and van

Zanten (2010) also obtained information rates of nonparametric Gaussian process methods based on the general results in van der Vaart and van Zanten (2008a); furthermore, they illustrated the computation of the upper bounds for the squared exponential covariance functions and the Matérn classes. In addition, the precise information rates can be obtained based on the approach by Castillo (2008).

On the other hand, another theoretical issue that we omitted here is the *regularization* viewpoint, which is also closely related to the RKHS framework and the representer theorem (Kimeldorf and Wahba, 1971). Chapter 6 of Rasmussen and Williams (2006) provided a clear exposition of these ideas, and discussed a number of concepts and models in regards to Gaussian process prediction, as well as theoretical aspects in these regards. In addition, in the same framework of the regularization, the concept of *equivalent kernel* (Silverman, 1984) can be used to understand Gaussian process regression. Sollich and Williams (2005) explained how to approximate the equivalent kernel of the widely used covariance functions (readers may also refer to Chapter 7 of Rasmussen and Williams, 2006). In addition, the monograph by Eggermont and LaRiccia (2009) is a good reference describing more theoretical details and fundamentals of the equivalent kernels and related spline smoothing.

This page intentionally left blank

Chapter 3

Inference and computation for Gaussian process regression model

Simply speaking, statistical inference based on a Gaussian process regression model can be performed in the following procedure. First of all, a prior distribution for the unknown regression function needs to be specified using a Gaussian process, namely, a Gaussian process prior should be defined. The concept of a Gaussian process prior was described in (1.11), in which a form of the covariance function $k(\cdot,\cdot)$ needed to be selected. Second, the posterior distribution of a regression curve is derived via Bayes theorem based on a multivariate Gaussian distribution with a suitable mean vector and a covariance matrix, for example, (2.7) and (2.8), after we observe a set of data. Although the posterior consistency can be achieved with suitably chosen covariance functions, as discussed in the previous chapter, under some regularity conditions, the empirical selection of covariance function and the selection of hyper-parameters in the covariance function are still important issues to be considered. Since making a suitable choice of a covariance function and its hyper-parameters can improve the prediction accuracy, this is particularly important for dealing with datasets of small and medium sample sizes.

Another problem existing in the Gaussian process regression implementation concerns the massive computation of the various matrices. Specifically, the predictive mean and variance in (2.7) and (2.8) involve the computation of the inverse of $n \times n$ covariance matrices, where n is the sample size of the training dataset. Thus, the complexity in computation grows at rate $O(n^3)$, resulting in a heavy computational burden associated with large training datasets. This can be a major limitation in the use of the GPR models. Fortunately, a variety of numerical methods have been developed to solve this problem.

In Section 3.1, we discuss how to use an empirical Bayesian approach to select hyper-parameters. In Section 3.2, a hyper-prior distribution is defined for the hyper-parameters, and a fully Bayesian approach using Markov chain Monte Carlo (MCMC) methods is discussed for both model learning and curve prediction. Some efficient methods and numerical algorithms used in imple-

mentation are discussed in Section 3.3. The selection of covariance functions as well as the variable selection problem is discussed in the next chapter.

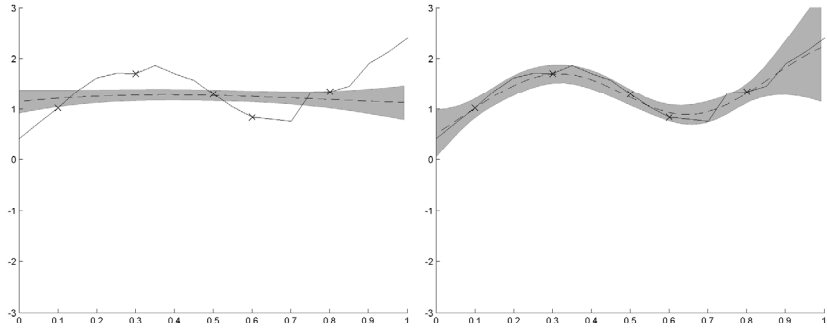
3.1 Empirical Bayes estimates

We consider a Gaussian process regression model with a given covariance function similar to the one discussed in Chapter 2, namely, $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon$, where $f(\cdot)$ has a Gaussian process prior meaning that $f(\cdot) \sim N(0,\ k(\cdot,\cdot))$ and $\mathrm{Cov}(f(\boldsymbol{x}), f(\boldsymbol{x}')) = k(\boldsymbol{x},\boldsymbol{x}')$. For example, we take an extended form of covariance kernel $k(\cdot,\cdot)$ based on a squared exponential covariance kernel given by (2.4). To be specific, the following covariance kernel is considered:

$$\begin{aligned} k(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}) &= \mathrm{Cov}(f(\boldsymbol{x}), f(\boldsymbol{x}')) \\ &= v_0 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w_q (x_q - x_q')^2\right) + a_1 \sum x_q x_q'. \end{aligned} \tag{3.1}$$

In a one-dimensional covariate case, i.e., $Q = 1$, we have $\boldsymbol{\theta} = (w_1, v_0, a_1)$. Note that the above covariance kernel is the combination of two kernel functions. The first one is a squared exponential kernel, while the second one is a nonstationary linear covariance kernel; see the details in Section 4.1.1. In the squared exponential covariance part, w_1^{-1} is the length scale (or so-called bandwidth parameter), and v_0 is the vertical scale of variations of a typical function of the input, known as the scaling parameter. These two hyper-parameters have the same role with the squared covariance function in (2.4). For instance, a very large value of the length scale means that y is expected to be a constant function of the input. In the linear covariance part, hyper-parameter a_1 defines the scale of nonstationary linear trends. The values of the hyper-parameters therefore offer the prior information about the model, and control the shape and the variability of the regression curve. If the values are misspecified, it may often result in a poor curve fitting especially when the sample size is small. This is demonstrated by an example presented in Figure 3.1. In both plots (a) and (b) of Figure 3.1, the same data with five observations marked by crosses are used to predict the curve, but different values of hyper-parameters are specified. The model corresponding to panel (a) is clearly not a good fit to the true curve, whereas the model for panel (b) gives a much better fit. Specifically, the hyper-parameters $\boldsymbol{\theta} = (w_1, \mathrm{v}_0, a_1)$ used in panel (b) are estimated by the empirical Bayes approach that is discussed below. On the other hand, the synthetic values of $\boldsymbol{\theta}$ in panel (a) are chosen to be similar to those from the empirical Bayes estimates in (b) except the value of w_1, for which a much smaller value is taken. Note that such a small value of w_1 implies a rather large value of the length scale of w_1^{-1}, which makes the function and thus the prediction nearly a constant.

As outlined above, the performance of Bayesian inference for Gaussian



(a) (b)

Figure 3.1 *Curve fitting based on the posterior distribution from a Gaussian process model with different values of hyper-parameters of (a)* $w_1 = 1$, $v_0 = 5$, $a_1 = 1$, *and* $\sigma_\varepsilon^2 = 0.01$ *and (b)* $w_1 = 16.95$, $v_0 = 5.3$, $a_1 = 0.6$, *and* $\sigma_\varepsilon^2 = 0.01$ *which are the empirical Bayes estimates; in both panels, the dashed line stands for the posterior mean, the solid line stands for the true curves, and the shade stands for the 95% predictive interval.*

process regression models often depends on the choice of the hyper-parameters $\boldsymbol{\theta}$. Unless each hyper-parameter has a clear meaning or physical interpretation and we have a very good prior knowledge on what value it may take, we should be cautious on selecting its value. In this case, rather than making assumptions on the probability structure for the hyper-parameters, the empirical Bayes approach uses the observed data to estimate them (Carlin and Louis, 1996). In this section, we discuss how to find an empirical Bayes estimate for $\boldsymbol{\theta}$ from a marginal density function. Alternatively, we can define a hyper-prior distribution for $\boldsymbol{\theta}$. This approach is discussed in the next section by using a Markov chain Monte Carlo algorithm.

Let us focus now on the empirical Bayes approach. As before, $\mathcal{D} = \{\boldsymbol{y}, \boldsymbol{X}\} = \{(y_i, x_{i1}, \ldots, x_{iQ}), i = 1, 2, \ldots, n\}$ denotes a set of observed data, where y_i is the response variable and $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iQ})^T$ is a Q-dimensional vector of covariates. A Gaussian process regression model is generally formulated as

$$y_i | f_i \overset{\text{ind}}{\sim} g(f_i) \quad \text{and} \tag{3.2}$$

$$(f_1, \ldots, f_n) \sim GP(\mathbf{0}, k(\cdot, \cdot; \boldsymbol{\theta})), \tag{3.3}$$

where '$\overset{\text{ind}}{\sim}$' means "independently distributed," and $GP(\mathbf{0}, k(\cdot, \cdot; \boldsymbol{\theta}))$ stands for a Gaussian process with zero mean and covariance function $k(\cdot, \cdot; \boldsymbol{\theta})$, where the covariance function contains the hyper-parameter $\boldsymbol{\theta}$. Here, $g(f_i)$ is a normal distribution with mean f_i and variance σ_ε^2 (from now on, we use this notation to emphasize that it is the variance of the error item ε),

$$y_i | f_i \overset{\text{ind}}{\sim} N(f_i, \sigma_\varepsilon^2) \tag{3.4}$$

if y_i is assumed to have a normal distribution. Note that the form of $g(f_i)$ in (3.2) depends on the type of responses. For example, modeling nonnormal responses such as binary data or Poisson data, the distribution of y_i would be generalized to an appropriate distribution function that belongs to the exponential family. In these cases, generalized Gaussian process regression models can be considered, and they are discussed in Chapter 7.

Thus, from the model specification of (3.2) and (3.3), the marginal distribution of $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ given $\boldsymbol{\theta}$ is written by

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{f}) p(\boldsymbol{f}|\boldsymbol{\theta}) d\boldsymbol{f}, \tag{3.5}$$

where $p(\boldsymbol{y}|\boldsymbol{f}) = \prod_{i=1}^{n} g(f_i)$ and $\boldsymbol{f} \sim N(\boldsymbol{0}, \boldsymbol{K})$. The (i,j)-th element of the covariance matrix $\boldsymbol{K}$ is calculated by using the covariance function

$$\boldsymbol{K}(i,j) = \mathrm{Cov}(f_i, f_j) = k(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}). \tag{3.6}$$

Consequently, for the continuous response with normal distribution as given in (3.4), the marginal distribution (3.5) has an analytical form as a multivariate normal. Following the results given in (2.5) and (2.6), the marginal distribution of $\boldsymbol{y}$ is a normal distribution $N(\boldsymbol{0}, \boldsymbol{\Psi})$, where $\boldsymbol{\Psi} = \boldsymbol{K} + \sigma_\varepsilon^2 \boldsymbol{I}$. Hence, the marginal log-likelihood of $\boldsymbol{\theta}$ is given by

$$l(\boldsymbol{\theta}|\mathcal{D}) = -\frac{1}{2}\log|\boldsymbol{\Psi}(\boldsymbol{\theta})| - \frac{1}{2}\boldsymbol{y}^T \boldsymbol{\Psi}(\boldsymbol{\theta})^{-1} \boldsymbol{y} - \frac{n}{2}\log 2\pi, \tag{3.7}$$

and its gradient is

$$\begin{aligned} \frac{\partial l}{\partial \theta_j} &= -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Psi}^{-1}\frac{\partial \boldsymbol{\Psi}}{\partial \theta_j}\right) + \frac{1}{2}\boldsymbol{y}^T \boldsymbol{\Psi}^{-1}\frac{\partial \boldsymbol{\Psi}}{\partial \theta_j}\boldsymbol{\Psi}^{-1}\boldsymbol{y} \\ &= \frac{1}{2}\mathrm{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \boldsymbol{\Psi}^{-1})\frac{\partial \boldsymbol{\Psi}}{\partial \theta_j}\right), \end{aligned} \tag{3.8}$$

where $\boldsymbol{\alpha} = \boldsymbol{\Psi}^{-1}\boldsymbol{y}$, and the notation $\mathrm{tr}(\boldsymbol{A})$ means the trace of matrix $\boldsymbol{A}$ – the sum of the diagonal elements of $\boldsymbol{A}$. In addition, the second derivatives can be calculated accordingly, and we list the formula here:

$$\frac{\partial^2 l}{\partial \theta_i \partial \theta_j} = \frac{1}{2}\mathrm{tr}\left[(\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \boldsymbol{\Psi}^{-1})\left(\frac{\partial^2 \boldsymbol{\Psi}}{\partial \theta_i \partial \theta_j} - \boldsymbol{A}_{ij}\right) - \boldsymbol{\alpha}\boldsymbol{\alpha}^T \boldsymbol{A}_{ji}\right], \tag{3.9}$$

where

$$\boldsymbol{A}_{ij} = \frac{\partial \boldsymbol{\Psi}}{\partial \theta_i}\boldsymbol{\Psi}^{-1}\frac{\partial \boldsymbol{\Psi}}{\partial \theta_j}.$$

Note that in the above description, the noise variance σ_ε^2 is also regarded as a hyper-parameter, thus $\boldsymbol{\theta}$ denotes the vector of $(w_1, \nu_0, a_1, \sigma_\varepsilon^2)^T$ for the Gaussian process regression model with covariance kernel (3.1).

The empirical Bayes estimate of $\boldsymbol{\theta}$ is calculated by maximizing the marginal log-likelihood (3.7). For this calculation some iterative optimization methods, such as the conjugate gradient method (see, e.g., Press et al., 2007), can be used. It is noticed that such a method requires the numerical evaluation of $\boldsymbol{\Psi}(\boldsymbol{\theta})^{-1}$, which takes time $O(n^3)$. Efficient implementation is therefore essential for problems with large sample sizes; this will be discussed in Sections 3.3 and 3.4.

Example 3.1. Using the five observations as shown in Figure 3.1, we estimate the values of $\boldsymbol{\theta}$ by the empirical Bayes approach. We used the covariance function with a one-dimensional covariate x, which is given by (3.1). The hyper-parameter is $\boldsymbol{\theta} = (w_1, v_0, a_1, \sigma_\varepsilon^2)^T$. The covariance matrix in (3.7) is

$$\boldsymbol{\Psi}(i,j) = v_0 \exp\left(-\frac{1}{2} w_1 (x_i - x_j)^2\right) + a_1 x_i x_j + \sigma_\varepsilon^2 \delta_{ij}, \quad \text{for } i,j = 1,\ldots,5.$$

Maximizing (3.7) gives the following empirical Bayes estimates:

$$\hat{w}_1 = 16.95,\ \hat{v}_0 = 5.3,\ \hat{a}_1 = 0.6,\ \text{and } \hat{\sigma}_\varepsilon^2 = 0.01.$$

They can therefore be used to calculate the predictive mean and variance using the formulas (2.7) and (2.9), and construct the prediction curve and the predictive intervals as shown in Figure 3.1(b).

For non-Gaussian data, the calculation of the marginal density (3.5) is an intractable and tedious process since it usually involves an n-dimensional integration of which n is the sample size and is usually quite large. We leave this problem to Chapter 7. Some asymptotic properties with regard to empirical Bayes estimates are briefly discussed in Appendix A.6.

3.2 Bayesian inference and MCMC

If we are not sure of the appropriate values for the hyper-parameter $\boldsymbol{\theta}$, we may try to quantify the uncertainty with a probability for $\boldsymbol{\theta}$ by using a second-stage prior distribution (or a *hyper-prior distribution*). Denoting the hyper-prior distribution by $p(\boldsymbol{\theta})$, our knowledge about the parameters now comes from both $p(\boldsymbol{\theta})$ and the observed dataset through the posterior distribution of $\boldsymbol{\theta}$, given by

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta}), \tag{3.10}$$

where $p(\boldsymbol{y}|\boldsymbol{\theta})$ is the marginal distribution of $\boldsymbol{y}$ given in (3.5). For the continuous response with a normal distribution, $p(\boldsymbol{y}|\boldsymbol{\theta})$ has a multivariate normal distribution $N(\mathbf{0}, \boldsymbol{\Psi})$, and thus

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto L(\boldsymbol{\theta}|\mathcal{D}) p(\boldsymbol{\theta}), \tag{3.11}$$

where $L(\boldsymbol{\theta}|\mathcal{D}) = \exp[l(\boldsymbol{\theta}|\mathcal{D})]$, and $l(\boldsymbol{\theta}|\mathcal{D})$ is given by (3.7).

One way to estimate the value of $\boldsymbol{\theta}$ is to find the mode from the posterior density $p(\boldsymbol{\theta}|\mathcal{D})$. This estimate is called the *maximum a posteriori* (MAP) estimate. When a noninformative or a uniform prior distribution is selected, MAP estimate is exactly the same as the empirical Bayes estimates discussed in the previous section.

Example 3.2. In Example 3.1, the hyper-parameters are $\boldsymbol{\theta} = (w_1, v_0, a_1, \sigma_\varepsilon^2)^T$. We can define a hyper-prior distribution for $\boldsymbol{\theta}$ using the following way. The w_1 has an inverse Gamma distribution $w^{-1} \sim Ga(\alpha, \alpha/\mu)$. Note that $\mathrm{E}(w^{-1}) = \mu$, which is the average value of the length scale in (3.1), and $\mathrm{Var}(w^{-1}) = \mu^2/\alpha$, thus small values of α produce vague priors. One example is to set $\mu = 1$ and $\alpha = 0.5$. The priors on a_1 and $\log(\sigma_\varepsilon^2)$ are taken as Gaussian, $N(-3, 3^2)$, corresponding to fairly vague priors, and the prior on $\log(v_0)$ is $N(-1, 1)$. More discussion on how to choose hyper-prior distributions for the squared exponential covariance function can be found in Rasmussen (1996) and Neal (1997).

However, we should be cautious when we use MAP estimates, particularly for the problem with small to medium sample size; this is because the posterior mode may be quite different to the posterior mean by using some kinds of priors such as the inverse Gamma and the log-normal distributions. In this case, it is better to use a fully Bayesian approach which is based on the complete description of "posterior" distribution $p(\boldsymbol{\theta}|\mathcal{D})$ in (3.11). However, in most cases, it is not easy to identify $p(\boldsymbol{\theta}|\mathcal{D})$ in an analytic form. We need to approximate $p(\boldsymbol{\theta}|\mathcal{D})$ by generating samples from it. In order to do this, we resort to numerical schemes such as Markov chain Monte Carlo methods. We explain this computational issue further in the context of the posterior predictive distribution of a new observation in the following way. Let $\boldsymbol{x}^*$ be a new input and let $f(\boldsymbol{x}^*) \stackrel{\mathrm{d}}{=} f^*$ be the related nonlinear function, where '$\stackrel{\mathrm{d}}{=}$' means "denoted by". When $\boldsymbol{\theta}$ is given, the posterior predictive distribution of f^* is also a normal distribution (see discussion in Section 2.1).

$$p(f^*|\mathcal{D}, \boldsymbol{\theta}) \sim N(\mu^*(\boldsymbol{\theta}), \sigma^{*2}(\boldsymbol{\theta})), \tag{3.12}$$

where $\mu^*(\boldsymbol{\theta})$ is given by (2.7) and $\sigma^{*2}(\boldsymbol{\theta})$ is given by (2.9). When $\boldsymbol{\theta}$ is unknown with a hyper-prior distribution assigned, the posterior predictive density is given by

$$p(f^*|\mathcal{D}) = \int p(f^*|\mathcal{D}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}. \tag{3.13}$$

However, this equation is not analytically tractable even for simple hyper-prior distributions such as those used in Example 3.2. In this case, posterior sampling with a suitable Markov chain Monte Carlo method, e.g., *hybrid Monte Carlo*, is a natural choice. We briefly describe the main idea of the posterior sampling method here.

We first generate a set of random variates from the posterior distribution:

$$\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(T)} \sim p(\boldsymbol{\theta}|\mathcal{D}), \tag{3.14}$$

using, for example, a Metropolis-Hastings algorithm or a hybrid Monte Carlo algorithm. Here we usually need a sufficiently large sample size T for accurate approximations. Thus, the posterior density in (3.13) can be approximated by

$$p(f^*|\mathcal{D}) \approx \frac{1}{T}\sum_{t=1}^{T} p(f^*|\mathcal{D},\boldsymbol{\theta}^{(t)}),$$

where $p(f^*|\mathcal{D},\boldsymbol{\theta}^{(t)})$ is the probability density function of a normal distribution $N(\mu^*(\boldsymbol{\theta}^{(t)}),\sigma^{*2}(\boldsymbol{\theta}^{(t)}))$ given in (3.12). Accordingly, if we use the posterior predictive mean of f^* as the prediction of f^*, it can be approximated by

$$\mathrm{E}(f^*|\mathcal{D}) \approx \frac{1}{T}\sum_{t=1}^{T}\mathrm{E}(f^*|\mathcal{D},\boldsymbol{\theta}^{(t)}) = \frac{1}{T}\sum_{t=1}^{T}\mu^*(\boldsymbol{\theta}^{(t)}) \stackrel{\mathrm{d}}{=} \hat{\mu}^*. \tag{3.15}$$

In addition, its variance can be calculated by

$$\begin{aligned}
\mathrm{Var}(f^*|\mathcal{D}) &= \mathrm{E}(f^{*2}|\mathcal{D}) - [\mathrm{E}(f^*|\mathcal{D})]^2 \\
&\approx \frac{1}{T}\sum_{t=1}^{T}\mathrm{E}(f^{*2}|\mathcal{D},\boldsymbol{\theta}^{(t)}) - [\mathrm{E}(f^*|\mathcal{D})]^2 \\
&= \frac{1}{T}\sum_{t=1}^{T}\left[\mathrm{Var}(f^*|\mathcal{D},\boldsymbol{\theta}^{(t)}) + [\mathrm{E}(f^*|\mathcal{D},\boldsymbol{\theta}^{(t)})]^2\right] - \hat{\mu^*}^2 \\
&= \frac{1}{T}\sum_{t=1}^{T}\left[\sigma^{*2}(\boldsymbol{\theta}^{(t)}) + [\mu^*(\boldsymbol{\theta}^{(t)})]^2\right] - \hat{\mu^*}^2.
\end{aligned} \tag{3.16}$$

Other statistics such as the predictive intervals can be calculated similarly.

It is important to note that we are not restricted to using the sampling methods mentioned above; any other sampling algorithms can be used to generate random variates of $\boldsymbol{\theta}$ from its posterior distribution (3.14); see, for example, Gamerman and Lopes (2006). Note that the dimension of $\boldsymbol{\theta}$ is $Q+3$ if the Q-dimensional version of the covariance function in (3.1) is used. The Q is the number of covariates varying from one to a few dozens (or even bigger; see examples discussed in the next chapter). In such a high-dimensional case, posterior sampling would be a computationally intensive work. Moreover, the posterior density function in (3.10) may have a complex form and may be multimodal. It is still a challenging problem to simulate from such a density function. In these regards, Rasmussen (1996) and Neal (1997) suggested that a hybrid Monte Carlo (HMC) method (Duane et al., 1987) is an efficient sampling scheme to generate samples from the above posterior distribution. The details of the hybrid Monte Carlo method are given in Appendix A.4.

3.3 Numerical computation

For functional regression models (3.2) and (3.3) with a continuous response variable and an additive normal noise (3.4), model learning can be achieved by

an empirical Bayes approach or a fully Bayesian approach as discussed in the previous two sections. Although the posterior derivation in Bayesian inference is straightforward in principle, it deals with numerical calculations that involve the inverse of the covariance matrix, $\boldsymbol{\Psi}^{-1}$, which needs to be implemented in both model learning and prediction. The complexity grows at a rate of $O(n^3)$, where n is the number of observations in the training set. This computational complexity limits the application of Gaussian process regression model unless other efficient methods are used. For practical purposes, a rule of thumb is that "no operations that scale greater than $O(n^2)$ will be allowed" (Gibbs and MacKay, 1997). In this section, we describe a few approximation methods.

3.3.1 Nyström method

As discussed in Section 1.2, if $f(\cdot)$ has a Gaussian process prior as defined in (1.10), it can be decomposed by (1.12) so that $f(\boldsymbol{x}) = \sum_{i=1}^{\infty} \phi_i(\boldsymbol{x})\xi_i$, where $\phi_i(\boldsymbol{x})$'s are the eigenfunctions of the covariance function $k(\boldsymbol{x},\boldsymbol{x}')$. The eigenfunctions are p-orthonormal and also satisfy the equation (1.14) or

$$\int k(\boldsymbol{x}',\boldsymbol{x})\phi_i(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \lambda_i\phi_i(\boldsymbol{x}'), \text{ and } \int \phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = \delta_{ij},$$

where $\lambda_1 \geq \lambda_2 \geq \ldots, \geq 0$ are the eigenvalues of the covariance function $k(\boldsymbol{x},\boldsymbol{x}')$.

Now, suppose that we have observed a random sample $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n\}$ from $p(\boldsymbol{x})$. The above equations are approximated by

$$\frac{1}{n}\sum_{h=1}^{n} k(\boldsymbol{x}',\boldsymbol{x}_h)\phi_i(\boldsymbol{x}_h) \approx \lambda_i\phi_i(\boldsymbol{x}'), \tag{3.17}$$

and

$$\frac{1}{n}\sum_{h=1}^{n} \phi_i(\boldsymbol{x}_h)\phi_j(\boldsymbol{x}_h) \approx \delta_{ij}. \tag{3.18}$$

Let $\boldsymbol{K}_n$ be an $n \times n$ covariance matrix (also called a Gram matrix) with elements $K_{ij} = k(\boldsymbol{x}_i,\boldsymbol{x}_j)$ for $i,j = 1,2,\ldots,n$. We can decompose $\boldsymbol{K}_n$ by

$$\boldsymbol{K}_n = \boldsymbol{V}_n\boldsymbol{\Lambda}_n\boldsymbol{V}_n^T, \text{ or } \boldsymbol{K}_n\boldsymbol{V}_n = \boldsymbol{V}_n\boldsymbol{\Lambda}_n, \tag{3.19}$$

where $\boldsymbol{\Lambda}_n = diag(\lambda_1^{(n)},\lambda_2^{(n)},\ldots,\lambda_n^{(n)})$ and $\lambda_1^{(n)} \geq \lambda_2^{(n)} \geq \ldots, \geq \lambda_n^{(n)} \geq 0$ are the eigenvalues of the matrix $\boldsymbol{K}_n$. The $\boldsymbol{V}_n$ is column orthonormal $\boldsymbol{V}_n^T\boldsymbol{V}_n = \boldsymbol{I}_n$. The i-th column of $\boldsymbol{V}_n$ is the eigenvector of $\boldsymbol{K}_n$ corresponding to the eigenvalue $\lambda_i^{(n)}$. By comparing equation (3.17) with (3.19), we see that

$$\phi_i(\boldsymbol{x}_h) \approx \sqrt{n}V_{hi,n}, \text{ and } \lambda_i \approx \frac{\lambda_i^{(n)}}{n}, \text{ for } i = 1,\ldots,n, \tag{3.20}$$

where $V_{hi,n}$ is the (h,i)-th element of $\boldsymbol{V}_n$. This leads to the so-called Nyström approximation to the i-th eigenfunction from (3.17) giving

$$\begin{aligned}\phi_i(\boldsymbol{x}') &\approx \frac{\sqrt{n}}{\lambda_i^{(n)}} \sum_{h=1}^{n} k(\boldsymbol{x}', \boldsymbol{x}_h) V_{hi,n} \\ &= \frac{\sqrt{n}}{\lambda_i^{(n)}} \boldsymbol{K}_n^T(\boldsymbol{x}') \boldsymbol{V}_{i,n}, \end{aligned} \tag{3.21}$$

where $\boldsymbol{K}_n^T(\boldsymbol{x}') = (k(\boldsymbol{x}_1, \boldsymbol{x}'), k(\boldsymbol{x}_2, \boldsymbol{x}'), \ldots, k(\boldsymbol{x}_n, \boldsymbol{x}'))$ and $\boldsymbol{V}_{i,n}$ is the i-th eigenvector of $\boldsymbol{K}_n$. The computational complexity for the Nyström approximation is $O(n^3)$.

Based on a sample of size n, we can only approximate the eigenfunction up to ϕ_n, and thus the decompositions in (1.12) and (1.13) for an infinite dimensional function are approximated as

$$f(\boldsymbol{x}) \approx \sum_{i=1}^{n} \phi_i(\boldsymbol{x}) \xi_i, \tag{3.22}$$

and

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{n} \lambda_i \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{x}'). \tag{3.23}$$

If n is large enough, we can expect these to be good approximations of (1.12) and (1.13). The error caused by expanding an infinite-dimensional function into a feature space of dimension n is called *approximation error*. This topic is beyond the scope of this book; the interested reader is referred to Cucker and Smale (2001) for further details.

We face a dilemma now over how to choose the sample size n. On the one hand, if we want to have a good approximation for the infinite-dimensional function, along with a good empirical Bayes estimation in model learning and a good prediction, we should choose a large sample size n. On the other hand, however, the computational complexity grows at the scale of $O(n^3)$, which is prohibitive if n is large.

We now discuss some methods for efficient implementation. One simple method is based on the above Nyström approximation. The idea is to select a subset of the training data of size m $(< n)$, and then use this subset to approximate the eigenfunction at all n points in (3.20). Let $I_m = \{i_k, k = 1, \ldots, m\}$ be a randomly selected subset from $\{1, \ldots, n\}$, and $\boldsymbol{K}_m$ be the $m \times m$ covariance matrix for the training dataset $\{\boldsymbol{x}_i, i \in I_m\}$; then all the formulas derived around (3.19) and (3.20) can also be applied to $\boldsymbol{K}_m$. Thus, from (3.20) and (3.21), it follows that

$$\lambda_i^{(n)} \approx \frac{n}{m} \lambda_i^{(m)} \quad \text{for } i = 1, \ldots, m, \tag{3.24}$$

and

$$V_{hi,n} \approx \frac{1}{\sqrt{n}}\phi_i(\boldsymbol{x}_h) \approx \frac{1}{\sqrt{n}}\frac{\sqrt{m}}{\lambda_i^{(m)}}\boldsymbol{K}_m^T(\boldsymbol{x}_h)\boldsymbol{V}_{i,m},$$

where $\boldsymbol{K}_m(\boldsymbol{x}_h)$ is a vector of dimension m with element $k(\boldsymbol{x}_h,\boldsymbol{x}_i)$ for $i \in I_m$. It then follows that

$$\boldsymbol{V}_{i,n} \approx \sqrt{\frac{m}{n}}\frac{1}{\lambda_i^{(m)}}\boldsymbol{K}_{n,m}\boldsymbol{V}_{i,m}, \tag{3.25}$$

where $\boldsymbol{K}_{n,m}$ is an $n \times m$ matrix, constructed by a subset of the columns of $\boldsymbol{K}_n$ corresponding to index I_m.

Based on the discussion above, we have the following important results:

$$\begin{aligned} \boldsymbol{K}_n &\approx \boldsymbol{K}_{n,m}\boldsymbol{K}_m^{-1}\boldsymbol{K}_{m,n}, & (3.26)\\ \boldsymbol{\Psi}^{-1} &= (\boldsymbol{K}_n + \sigma_\varepsilon^2\boldsymbol{I}_n)^{-1} \\ &\approx \sigma_v^{-2}\left[\boldsymbol{I}_n - \boldsymbol{K}_{n,m}(\sigma_\varepsilon^2\boldsymbol{K}_m + \boldsymbol{K}_{m,n}\boldsymbol{K}_{n,m})^{-1}\boldsymbol{K}_{m,n}\right]. & (3.27) \end{aligned}$$

The proof of the above two equations is quite straightforward, and can be obtained from (3.19), (3.24), and (3.25),

$$\begin{aligned} \boldsymbol{K}_n &= \sum_{i=1}^{n}\lambda_i^{(n)}\boldsymbol{V}_{i,n}\boldsymbol{V}_{i,n}^T \approx \sum_{i=1}^{m}\lambda_I^{(n)}\boldsymbol{V}_{i,n}\boldsymbol{V}_{i,n}^T \\ &\approx \sum_{i=1}^{m}\left[\left(\frac{n}{m}\lambda_i^{(m)}\right)\left(\sqrt{\frac{m}{n}}\frac{1}{\lambda_i^{(m)}}\boldsymbol{K}_{n,m}\boldsymbol{V}_{i,m}\right)\left(\sqrt{\frac{m}{n}}\frac{1}{\lambda_i^{(m)}}\boldsymbol{K}_{n,m}\boldsymbol{V}_{i,m}\right)^T\right] \\ &= \boldsymbol{K}_{n,m}\left[\sum_{i=1}^{m}(\lambda_i^{(m)})^{-1}\boldsymbol{V}_{i,m}\boldsymbol{V}_{i,m}^T\right]\boldsymbol{K}_{m,n} \\ &= \boldsymbol{K}_{n,m}\boldsymbol{K}_m^{-1}\boldsymbol{K}_{m,n}. \end{aligned}$$

This proves equation (3.26). Equation (3.27) can be proved directly by using formula (A.15) in Appendix A.8.

The Nyström method in the application to kernel machines was introduced by Williams and Seeger (2001), and it is based on the numerical eigendecomposition of a kernel $k(\cdot,\cdot)$ as described before. The computation complexity involved in (3.26) and (3.27) is $O(m^2n)$, which is acceptable in practice if m is much less than n. Williams and Seeger (2001) and Williams et al. (2002) pointed out that the Nyström approximation can be effective when the $(m+1)$-th eigenvalue of $\boldsymbol{K}_n$ is considerably smaller than σ_ε^2.

3.3.2 Active set and sparse greedy approximation

Let $\mathcal{A}$ be a subset of $\{1,2,\ldots,n\}$ with size of m; then $\mathcal{D}_\mathcal{A} = \{(y_i,\boldsymbol{x}_i), i \in \mathcal{A}\}$ is a subset of the training set $\mathcal{D} = \{(y_i,\boldsymbol{x}_i), i = 1,2,\ldots,n\}$. The idea of a sparse

greedy approximation is to use $\mathcal{D}_{\mathcal{A}}$ to replace $\mathcal{D}$. Using this method, the loss of information is unavoidable, but most of them may be kept if a good subset is chosen. The index set $\mathcal{A}$ is called the *active set*. We denote $R = \{1,2,\ldots,n\}\backslash\mathcal{A}$ as the remaining indexes not included in $\mathcal{A}$. The algorithm on how to select an active set is described as follows.

Algorithm 3.1 (Selection of an active set)**.** *The algorithm to select an active set includes the following steps:*

1. *Initialize with an empty set for $\mathcal{A}$ and $R = \{1,2,\ldots,n\}$;*
2. *Based on the current set of $\mathcal{A}$, calculate d_j for each $j \in R$ and select the index, say, i, with the largest value of d_j's;*
3. *Add index i into $\mathcal{A}$ and remove it from R; repeat Step 2;*
4. *Return $\mathcal{A}$ when $\max_j d_j$ is very small or when the size of $\mathcal{A}$ reaches a prefixed value.*

Here, d_j is the criterion on selecting the element of active set, which plays a key role in the algorithm. Many different criteria have been developed. We briefly describe two criteria here; one is based on the Kullback−Leibler divergence, and the other on the concept of entropy.

Specifically, the first method uses *information gain criterion* (Seeger et al., 2003), derived from a Kullback−Leibler divergence. Based on the "current" set of $\mathcal{A}$, we need to decide which of $j \in R$ should be added to $\mathcal{A}$ in the next step. Note that $\mathcal{D}_{\mathcal{A}}$ denotes the training data corresponding to $\mathcal{A}$, and that $\mathcal{D}_{\mathcal{A}+j} = \{(y_i, \boldsymbol{x}_i), i \in \mathcal{A} \cup \{j\}\}$. Now we consider the conditional distribution of f_j=$f(\boldsymbol{x}_j)$ given $\mathcal{D}_{\mathcal{A}}$ and $\mathcal{D}_{\mathcal{A}+j}$, respectively. The information gain criterion d_j is defined by Kullback−Leibler divergence between those two conditional distributions as defined in (2.16),

$$d_j = D(p(f_j|\mathcal{D}_{\mathcal{A}+j})||p(f_j|\mathcal{D}_{\mathcal{A}})). \tag{3.28}$$

The d_j is clearly a measure of the "information gain" by adding extra training data $(y_j, \boldsymbol{x}_j)$ to the current set $\mathcal{D}_{\mathcal{A}}$ in terms of the predictive distribution.

As discussed in Section 2.1, $p(f_j|\mathcal{D}_{\mathcal{A}})$ has a normal distribution

$$p(f_j|\mathcal{D}_{\mathcal{A}}) \sim N(\mu_{\mathcal{A}}, \sigma^2_{\mathcal{A}}), \tag{3.29}$$

where $\mu_{\mathcal{A}}$ and $\sigma^2_{\mathcal{A}}$ are given in (2.7) and (2.8), respectively. Note the fact that

$$\begin{aligned} p(f_j|\mathcal{D}_{\mathcal{A}+j}) &= p(f_j|\mathcal{D}_{\mathcal{A}}, y_j) \\ &\propto p(f_j|\mathcal{D}_{\mathcal{A}})p(y_j|\mathcal{D}_{\mathcal{A}}, f_j) = p(f_j|\mathcal{D}_{\mathcal{A}})p(y_j|f_j) \end{aligned}$$

and $p(y_j|f_j) \sim N(f_j, \sigma^2_\varepsilon)$. This problem is equivalent to a simple Bayesian inference problem, in which f_j has the prior distribution in (3.29). Given the

observation y_j, the "posterior" distribution $p(f_j|\mathcal{D}_{\mathcal{A}+j})$ is still normal with the following mean and the covariance

$$\mu^j_{\mathcal{A}} = \frac{\mu_{\mathcal{A}}/\sigma^2_{\mathcal{A}} + y_j/\sigma^2_\varepsilon}{1/\sigma^2_{\mathcal{A}} + 1/\sigma^2_\varepsilon}, \quad \sigma^{j2}_{\mathcal{A}} = \frac{1}{1/\sigma^2_{\mathcal{A}} + 1/\sigma^2_\varepsilon}. \tag{3.30}$$

Thus d_j is calculated by the Kullback−Leibler divergence for two normal distributions, which results in

$$\begin{aligned} d_j &= D(p(f_j|\mathcal{D}_{\mathcal{A}+j})||p(f_j|\mathcal{D}_{\mathcal{A}})) \\ &= \frac{1}{2}\left[\log\frac{\sigma^2_{\mathcal{A}}}{\sigma^{j2}_{\mathcal{A}}} + \frac{\sigma^{j2}_{\mathcal{A}}}{\sigma^2_{\mathcal{A}}} + \sigma^2_{\mathcal{A}}(\mu^j_{\mathcal{A}} - \mu_{\mathcal{A}})^2 - 1\right] \\ &= \frac{1}{2}\log\left(1 + \frac{\sigma^2_{\mathcal{A}}}{\sigma^2_\varepsilon}\right) + \frac{1}{2}\frac{\sigma^2_{\mathcal{A}}}{\sigma^2_{\mathcal{A}} + \sigma^2_\varepsilon}\left(1 + \frac{(y_j - \mu_{\mathcal{A}})^2}{\sigma^2_{\mathcal{A}} + \sigma^2_\varepsilon}\right). \end{aligned} \tag{3.31}$$

The second criterion is the *differential entropy score* (Lawrence et al., 2003), which is defined by

$$d_J = H(p(f_j|\mathcal{D}_{\mathcal{A}})) - H(p(f_j|\mathcal{D}_{\mathcal{A}+j})),$$

where $H(p)$ is the entropy. For a normal distribution $N(\mu,\sigma^2)$, it is defined as

$$H(p) = \mathrm{E}_p[-\log(p)] = \frac{1}{2}(\log\sigma^2 + \log(2\pi e)).$$

Thus, the differential entropy score is expressed as

$$d_j = \frac{1}{2}\log\left(1 + \frac{\sigma^2_{\mathcal{A}}}{\sigma^2_\varepsilon}\right).$$

By comparing it with (3.31), this measure includes only the first term in information gain criterion.

In practice, we may fix beforehand the value of m, and then use Algorithm 3.1 to select an active set $\mathcal{A}$. The inference for the GPR model is therefore carried out based on $\mathcal{D}_{\mathcal{A}}$ with the size m. Usually, m is much less than n, resulting in an efficient implementation. Further details on how to choose m by other methods and how to measure the information loss can be found in Lawrence et al. (2003); Seeger et al. (2003), and Rasmussen and Williams (2006).

3.3.3 Filtering approach

The idea of the filtering approach is to use a set of transformed data to replace the original observations $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^T$ in $\mathcal{D}$. Suppose that

$\boldsymbol{z} = (z_1, z_2, \ldots, z_r)^T$ is obtained by the following linear transformation with a vector $\boldsymbol{C}_k$,

$$z_k = \boldsymbol{C}_k^T \boldsymbol{y}, \quad \text{for } k = 1, \ldots, r. \tag{3.32}$$

The above transformations define r data filters, and r is usually much less than n. A special case is to construct $\boldsymbol{C}_k$ using the k-th eigenvector of $\boldsymbol{K}_n$, where $\boldsymbol{K}_n$ is the covariance matrix of $\boldsymbol{y}$. The r filtered observations $\boldsymbol{z}$ correspond to the r largest eigenvalues; it can be expressed by $\boldsymbol{z} = \boldsymbol{C}\boldsymbol{y}$ where the k-th row of $\boldsymbol{C}$ is constructed by the k-th eigenvector of $\boldsymbol{K}_n$. Based on the r largest eigenvalues and the eigenvectors, we can approximate the covariance kernel in (3.23) by

$$k(\boldsymbol{x}, \boldsymbol{x}') \approx \sum_{i=1}^{r} \lambda_i \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{x}'). \tag{3.33}$$

Then the subspace spanned by the r-dimensional transformed data contains the "best" r-dimensional view of the original n-dimensional data. If the remaining eigenvalues are very small in comparison, (3.33) should be a good approximation to (3.23). Using eigenvectors to obtain transformed data in (3.32) is actually equivalent to finding the first r principal components, but applied to the observations rather than the usual variables in principal component analysis.

The Nyström method discussed in Section 3.3.1 can be used to find the eigenvectors of $\boldsymbol{K}_n$. We first select a subset of size m as training data, and then use (3.24) and (3.25) to calculate the related eigenvalues and eigenvectors. We then select the first r eigenvalues and eigenvectors to construct the filtered data in (3.32). Here, one problem is how to select m and r. The idea of the filtering approach is to use (3.33) to approximate (3.23). In the extreme case where $\lambda_k = 0$ for all $k > r$, the filtered data are equivalent to the original data in terms of calculating the covariance kernel. This typically does not happen in practice. However, we can compare the values of the eigenvalues and choose r such that the remaining eigenvalues are very small in comparison with the largest eigenvalue.

The problem of choosing m in this approach is relatively less important compared to the other approximation methods discussed previously, although a larger value of m should lead to more accurate approximations of eigenvalues and eigenvectors. One usually needs to learn the eigen-structure just once in the first stage, since it can then be used repeatedly in similar systems. Thus, the computational burden would not increase very much if we select a relatively large value of m. On the other hand, since the eigenvectors are used to generate a "best" r-dimensional view of the original data, the accuracy of the "design" in the first stage would not have much influence on what is carried out in the second stage.

Since the actual observed filtered data may be obtained through physical filters, we now need to consider the observed errors. The observed filtered data

are assumed to be

$$s_k = z_k + e_k, \text{ for } k = 1, \cdots, r,$$

where e_k's are independent and identically distributed as $N(0, \sigma_s^2)$, which is the random error when the filtered data are observed. In matrix form, $\boldsymbol{s} = (s_1, \cdots, s_n)^T$ is distributed as

$$\boldsymbol{s} \sim N(0, \boldsymbol{\Psi}_s), \quad \boldsymbol{\Psi}_s = \boldsymbol{C}\boldsymbol{K}_n\boldsymbol{C}^T + \sigma_s^2\boldsymbol{I}_n,$$

where $\boldsymbol{C}$ is the known matrix as defined before. We still use $\boldsymbol{\theta}$ to denote the unknown parameters in $\boldsymbol{\Psi}_s$, which includes σ_s^2 and the unknown parameters involved in the kernel covariance function. Then the marginal log-likelihood of $\boldsymbol{\theta}$ is given by

$$l(\boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{\Psi}_s| - \frac{1}{2}\boldsymbol{s}^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{s} - \frac{r}{2}\log(2\pi).$$

Maximizing $l(\boldsymbol{\theta})$ leads to an empirical Bayes estimate of $\boldsymbol{\theta}$.

Suppose that we now wish to predict $z^* = \boldsymbol{C}^{*T}\boldsymbol{y}^*$, where $\boldsymbol{y}^* = \boldsymbol{y}(\boldsymbol{X}^*)$ is q-dimensional, and $\boldsymbol{X}^*$ corresponds to the covariates of the q test data points; each column of $\boldsymbol{y}^*$ is associated to one data point. Here, $\boldsymbol{C}^*$ is a known q-dimensional vector, which may correspond to another physical filter. Therefore, the joint distribution of $(\boldsymbol{s}, z^*)$ is the following normal distribution:

$$\begin{pmatrix} \boldsymbol{s}^T \\ z^* \end{pmatrix} \sim N\left(\boldsymbol{0}, \begin{pmatrix} \boldsymbol{\Psi}_s & \boldsymbol{C}\boldsymbol{K}_{n,q}\boldsymbol{C}^* \\ \boldsymbol{C}^{*T}\boldsymbol{K}_{q,n}\boldsymbol{C}^T & \boldsymbol{C}^{*T}\boldsymbol{K}_q\boldsymbol{C}^* \end{pmatrix}\right),$$

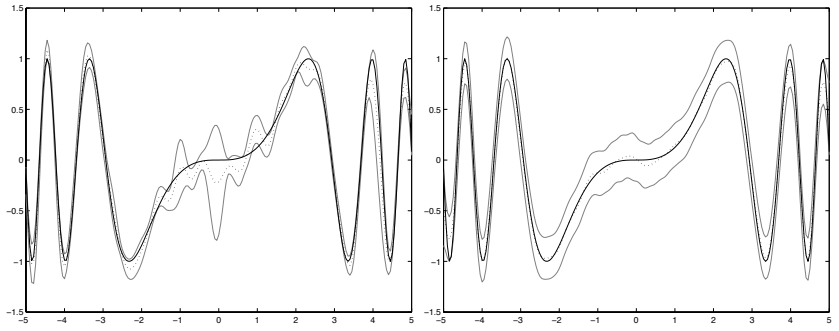
where $\boldsymbol{K}_{q,n} = \left(k(\boldsymbol{x}_i^*, \boldsymbol{x}_j; \hat{\boldsymbol{\theta}})\right)$ is the $q \times n$ covariance matrix between $\boldsymbol{y}^*$ and $\boldsymbol{y}$, evaluated by the covariates of q test data points and n training data points, and $\boldsymbol{K}_q$ is the $q \times q$ covariance matrix of $\boldsymbol{y}^*$. Given the filtered data $\boldsymbol{s}$, the z^* is still normal with the following conditional mean and variance:

$$\begin{aligned} \hat{\mu}^* &= \boldsymbol{C}^{*T}\boldsymbol{K}_{q,n}\boldsymbol{C}^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{s}, \\ \hat{\sigma}^{*2} &= \boldsymbol{C}^{*T}\boldsymbol{K}_q\boldsymbol{C}^* - \boldsymbol{C}^{*T}\boldsymbol{K}_{q,n}\boldsymbol{C}^T\boldsymbol{\Psi}_s^{-1}\boldsymbol{C}\boldsymbol{K}_{n,q}\boldsymbol{C}^*. \end{aligned}$$

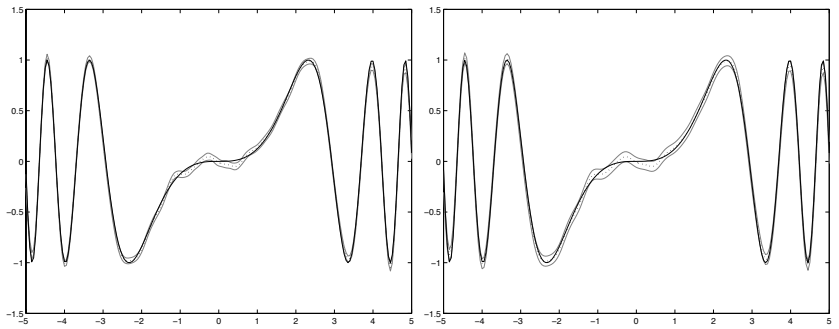
If we want to predict $y^* = y(\boldsymbol{x}^*)$ at a new data point $\boldsymbol{x}^*$, we just need to take $q = 1$ and $K^* = 1$. Further details can be found in Shi et al. (2005b).

Example 3.3. The original $n = 500$ training data are generated from $y_i = \sin((0.5x_i)^3) + \varepsilon_i$, where the ε_i's are independent and identical distributed as $N(0, 0.01^2)$ and $x_i \in (-5, 5)$. Figure 3.2 presents the results when we take $m = 100$. Panel (a) plots the prediction and its 95% prediction intervals as well as the true curves when 100 randomly selected data points are used, while the other panels present similar results but using filtered data with different values of r. The values of the root mean of squared errors (RMSE) between the predictions and their underlying true values are used to compare performance.

Overall, the fit in panels (c) and (d) are much better than those in panels (a) and (b), meaning that using a randomly selected subset of size 100 in panel (a) and using the filtered data of size $r = 39$ in panel (b) result in a significant information loss. However, when the size of filtered data increases to $r = 46$ in panel (c), the performance improves significantly. When we added more filtered data, moving from the case of $r = 46$ to $r = 56$ in panel (d), the performance did not improve further. This indicates that the filtered data of size $r = 46$ includes most of the information of the original training data of size 500.



(a) RMSE=.0695 (b) RMSE=.0611



(c) RMSE=.0293 (d) RMSE=.0291

Figure 3.2 *Simulation study with* $n = 500$ *and* $m = 100$*: plots of the true curves (solid line), the predictions (dotted line), and the* 95% *confidence intervals (light black). The predictions are, respectively, calculated based on (a) 100 random selected data points, and the filtered datasets of size (b)* $r = 39$*, (c)* $r = 46$*, and (d)* $r = 56$*.*

3.4 Further reading and notes

Statistical inference for a Gaussian process regression model is a well-developed topic since the seminal work by O'Hagan (1978), and the basic in-

ferential procedures can be understood in the context of Bayes linear model and multivariate normal theory (see, e.g., Lindley and Smith, 1972). In addition, a large amount of literature has studied the problems from different angles. Instead of giving a comprehensive review on the subject, we have tried to provide the fundamentals on the GPR models by focusing on the estimation problem as well as its computational issues. For additional details, literature reviews, and other computational aspects, readers are referred to Rasmussen and Williams (2006), and are encouraged to visit the Gaussian processes web-site (*http://www.gaussianprocess.org/*).

We discussed some commonly used methods for the GPR models in this chapter. Although most of these methods have been developed in the engineering and machine learning communities where rather different terminologies are used, we have adopted terminologies that are common in statistics both for clarity and familiarity for the target audience.

In pragmatic Bayesian territory, empirical Bayes is a commonly used method on the selection of hyper-parameters, as discussed in the first two sections in this chapter. If the hyper-parameters are treated as tuning parameters in a nonparametric model, then some other methods can be used. One popular choice is to make use of cross-validation. The idea originates from Larson (1931), and has been well developed and applied widely in different fields; for example, see Mosteller and Wallace (1963), Mosteller and Turkey (1968), Stone (1974), and Geisser (1975). The cross-validation approach has been used for hyper-parameter selection in Gaussian process regression model by, e.g., Sundararajan and Keerthi (2001).

An excellent comprehensive discussion on approximation methods for large datasets is given in Chapter 8 of Rasmussen and Williams (2006). Additional references can also be found in the Gaussian processes website as listed before.

Chapter 4

Covariance function and model selection

Model selection is the task of selecting a statistical model from a set of potential models, given the data. It is always an important issue in statistical inference. The methods and theory for Gaussian process regression discussed in this book so far have been built on an assumption that the data come from the true model (1.10) and (1.11), implying that we know in advance which covariance function we should use and which of the covariates the response variable is dependent on. However, in practice, we would always ask questions before we apply any model to a dataset. What data are used? What kind of statistical analysis or model can we use? What model is the best choice if a variety of models can be used? This is the problem of model selection. With regard to Gaussian process regression, there are three major issues: the selection of the covariance function; the selection of covariates which should be included in the covariance kernel function, i.e., variable selection in Gaussian process regression; and the selection of the values of the hyper-parameters. We have discussed the third issue in the last chapter by using an empirical Bayesian approach and other methods, assuming that a particular covariance kernel with a certain set of covariates has already been selected. In this chapter, we focus on the other two issues, i.e., how to select a suitable covariance kernel and how to select covariates on which the response variable is dependent.

Different types of covariance functions along with their main features are discussed in Section 4.1, followed by a discussion of covariance function selection based on Bayes factors in Section 4.2. Automatic relevance determination and a penalized Gaussian process approach are discussed in Section 4.3 to cover the topic of variable selection; some asymptotic properties are also provided. Further comments are left for Section 4.4.

4.1 Examples of covariance functions

In the previous chapters we always used a predetermined squared exponential covariance function as given in (2.4). Actually, a wide variety of other covari-

ance functions can also be used in a Gaussian process regression model. As described previously, since a covariance matrix is generated by a covariance function, a *covariance function* is well defined if it can generate a non-negative definite covariance matrix for any set of points.

Definition 4.1 (Covariance function)**.** *Let $k(\boldsymbol{x},\boldsymbol{x}')$ be a function $k : \mathcal{R}^Q \times \mathcal{R}^Q \to \mathcal{R}$, and $\boldsymbol{x},\boldsymbol{x}' \in \mathcal{R}^Q$, then $k(\boldsymbol{x},\boldsymbol{x}')$ is a valid covariance function if an $n \times n$ matrix $\boldsymbol{K}$, where its (i,j)-th element is given by $k(\boldsymbol{x}_i,\boldsymbol{x}_j)$, is a non-negative definite matrix for any set of data points $\{\boldsymbol{x}_i, i = 1,2,\ldots,n\}$.*

Simple Definition 4.1 for covariance function can be generally applied, and it is known that the class of covariance functions must belong to the class of positive definite functions. Abrahamsen (1997) gave a comprehensive review of covariance and correlation functions for *Gaussian random fields*, a synonym of Gaussian process. In this section, we limit our discussion to some commonly used covariance functions that can be used in Gaussian process regression analysis.

4.1.1 Linear covariance function

A linear covariance function is associated with a linear functional regression model. We consider such a model defined as follows:

$$y(t) = \sum_{q=1}^{Q} x_q(t)\beta_q + \varepsilon(t), \tag{4.1}$$

at $t = t_i$ for $i = 1,2,\ldots,n$. If we assume $\varepsilon(t_i) \overset{\text{iid}}{\sim} N(0,\sigma_\varepsilon^2)$ and also assume an independent Gaussian prior for β_q as $\beta_q \sim N(0,a_q)$, then $y(t_i)$'s in (4.1) have a multivariate normal distribution with zero mean and covariance

$$\text{Cov}(y(t_i),y(t_j)) = \sum_{q=1}^{Q} a_q x_q(t_i)x_q(t_j) + \delta_{ij}\sigma_\varepsilon^2.$$

Thus, the linear regression model in (4.1) can be rewritten as a GPR model with a particular covariance function:

$$y(t) = f(\boldsymbol{x}(t)) + \varepsilon(t), \text{ or } y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon(\boldsymbol{x})$$

where $\boldsymbol{x} = (x_1,\ldots,x_Q)^T$, $f(\cdot) \sim GP(0,k_{lin}(\cdot,\cdot))$ and

$$k_{lin}(\boldsymbol{x},\boldsymbol{x}') = \sum_{q=1}^{Q} a_q x_q x_q'. \tag{4.2}$$

We call $k_{lin}(\cdot,\cdot)$ a *linear covariance function* since it corresponds to the linear functional regression model in (4.1). As we discuss later, the linear covariance function is nonstationary, and is usually used as a part of a covariance kernel combined with other types of covariance functions such as the one in (3.1).

4.1.2 Stationary covariance functions

One of the most frequently used classes is the *stationary covariance function*. This defines a symmetrical covariance function under a transformation, i.e., it is invariant under translations. Further, it is *isotropic* if it is invariant under rotations as well (Abrahamsen, 1997). A stationary covariance function is corresponding to a *stationary process* or a *stationary random field*. Since we always define a constant or zero mean for a Gaussian process, we focus our discussion on the covariance matrix. Readers are referred to Abrahamsen (1997) for the discussion of more general cases.

Definition 4.2 (Stationary covariance function)**.** *A covariance function is stationary if*

$$k(\boldsymbol{x},\boldsymbol{x}') = k(\boldsymbol{\nu}), \quad \textit{where} \quad \boldsymbol{\nu} = \boldsymbol{x} - \boldsymbol{x}', \tag{4.3}$$

for any $\boldsymbol{x},\boldsymbol{x}' \in \mathcal{R}^Q$, i.e., it is invariant under arbitrary translation.

Furthermore, a covariance function is called *isotropy covariance function* if it depends on the distance $||\boldsymbol{\nu}||$ alone. A typical example is to take a squared exponential kernel with an Euclidean or L_2 norm as $||\boldsymbol{\nu}||^2 = \sum_{q=1}^{Q} \nu_q^2$, where ν_q is the q-th element of $\boldsymbol{\nu}$; it results in

$$k_{se}(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}) = k(||\boldsymbol{\nu}||) = v_0 \exp\{-\frac{1}{2} w \sum_{q=1}^{Q} (x_q - x_q')^2\}. \tag{4.4}$$

Here, $\boldsymbol{\theta} = (v_0, w)$ are hyper-parameters. How to choose the value $\boldsymbol{\theta}$ has been discussed in the last chapter. Among isotropic covariance functions, popular choices are the powered exponential, the rational quadratic, and the Matérn covariance functions as listed in Table 4.1.

Table 4.1 *Examples of isotropic covariance functions*

Model	Covariance function, $k(\boldsymbol{x},\boldsymbol{x}') =$
Powered exponential	$v_0 \exp\left(-w||\boldsymbol{x}-\boldsymbol{x}'||^{\gamma}\right), \ 0 < \gamma \leq 2$
Rational quadratic	$\left(1 + s_\alpha w||\boldsymbol{x}-\boldsymbol{x}'||^2\right)^{-\alpha}, \ \alpha, w \geq 0$
Matérn	$\frac{1}{\Gamma(v)2^{v-1}}(w||\boldsymbol{x}-\boldsymbol{x}'||)^{\nu} \mathcal{K}_v(w||\boldsymbol{x}-\boldsymbol{x}'||), \ w \geq 0$ and $\mathcal{K}_v(\cdot)$ is a modified Bessel function of order v

For isotropic covariance functions, there exist sufficient conditions for the sample paths, or the corresponding stochastic process $y(\boldsymbol{x})$, of these covariance functions to be continuously differentiable. A formal statement of the sufficient conditions is given in Theorem A.3, Appendix A.5 based on correlation functions, $\sigma(\boldsymbol{x},\boldsymbol{x}')$, i.e., a standardized version of $k(\boldsymbol{x},\boldsymbol{x}')$. It presents the analytic properties of the sample functions of a real stationary Gaussian process with

obvious extensions to the complex case that includes the vector valued Gaussian process.

A stationary covariance function is *anisotropic* if it depends on $\mathbf{v}$ through a non-Euclidean norm (Abrahamsen, 1997). One example is to extend the Euclidean norm to a more general norm

$$||\mathbf{v}||^2 = \sum_{q=1}^{Q} w_q \mathrm{v}_q^2. \tag{4.5}$$

The generalization of the differentiability theorem for anisotropic and non-stationary covariance functions is given in Theorem A.4, Appendix A.5. For more in-depth details concerning analytic properties such as sample paths continuity and mean square differentiability, interested readers should refer to Abrahamsen (1997).

Anisotropic covariance functions are very useful in practice. For example, if we use the general norm, we can extend the squared exponential covariance kernel defined in (4.4) to the following one:

$$k_{se}(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}) = v_0 \exp\{-\frac{1}{2}\sum_{i=1}^{Q} w_q (x_q - x_q')^2\}, \tag{4.6}$$

where the hyper-parameters are $\boldsymbol{\theta} = (v_0, w_1, \ldots, w_Q)$. Although both covariance functions (4.4) and (4.6) recognize the high correlation between the output of cases with nearby input, (4.6) brings more flexibility. In (4.6), we use the inverse of w_q to measure the length scale for each input covariate. A very large length scale, i.e., a very small value of w_q, means that the corresponding covariate may have little contribution in the covariance function and may therefore be removed from the covariance function and thus excluded from the model. This idea can be applied in model selection as we discuss in the next section.

A Gaussian process with covariance kernel (4.6) can actually be obtained from a regression model based on arbitrary smooth functions (Neal, 1997; MacKay, 1998a). To illustrate this idea, let us take as an example a one-dimensional case with Gaussian radial basis functions.

Example 4.1. Consider a nonlinear regression model $y(x) = f(x) + \varepsilon$ with

$$f(x) = \sum_{h=1}^{H} \xi_h \phi_h(x), \tag{4.7}$$

with H basis functions $\phi_h(x)$'s. We may take a Gaussian radial basis function which is given by

$$\phi_h(x) = \exp\left\{\frac{(x - t_h)^2}{2r^2}\right\},$$

where t_h's are the knots with fixed values. If we assume independent Gaussian prior $\xi_h \overset{\text{ind.}}{\sim} N(0, \sigma_\xi^2)$ for $h = 1, 2, \ldots, H$, and assume further that the knots t_h's are taken with equal space in (t_{min}, t_{max}), we have

$$\begin{aligned} \mathrm{Cov}(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) &= \sigma_\xi^2 \sum_{h=1}^{H} \phi_h(\boldsymbol{x}_i)\phi_h(\boldsymbol{x}_j) \\ &= \frac{\sigma_\xi^2}{\Delta t} \sum_{h=1}^{H} \Delta t \exp\left\{\frac{(x_i - t_h)^2}{2r^2}\right\} \exp\left\{\frac{(x_j - t_h)^2}{2r^2}\right\}, \end{aligned}$$

where $\Delta t = (t_{max} - t_{min})/H$. We take the limit $H \to \infty$ and make σ_ξ^2 scale as $C\Delta t$, then

$$\mathrm{Cov}(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) = C \int_{t_{min}}^{t_{max}} \exp\left\{\frac{(x_i - t)^2}{2r^2}\right\} \exp\left\{\frac{(x_j - t)^2}{2r^2}\right\} dt.$$

If we take (t_{min}, t_{max}) as $(-\infty, \infty)$, it is easy to prove from the above equation that

$$\mathrm{Cov}(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) = \frac{C}{\sqrt{\pi r^2}} \exp\left\{\frac{(x_i - x_j)^2}{4r^2}\right\},$$

which results in a special case of the squared exponential covariance function in (4.6).

A more general exponential covariance function has the form

$$k_{ge}(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}) = v_0 \exp\{-\frac{1}{2} \sum_{q=1}^{Q} w_q |x_q - x_q'|^\gamma\}, \tag{4.8}$$

with $w_q \geq 0$, and $0 < \gamma \leq 2$, sometimes known as *powered exponential covariance function*. For the covariance function to be positive definite, γ must be in the range between 0 and 2 (Neal, 1997). Covariance functions with different values of γ, v_0, and w_q's can be used to model different classes of curves. In particular, the value of γ determines the smoothness of sample paths, such as differentiability.

Figure 4.1 shows different types of sample paths drawn from Gaussian processes with powered exponential covariance functions in one-dimensional cases for different values of γ and the fixed values of $w_q = 10$ and $v_0 = 1$. Note that the covariance kernel with $\gamma = 2$ (i.e., the squared exponential covariance kernel) produces smooth curves, while the covariance kernels with other values of γ produce rough curves; this is because only the curves produced from the squared exponential covariance kernel are infinitely differentiable.

The *rational quadratic covariance function* is another useful class commonly used in Gaussian process regression. The isotropic form is defined in

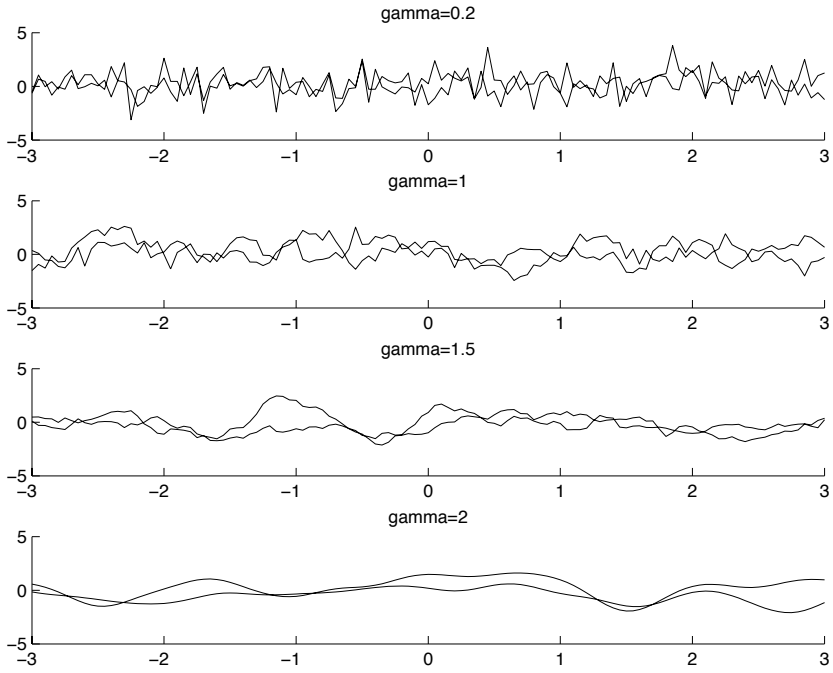


Figure 4.1 *Sample paths drawn from a Gaussian process with the powered exponential covariance function* (4.8) *for one-dimensional covariate, where* $v_0 = 1$, $w_1 = 10$, *and* γ *takes different values as shown in each panel.*

Table 4.1, and an extended form is given by

$$k_{rq}(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}) = \left(1 + s_\alpha \sum_{i=1}^{Q} w_q (x_q - x_q')^2\right)^{-\alpha}, \quad \alpha, w_q \geq 0, \tag{4.9}$$

where s_α is a scaling factor, taking value such as (Abrahamsen, 1997)

$$s_\alpha = 20^{1/\alpha} - 1. \tag{4.10}$$

The above covariance function is also called *Cauchy covariance kernel*. Figure 4.2 shows different types of sample paths drawn from Gaussian processes with the above rational quadratic covariance function in one-dimensional case for different values of α. The value of w_q is fixed as 1 and the scaling factor s_α takes the value in (4.10). The sample paths given in Figure 4.2 look quite similar to those shown in Figure 4.1, although the curves generated from the rational quadratic kernel are smoother than those from the powered exponential covariance kernel. Actually, as shown in Figure 4.2, only very small values of

α would produce some rough curves. When α turns large, the produced curves become indistinguishable with curves produced from the squared exponential covariance kernel.

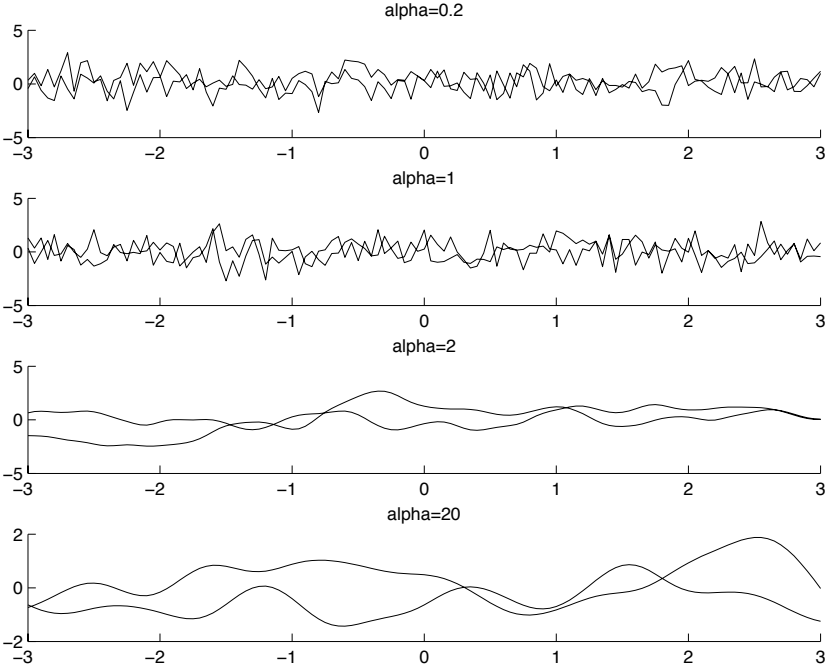


Figure 4.2 *Sample paths drawn from a Gaussian process with general exponential covariance function* (4.9) *for one-dimensional covariate, where* $w_1 = 1$*,* s_α *takes values given in* (4.10)*, and* α *takes different values as shown in each panel.*

Further theoretical details of the Gaussian processes, covariance functions, and their analytic properties can be found in, e.g., Cramer and Leadbetter (1967), Adler (1981), Abrahamsen (1997), Adler and Taylor (2007), and references therein. Note that in particular, sample path properties of familiar isotropic correlation functions are well reviewed in Abrahamsen (1997) among others. Another important characteristic of stochastic processes, in addition to sample path properties, is mean square properties, related to derivatives of the covariance function and moments of the spectral distribution. These properties are not covered in this book. Interested readers can refer to Stein (1999) as well as Abrahamsen (1997) for further details. More examples of stationary covariance functions can also be found in Abrahamsen (1997), MacKay (1998a), and Rasmussen and Williams (2006).

4.1.3 Other covariance functions

The covariance function given in (3.1) consists of two parts. The first part belongs to the class of stationary squared exponential covariance functions, and the second part is a nonstationary linear covariance function. The combination of these two classes can be used to model many different types of smooth curves encountered in practice. This covariance function is actually constructed by using a property that the sum of two covariance functions is still a covariance function. Similar properties can also be derived under multiplication, limits, integration, and convolution. The results are formally stated in Theorem 4.1.

Theorem 4.1. *For* $\boldsymbol{x},\boldsymbol{x}' \in \mathcal{R}^Q$

(i) (Sum of covariance functions), *if* $k_1(\boldsymbol{x},\boldsymbol{x}')$ *and* $k_2(\boldsymbol{x},\boldsymbol{x}')$ *are both covariance functions defined in the same space, then* $a_1k_1(\boldsymbol{x},\boldsymbol{x}')+a_2k_2(\boldsymbol{x},\boldsymbol{x}')$ *is also a covariance function for any* $a_1 \geq 0$ *and* $a_2 \geq 0$*;*

(ii) (Multiplication of covariance functions), *if* $k_1(\boldsymbol{x},\boldsymbol{x}')$ *and* $k_2(\boldsymbol{x},\boldsymbol{x}')$ *are both covariance functions in the same space, then* $[k_1(\boldsymbol{x},\boldsymbol{x}') \cdot k_2(\boldsymbol{x},\boldsymbol{x}')]$ *is also a covariance function;*

(iii) (Limit of covariance functions), *if* $k_i(\boldsymbol{x},\boldsymbol{x}')$ *is a covariance function for* $i = 1,2,\ldots,$ *and* $k(\boldsymbol{x},\boldsymbol{x}') = \lim_{i\to\infty} k_i(\boldsymbol{x},\boldsymbol{x}')$ *exist for all pairs* $\boldsymbol{x},\boldsymbol{x}'$*, then* $k(\boldsymbol{x},\boldsymbol{x}')$ *is also a covariance function;*

(iv) (Integration of a covariance function), *if* $k(\boldsymbol{x},\boldsymbol{x}';a)$ *is a covariance function for all* $a \in A$*, then*

$$k(\boldsymbol{x},\boldsymbol{x}') = \int_A k(\boldsymbol{x},\boldsymbol{x}';a)da$$

is also a covariance function;

(v) (Convolution of covariance functions), *if* $k(\boldsymbol{x},\boldsymbol{x}')$ *is a covariance function, then the one by convolving it with an arbitrary kernel h,*

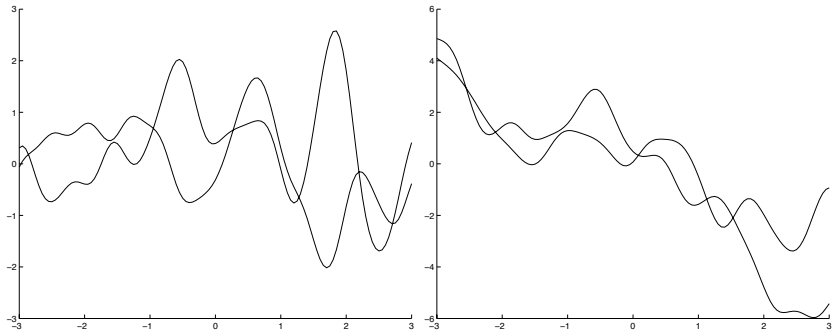
$$\tilde{k}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{x}}') = \int h(\tilde{\boldsymbol{x}},\boldsymbol{x})k(\boldsymbol{x},\boldsymbol{x}')h(\tilde{\boldsymbol{x}}',\boldsymbol{x}')d\boldsymbol{x}d\boldsymbol{x}', \tag{4.11}$$

is still a covariance function.

The proof of the above theorem is similar to the proof of Theorem 3.1 in Abrahamsen (1997) for correlation functions. We omit the details here. The properties in Theorem 4.1 are very useful for constructing new covariance functions or defining new Gaussian processes based on old ones.

The covariance function given in (3.1) is constructed by summing a squared exponential covariance kernel with a linear covariance kernel as given in (4.2). It is one of the most commonly used covariance kernels. In Figure 4.3, the left panel shows curves produced by using a single squared exponential covariance kernel, while the right panel shows the curves produced by using the combined kernel. The latter shows more flexibility and can catch a nonstationary

linear trend. This is particularly useful when it is used, e.g., in multistep-ahead forecasting (see, e.g., Girard and Murray-Smith, 2005; Shi et al., 2007). More examples of combining different types of covariance kernels together can be found in Rasmussen and Williams (2006).



(a) (b)

Figure 4.3 *Curves drawn from a Gaussian process for one-dimensional case with (a) the squared exponential covariance kernel with* $w = 10$, $v_0 = 1$ *and (b) combination of the covariance kernel used in (a) and the linear covariance kernel* (4.2) *with* $a_1 = 2$.

Another example is to construct a new Gaussian process from a white noise process, known as *blurring*.

Example 4.2 (blurring)**.** Let us assume a one-dimensional white noise process, i.e., each random variable in the process is independent and identically distributed with a normal distribution. Thus, the covariance kernel is given by $k(x,x') = \delta(x,x')$ if we further assume a standard normal distribution (i.e., white noise with unit variance). Now, to construct a new covariance function, we use property (v) in Theorem 4.1. We select the following Fourier transform of the *top hat* function as a kernel h in (4.11):

$$h(x,\tilde{x}) = \begin{cases} 1 & \text{if } |x-\tilde{x}| < 1/2 \\ 0 & \text{otherwise} \end{cases}.$$

Then it is easy to see that the new covariance function generated by the convolution in (4.11) is given by

$$\tilde{k}(\tilde{x},\tilde{x}') = \begin{cases} 1-|\tilde{x}-\tilde{x}'| & \text{if } |\tilde{x}-\tilde{x}'| < 1 \\ 0 & \text{otherwise} \end{cases}.$$

This is called *Gaussian blurring*; it is often used in engineering applications such as image analysis. For instance, using Gaussian blurring in image analysis, the original pixel's value has the largest weight value and neighboring pixels have smaller weights as their distance to the original pixel increases.

This can reduce both image noise and image detail, and is a widely used technology in practice. Some other kernels can also be used in blurring (see, for example, MacKay, 1998a; Rasmussen and Williams, 2006).

More discussion on covariance functions, in particular on the use of nonstationary kernels in methodological developments and practical applications, can be found in Sampson and Guttorp (1992), Abrahamsen (1997), MacKay (1998a), Schmidt and O'Hagan (2003), Paciorek and Schervish (2004), and Adams and Stegle (2008).

4.2 Selection of covariance functions

As we discussed in the previous sections, a variety of different covariance functions can be used in a Gaussian process regression model and can therefore be used to model different types of response curves. Empirically, we can select a suitable class of covariance functions based on their features as discussed with sample paths in Section 4.1.2. However, when the dimension of the covariates $\boldsymbol{x}$ is large, it is usually quite difficult to use those empirical guidelines as they are based on empirical study for one- or two-dimensional cases. We therefore need to use certain quantified statistical methods to help us in this stage.

A typical method for model selection from the Bayesian perspective is to use the Bayes factor, defined as the ratio of two marginal likelihoods of two competing models. Thus, we can make use of the Bayes factor as a criterion to determine a suitable covariance function that fits the data. For this purpose, we consider two competing models as two Gaussian process regression models with two different covariance functions, given by

$$y_h = f_h(\boldsymbol{x}) + \varepsilon, \;\; f_h(\cdot) \sim GP_h(0, k_h(\cdot,\cdot)), \;\; h = 1,2. \tag{4.12}$$

We denote the models defined in (4.12) by $\mathcal{M}_h$ for $h = 1,2$. All the unknown parameters involved in $\mathcal{M}_h$ are denoted by $\boldsymbol{\theta}_h$. If we have observed data $\mathcal{D} = \{(y_i, \boldsymbol{x}_i), i = 1,\ldots,n\}$, then

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_N)|\boldsymbol{\theta}_h, \mathcal{M}_h \sim N(\boldsymbol{0}, \boldsymbol{\Psi}_h(\boldsymbol{\theta}_h)),$$

where the (i,j)-th element of $\boldsymbol{\Psi}_h$ is given by

$$\boldsymbol{\Psi}_h(i,j) = \mathrm{Cov}(y_i, y_j|\mathcal{M}_h, \boldsymbol{\theta}_h) = k_h(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}_h) + \sigma_{h,\varepsilon}\delta_{ij}.$$

Note that in this model comparison, two competing model specifications $\mathcal{M}_1$ and $\mathcal{M}_2$ have Gaussian process regression models with different covariance structure k_1 and k_2, respectively; thus the likelihood of $\boldsymbol{y}$ is the marginal likelihood after integrating out the $f_h(\cdot)$ as in (3.7).

The Bayes factor (BF) (Kass and Raftery, 1995) for $\mathcal{M}_1$ and $\mathcal{M}_2$ is defined by the ratio of two marginal distributions,

$$BF_{12} = \frac{Pr(\mathcal{D}|\mathcal{M}_1)}{Pr(\mathcal{D}|\mathcal{M}_2)}, \tag{4.13}$$

where

$$Pr(\mathcal{D}|\mathcal{M}_h) = \int p(\boldsymbol{y}|\boldsymbol{\theta}_h, \mathcal{M}_h) p(\boldsymbol{\theta}_h|\mathcal{M}_h) d\boldsymbol{\theta}_h$$

for $h = 1, 2$, and $p(\boldsymbol{\theta}_h|\mathcal{M}_h)$ is the prior density function of $\boldsymbol{\theta}_h$. Each integrand in (4.13) is therefore the posterior density of $\boldsymbol{\theta}_h$ subject to a normalizing constant. In many cases, the Bayes factor is not analytically tractable unless the assumed model has a simple structure, e.g., normal linear regression with an additive normal error and normal prior distributions. If the integrals involved in the Bayes factor do not work out analytically, the solution will require the calculation of multiple numerical integrations; this is particularly difficult for models with a complex structure such as the Gaussian process regression model and when the dimension of $\boldsymbol{\theta}_h$ is large. Fortunately, some specific Monte Carlo methods have been developed in these regards, which includes Bridge sampling (Meng and Wong, 1996) for computing the marginalizing constants, and Chib's approximation (Chib, 1995) to the marginal likelihood. For a detailed exposition, readers are referred to Chen et al. (2000) and Choi et al. (2010).

Alternatively, one can attempt analytic approximations to the integral in (4.13), and one of the most useful techniques is the Laplace approximation (Tierney and Kadane, 1986). The basic description of the Laplace approximation is given in Appendix A.9. Now, we use a Laplace approximation to the marginal likelihood for the numerator and the denominator, respectively, in (4.13) given by

$$Pr(\mathcal{D}|\mathcal{M}_h) = \int \exp\{l_h^*(\boldsymbol{\theta}_h)\} d\boldsymbol{\theta}_h \approx (2\pi)^{d_h/2} |\boldsymbol{A}_h|^{-1/2} \exp\{l_h^*(\hat{\boldsymbol{\theta}}_h)\}, \quad (4.14)$$

where d_h is the dimension of $\boldsymbol{\theta}_h$ and $l_h^*(\boldsymbol{\theta}_h) = l_h(\boldsymbol{\theta}_h) + \log p(\boldsymbol{\theta}_h)$. Here $l_h(\boldsymbol{\theta}_h)$ is the marginal log-likelihood of $\boldsymbol{\theta}_h$ given by (3.7) as mentioned before, and $\hat{\boldsymbol{\theta}}_h$ is the estimate of $\boldsymbol{\theta}_h$ obtained by maximizing $l_h^*(\boldsymbol{\theta}_h)$, which is actually the MAP estimate as discussed in Section 3.2. The matrix $\boldsymbol{A}_h$ is given by

$$\boldsymbol{A}_h = -D^2 l_h^*|_{\hat{\boldsymbol{\theta}}_h},$$

where $D^2 l_h$ is the Hessian matrix, the second derivatives in terms of $\boldsymbol{\theta}_h$, evaluated at the MAP estimate $\hat{\boldsymbol{\theta}}_h$. The first two derivatives of l_h^* are given in equations (3.8) and (3.9).

The Bayes factor depends on the prior allocated to the model parameters $\boldsymbol{\theta}$. In practice, when this reliance on the prior specification is undesirable, the Bayesian Information Criterion (BIC) (Schwarz, 1978) can be used as an alternative. In addition, the BIC will penalize model complexity and will favor models with a lesser number of parameters. For a Gaussian process regression model, this measure is defined as

$$BIC_h = l_h - \frac{d_h}{2} \log n,$$

where l_h and d_h are defined as before. Actually, BIC is an approximation to $-2\log(BF)$. Further discussion on Bayes factors and their relationship with BIC can be found in Kass and Raftery (1995) and references therein. More discussion on model comparison in Gaussian process regression can be found in Choi et al. (2010).

We now present an example.

Example 4.3. We consider a set of simulated data. The true model used to generate data is $y = 0.6x + 0.5\sin(3x) + \varepsilon$, $x \in [0,6]$. The data are presented as circles in Figure 4.4. Let us consider four different models for this dataset:

$$y = f(x) + \varepsilon, \;\; \varepsilon \sim N(0, \sigma_\varepsilon^2),$$

with

$$\begin{aligned} \mathcal{M}_0 : &\quad f(x) = \beta_0 + \beta_1 x; \\ \mathcal{M}_j : &\quad f(x) \sim GPR(0, k_j(\boldsymbol{\theta}_j)|x), \; j = 1,2,3, \end{aligned}$$

where $\mathcal{M}_0$ is a linear regression model and the other models are Gaussian process regression models. We take the squared exponential covariance kernel (4.6) in model $\mathcal{M}_1$, a combination of the squared exponential covariance kernel with a linear covariance kernel in $\mathcal{M}_2$, and in $\mathcal{M}_3$ we used the sum of three covariance kernels—the two kernels used in $\mathcal{M}_2$ plus the following periodic covariance kernel (Chapter 5, Rasmussen and Williams, 2006):

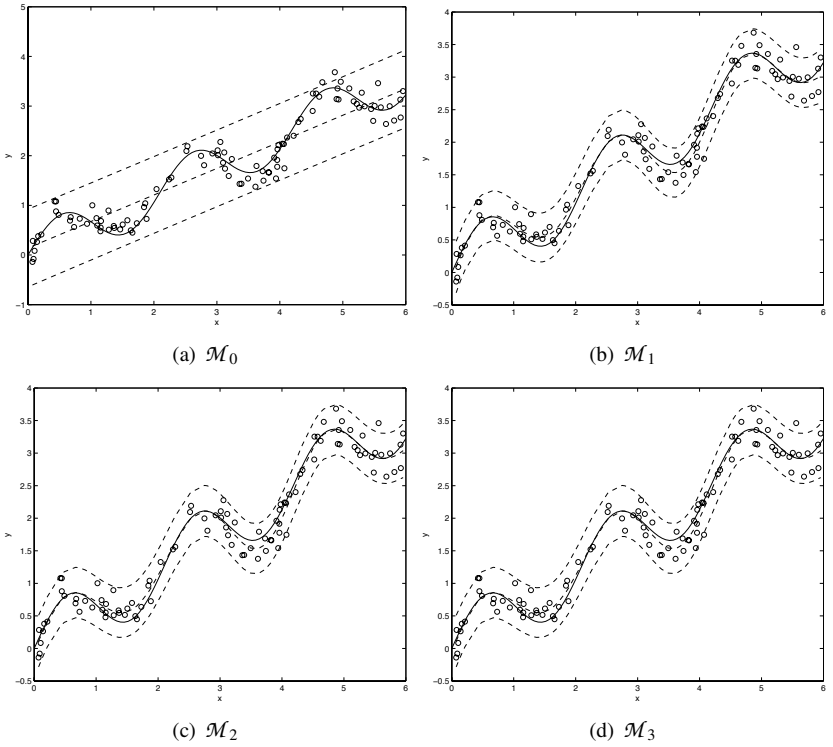
$$k(x,x') = v_1 \exp\left(-\frac{2\sin^2(\pi(x-x'))}{v_2^2} \right).$$

Table 4.2 *Values of Bayes factor (BF) and BIC*

	$\mathcal{M}_0$	$\mathcal{M}_1$	$\mathcal{M}_2$	$\mathcal{M}_3$
BF		$BF_{01} = 1.74 \times 10^{21}$	$BF_{12} = 24.17$	$BF_{23} = 0.22$
BIC	131.05	34.08	33.78	54.81

Using Laplace approximation, we can calculate the values of Bayes factor for each pair of models. The results are presented in Table 4.2. The very large value of BF_{01} indicates that model $\mathcal{M}_1$ is significantly better than model $\mathcal{M}_0$. The value of BF_{12} is also quite large, indicating that model $\mathcal{M}_2$ further improves the curve fitting compared with model $\mathcal{M}_1$. But the value of BF_{23}, which is less than one but is not very far away from one, shows that the difference is not significant when comparing model $\mathcal{M}_2$ with $\mathcal{M}_3$. Thus, we should select model M_2. This conclusion is also supported by using the criterion of BIC; their values are also given in Table 4.2. It shows that model $\mathcal{M}_2$ takes the minimal value of BIC, meaning that $\mathcal{M}_2$ is the best in all four models.

Figure 4.4 shows the true curves as well as the fitted curves obtained by using each of the four models. It is obvious that model M_0 is not acceptable in regard to curve fitting while the other three models give quite a good fitting. Among them, M_2 fits the data best.



(a) $\mathcal{M}_0$ (b) $\mathcal{M}_1$

(c) $\mathcal{M}_2$ (d) $\mathcal{M}_3$

Figure 4.4 *The fitted and true curves: the solid lines stand for true curves, the dashed lines stand for fitted curves and their 95% confidence intervals which are calculated from different models, and the circles stand for the observed data.*

4.3 Variable selection

Although in principle a Gaussian process regression model can deal with covariates of arbitrary dimension, there are several issues to be considered when the dimension of the covariates grows. First, not all candidate covariates are expected to be related to the response variable, and a simple model is always preferred for easy interpretation. Second, more input variables in the model may increase the fit quality, but may result in poor prediction due to the bias-variance trade-off. Computationally, a large number of covariates may result

in a singular Hessian matrix, particularly for models with anisotropic covariance functions such as (4.6) and (4.8); this may cause intractable numerical problems in model learning and prediction calculation.

To select an optimal subset of covariates, one simple way is to compare the models with all possible subsets by using some statistical quantities like the Bayes factor. However, this is impractical even if the number of covariates Q is of moderate size, since the number of all possible subsets is 2^Q for model comparison. Therefore, one needs appropriate methods to select covariates. In this section, we discuss two specific methods for variable selection in Gaussian process regression. The first is called *Automatic Relevance Determination* (ARD) and the second method makes use of penalized techniques.

4.3.1 Automatic relevance determination (ARD)

We now assume that we have observed data $\mathcal{D} = \{(y_i, \boldsymbol{x}_i), i = 1, \ldots, n\}$, and we use a Gaussian process regression model to fit the data

$$y_i = f(\boldsymbol{x}_i) + \varepsilon_i, \;\; f(\cdot) \sim GP(0, k(\cdot,\cdot)), \;\; \varepsilon_i \sim N(0, \sigma_\varepsilon^2). \tag{4.15}$$

If we consider a special covariance function of the squared exponential kernel (4.6), the marginal density function is given as before,

$$\boldsymbol{y}|\boldsymbol{\theta} \sim N(\boldsymbol{0}, \boldsymbol{\Psi}(\boldsymbol{\theta})), \tag{4.16}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)^T$, and the (i, j)-th element of covariance matrix $\boldsymbol{\Psi}_{n\times n}(\boldsymbol{\theta})$ is given by

$$\Psi_{ij}(\boldsymbol{\theta}) = \mathrm{Cov}(y(\boldsymbol{x}_i), y(\boldsymbol{x}_j)) = v_0 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w_q (x_{iq} - x_{jq})^2\right) + \sigma_\varepsilon^2 \delta_{ij}, \tag{4.17}$$

where $\boldsymbol{\theta} = (w_1, \ldots, w_Q, v_0, \sigma_\varepsilon^2)^T$ indicates the hyper-parameters involved in the covariance function.

As we discussed in Section 4.1.2, in the squared exponential covariance function (4.17), regression coefficients $(w_1, \ldots, w_Q)$ acting as characteristic length-scales can determine the relevance of the corresponding covariates to the response variable. The larger the value of the w_q, the more relevant the corresponding covariates are. Small values of w_q indicate the corresponding covariate might be a nuisance factor. The idea of automatic relevance determination (Neal, 1996; MacKay, 1998b) is to simply remove those covariates with small values of w_q from the model.

We have discussed empirical Bayes estimates and MAP estimates for the GPR model in Sections 3.1 and 3.2. The estimates of $\boldsymbol{\theta}$ and their standard errors can be calculated. Regarding variable selection, the ARD method is concerned with the estimation of w_q and its standard deviation. If both are close to zero,

or the absolute value of $\hat{w}_q$ is less than a *threshold*, the corresponding covariates can be excluded from the model. However, it is not an easy job to find a good *threshold*. It depends on the scale of each x although empirically we can standardize x into zero mean and unit standard deviation. We also need to select the threshold such that the selected variables and log-likelihood are not susceptible to small changes in this threshold. When the number of covariates Q is very large and when the covariates are highly correlated, selecting an accurate threshold is rather difficult. Some empirical study and examples can be found in, e.g., Neal (1996), Williams and Rasmussen (1996), MacKay (1998b), Burden (2000), and Chen and Martin (2009).

A Bayesian ARD is based on the Bayesian learning method discussed in Section 3.2. We can select variables based on the posterior distribution of w_q. If both the posterior mean and posterior variance of w_q are close to zero, we may exclude the corresponding covariate from the model. However, the selection of threshold remains a challenging problem.

4.3.2 Penalized Gaussian process regression

In contrast to the ARD method, a penalized technique is a more flexible and efficient method on variable selection. Simply speaking, in this penalized framework, we adopt a commonly used penalized technique to Gaussian process regression model, referring to it as a *penalized Gaussian process regression* method. We also start the discussion from a GPR model with the squared exponential kernel (4.6). Given a set of data $\mathcal{D}$, the model is given in (4.16) and (4.17). The basic idea of a penalized technique is in a sense analogous to the ARD method; that is, we focus on the regression coefficient w_q. Small value of w_q may result in the removal of the q-th covariate from the model. However, instead of finding a threshold as in the ARD method, penalized technique would force the estimates of w_q for those irrelevant covariates to be zeros. Thus, the irrelevant covariates are removed from the model automatically and only the relevant covariates are kept in the model.

We can link the penalized technique to the empirical Bayesian approach as discussed in Section 3.1. Let us recall the marginal log-likelihood in (3.7),

$$l_n(\boldsymbol{\theta}|\mathcal{D}) = -\frac{1}{2}\log|\boldsymbol{\Psi}(\boldsymbol{\theta})| - \frac{1}{2}\boldsymbol{y}^T\boldsymbol{\Psi}(\boldsymbol{\theta})^{-1}\boldsymbol{y} - \frac{n}{2}\log 2\pi.$$

The empirical Bayesian approach estimates $\boldsymbol{\theta}$ by maximizing the above log-likelihood. The idea of penalized technique is to penalize regression coefficients w_q's by adding a penalty term $P_{\lambda_n}(w_q)$ to the log-likelihood. The penalized likelihood of the GPR model can therefore be defined as

$$l_p(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\lambda}_n) = -\frac{1}{n} l_n(\boldsymbol{\theta}|\mathcal{D}) + \sum_{q=1}^{Q} P_{\lambda_n}(w_q). \tag{4.18}$$

Different types of penalty function can be used, for example, the Ridge penalty given by

$$P_{\lambda_n}(w_q) = \lambda_n w_q^2;$$

it is also called L_2 penalty. The most notable one may be the LASSO penalty (Tibshirani, 1996) given by

$$P_{\lambda_n}(w_q) = \lambda_n |w_q|;$$

this is also called L_1 penalty. The Bridge penalty (Frank and Friedman, 1993; Fu, 1998) is given by

$$P_{\lambda}(w_q) = \lambda_n |w_q|^{\gamma}, \;\; 0 < \gamma < 1,$$

where λ_n and γ are parameters to be estimated. Some other commonly used penalty functions include the SCAD penalty (Fan and Li, 2001) and the adaptive LASSO penalty (Zou, 2006). The former is defined as

$$P_{\lambda_n,a}(w_q) = \begin{cases} \lambda_n w_q, & \text{if } 0 \le w_q \le \lambda_n, \\ -\frac{w_q^2 - 2a\lambda_n w_q + \lambda_n^2}{2(a-1)}, & \text{if } \lambda_n < w_q \le a\lambda_n, \\ \frac{(a+1)\lambda_n^2}{2}, & \text{if } w_q > a\lambda_n; \end{cases} \tag{4.19}$$

and the latter is defined by

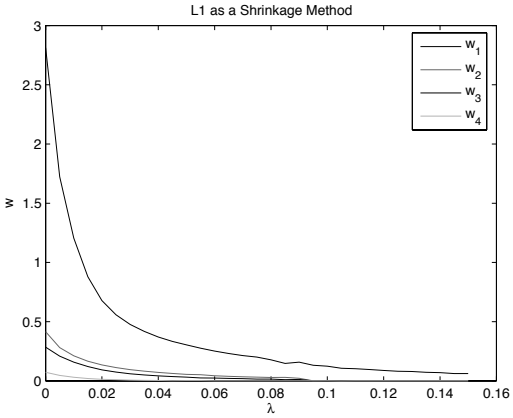
$$P_{\lambda_n,a}(w_q) = \omega_q \lambda_n |w_q|, \tag{4.20}$$

where ω_q is a weight. In this penalty function, we can take, for example, $\omega_q = 1/\hat{w}_q^{\gamma}$, where γ is another tuning parameter and $\hat{w}_q$ is the related empirical Bayes estimate. Both the SCAD and the adaptive LASSO penalties involve two parameters.

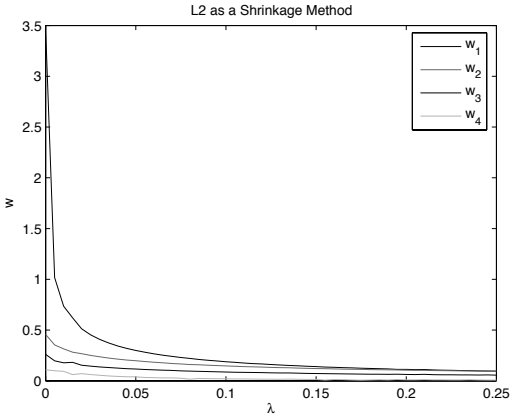
Since the parameter w_q is non-negative, if we use a LASSO penalty function, the estimate from the penalized Gaussian process regression model can be expressed as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} l_p(\boldsymbol{\theta}; \mathcal{D}, \lambda_n) = \arg\min_{\boldsymbol{\theta}} \left[-\frac{1}{n} l_n(\boldsymbol{\theta}|\mathcal{D}) + \lambda_n \sum_{q=1}^{Q} w_q \right]. \tag{4.21}$$

To find the constraint optimum solution from the above equation, λ_n acts as a regularizer or tuning parameter. As the value of λ_n increases from zero, the values of $\hat{w}_q$'s in (4.21) begin to shrink. When λ_n reaches a certain value, one or more of $\hat{w}_q$'s will tend toward zero. When λ_n is larger than a certain value, all $\hat{w}_q$'s become zeros. This can be illustrated from Figure 4.5, which is based on simulated data where the empirical Bayes estimates of the hyper-parameters are $(\hat{w}_1, \hat{w}_2, \hat{w}_3, \hat{w}_4) = (2.812, 0.416, 0.285, 0.0727)$. As can be seen from panel (a) in Figure 4.5 where a LASSO penalty is used, when $\lambda_n = 0.065$, w_4 is



(a)



(b)

Figure 4.5 *Penalized estimates of* $\hat{w}_q$ *for* $q = 1, \ldots, 4$ *(from top to bottom) against regularizer parameter* λ_n *using (a) LASSO and (b) Ridge penalties, respectively.*

shrunk to zero. As λ_n continues to increase, w_3 and w_2 are set to zero one by one. Actually, only w_1 is not equal to zero in the true model.

We therefore need to find an optimum λ_n where some of the hyperparameters are set to zero, and some are still kept in the model. One commonly used approach to select an optimal regularizer parameter is the generalized cross-validation (GCV). Since "leave-one-out" cross-validation (CV) is computationally too costly, we can use a k-fold GCV method (see, e.g., Hastie and Tibshirani, 1990; McLachlan et al., 2004). The idea is to divide the training

samples into k equal (or nearly equal) groups. For each group, use the data from the other $k-1$ groups to calculate the penalized estimates from (4.21) for each given value of λ_n. The penalized estimates $\hat{\boldsymbol{\theta}}$ are used to calculate the predictions for all the data points in this group. The value of the RMSE (root mean of squared errors) between the predictions and the observed values of the response variable can be calculated by

$$RMSE_j^2 = \frac{1}{n_j} \sum_{i \in G_j} (y_i - \hat{y}_i)^2, \tag{4.22}$$

where $\{i \in G_j\}$ corresponds to the data points in group j of sample size n_j, and y_i and $\hat{y}_i$ are the observation and its prediction, respectively. The prediction is calculated using (2.7). We repeat the procedure for each group. The value of $GCV(\lambda_n)$ is the average of $RMSE_j$'s for all $j = 1, \ldots, k$. We then find the optimal λ_n by minimizing $GCV(\lambda_n)$.

Once we obtain the optimal λ_n, we can calculate the penalized estimates $\hat{\boldsymbol{\theta}}$. All the variables with $\hat{w}_q = 0$ are removed from the model automatically.

The procedure can be extended to any other penalty functions in a similar way. But for the Bridge penalty, there are two regularizer parameters λ_n and γ. We therefore need to calculate values of GCV for pairs of (λ_n, γ). This obviously involves a much heavier computational burden than the case with only one regularizer. The same problem exists for the SCAD and the adaptive LASSO penalties. This requires efficient methods. Some more discussion on this issue will be given in Section 4.4.

In panel (b) of Figure 4.5, we use the ridge penalty, an L_2 penalty. We note that none of the estimates is equal to zero no matter what value of the regularizer λ_n takes, and thus, the ridge penalty cannot be used for variable selection. The will be further discussed in Section 4.3.4.

In some problems, the correlation between covariates may be very high; in other problems, covariates may be grouped naturally. In those cases, we may want to select groups of variables. This can be achieved by combining more than one penalty function. For example, elastic net makes use of both the LASSO penalty and the Ridge penalty (Zou and Hastie, 2005). Further details are given in, for example, Yuan and Lin (2006), Wang et al. (2009), and Yi (2009).

If we applied the general norm in (4.5) to other stationary covariance functions, as is the case with the square exponential in (4.8), the regression coefficients $(w_1, \ldots, w_Q)$ would still be acting as characteristic length-scales; then, they would determine the relevance of the corresponding covariates in the response variable. We could still apply this penalized framework to those regression coefficients and then select variables using the same procedure as described above.

A comprehensive study of penalized Gaussian process regression models with different types of penalty functions can be found in Yi (2009) and Yi et al.

(2011). Their study shows that the adaptive LASSO and the Bridge penalties usually achieve better performance than others in penalized Gaussian process regression models, although there are exceptions. In practice, we should try different penalty functions and select the one with the best performance.

4.3.3 Examples

We provide two examples in this section to illustrate the penalized Gaussian process regression framework we have just discussed.

Example 4.4 (Paraplegia data)**.** We come back to the example with paraplegia data that we discussed in Section 1.3.1. In this example, we used the data recorded in the fourth and fifth standing-up for patient "bj". There are 480 and 345 observations in the two sets, respectively. We randomly select 20 samples from each set, and use them as training data for model learning and variable selection. Since the data include dependent response variables, and highly correlated input variables, we encountered problems in model learning when we used all 33 covariates in the GPR model with the covariance function (4.17). Notice that the iterative method used in the empirical Bayesian approach cannot achieve convergence due to a nearly singular Hessian matrix. The same problem is encountered when we use the penalized GPR with the SCAD penalty. However, the other penalized methods work well. The results presented in Table 4.3 are based on the optimal regularizer parameters selected by a fourfold GCV. It shows that the penalized Gaussian process regression with the Bridge penalty performs the best in terms of the RMSE and sparsity (i.e., the number of covariates selected, the smaller, the better). Thus, we can conclude from these results that only five variables need to be included in the model.

The adaptive LASSO (denoted by AdLASSO in Table 4.3), as it turns out, does not perform well in this example although it usually works better than other penalties in penalized Gaussian process regression. We conjecture that this may be because the best weight vector should be constructed using the estimates of the regression coefficients, but for this dataset, estimates using the empirical Bayesian approach are not available. We have to use the Ridge estimator instead, which unfortunately brings a bias to the weight vector.

Example 4.5 (Meat data)**.** Near-infrared spectroscopy (NIRS) is a technology that is widely used in the pharmaceutical industry, medical diagnostics, food and agrochemical quality control, and combustion research. The molar absorptivity of different substances varies at different near-infrared regions of the electromagnetic spectrum (from about 800 nm to 2500 nm). The transmittance at each wavelength is an observation of a covariate, and therefore, NIRS data always have a very large number of covariates. The covariates with near wavelength are extremely highly correlated.

This example concerns how to use NIRS technology to detect meat fat. For

Table 4.3 *Variable selection results for paraplegia data*

	RMSE	No. of variables selected	Optimal values of tuning parameters
GPR(Ridge)	12.581	N/A	$\lambda_n = 0.01$
GPR(LASSO)	12.158	11	$\lambda_n = 0.00002$
GPR(Bridge)	9.609	5	$\gamma = 0.01$, $\lambda_n = 0.8$
GPR(AdLASSO)	78.894	2	$\gamma = 0.5$, $\lambda_n = 0.08$

Table 4.4 *Variable selection results for meat data*

	RMSE	Number of variables selected
GPR(MLE)	0.890	All
GPR(Ridge)	0.711	All
GPR(LASSO)	0.649	26
GPR(Bridge)	0.432	4
GPR(SCAD)	0.530	15
GPR(Adaptive LASSO)	0.390	3

each finely chopped pure meat sample, a 100-channel spectrum of absorbances (covariates) was recorded on a Tecator Infratec Food and Feed Analyzer working in the wavelength range 850 to 1050, where the absorbance is "$-\log_{10}$" of the transmittance measured by the spectrometer. The fat contents (response variable) were also detected using laboratory methods. We then need to find a model to infer the response variable by using the covariates. The data are available from *http://lib.stat.cmu.edu/datasets/tecator*.

The data are standardized such that the response variable has zero mean and all the input variables have zero means and unit variances. The dataset is split into a training dataset of 172 samples and a test dataset of 43 samples. Since the data show a significantly nonlinear pattern, we should use a nonlinear regression model. In this example, we use the Gaussian process regression model, while different penalty functions are used for variable selection. The results given in Table 4.4 are based on the optimal regularizer parameters chosen by a fourfold GCV. In this example, the Adaptive LASSO and the Bridge penalized GPR model, using only 3 and 4 input variables, respectively, out of 100 candidate variables, achieved the best results in terms of RMSE. In addition, these results have also shown very important practical implications: not only we can record much less data, but also the model with fewer covariates increases the prediction accuracy.

4.3.4 Asymptotics

Penalized techniques have been widely applied to different regression models on variable selections due to their efficiency and their decent theoretical properties (see, e.g., Fan and Li, 2001). As discussed in the previous subsections, penalized methods are also efficient on selecting variables for Gaussian process regression models (see also Yi et al., 2011). We now discuss some asymptotic properties of the penalized GPR model.

Recall that penalty functions $P_{\lambda_n}(w_q)$ in (4.18) play a key role on variable selection using penalization. Thus, we begin our discussion with the analytic aspects of penalty functions. Although many different functions can be used as penalty functions, they have to satisfy some conditions. First of all, they should be non-negative and satisfy

$$P_{\lambda_n}(w_q) \geq 0 \text{ and } P_{\lambda_n}(0) = 0. \tag{4.23}$$

It also should satisfy the following condition:

$$P_{\lambda_n}(w_q^*) \geq P_{\lambda_n}(w_q) \text{ if } |w_q^*| \geq |w_q|, \tag{4.24}$$

which implies that $P_{\lambda_n}(w_q)$ penalizes larger regression coefficients no less than the smaller ones.

In order to discuss the asymptotic properties for penalized Gaussian process regression models, we need to use the asymptotic properties of the empirical Bayes estimates $\boldsymbol{\theta}$ from (3.7). As proved in Appendix A.6, the empirical Bayes estimate $\hat{\boldsymbol{\theta}}$ is an r_n consistent estimate that is specified in equation (A.6).

Before discussing the main asymptotic results, we need to define some notations. Let $\boldsymbol{\theta}_0$ be the true value of the hyper-parameters $\boldsymbol{\theta}$; for example, it is $\boldsymbol{\theta}_0 = (w_1^{(0)}, \ldots, w_Q^{(0)}, v_0^{(0)}, \sigma_\varepsilon^{(0)2})^T$ for the squared exponential covariance function as defined in (4.17). Let

$$\mathcal{A} = \{q : w_q^{(0)} \neq 0\} \text{ and } \mathcal{B} = \{q : w_q^{(0)} = 0\},$$

so that $\boldsymbol{\theta}_0 = (\boldsymbol{w}_{\mathcal{A}}^{(0)T}, \boldsymbol{w}_{\mathcal{B}}^{(0)T}, v_0^{(0)}, \sigma_\varepsilon^{(0)2})^T$. This implies that the true model includes only a subset of input variables corresponding to set $\mathcal{A}$, with the rest of the input variables regarded as nuisance factors or noise in the model. Note that the cardinality of $\mathcal{A}$, $|\mathcal{A}| = d \leq Q$. That is, d is the number of nonzero values among $(w_1^{(0)}, \ldots, w_Q^{(0)})^T$. Furthermore, we denote $\hat{\boldsymbol{\theta}}_n$ as the parameter estimator that is obtained by minimizing the penalized minus log likelihood (4.18).

In addition, we assume all the second-order derivatives of the penalty functions $P_{\lambda_n}(w_q)$ are continuous at $\boldsymbol{w}_{\mathcal{A}}^{(0)}$. Now we state two theorems that establish consistency in parameter estimation and model sparsity in variable selection. If we define

$$a_n = \max\left\{P'_{\lambda_n}(w_q^{(0)}) : q \in \mathcal{A}\right\}, \tag{4.25}$$

$$b_n = \max\left\{P''_{\lambda_n}(w_q^{(0)}) : q \in \mathcal{A}\right\}, \tag{4.26}$$

we have the following theorem.

Theorem 4.2. *Let $p^n_{\boldsymbol{\theta}}$ denote the joint probability density of $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^n$ that satisfies regularity conditions $(C1)$–$(C4)$ as specified in Appendix A.6. Also, assume conditions in (4.23) and (4.24) hold for the penalty function P_{λ_n}. Suppose now that the empirical Bayes estimate of $\boldsymbol{\theta}$ from (3.7) is r_n consistent, and b_n in (4.26) converges to 0, then there exists a local minimizer $\hat{\boldsymbol{\theta}}_n$ of $l_p(\boldsymbol{\theta})$ in (4.18) such that $\|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}\| = O_p(r_n^{-1} + a_n)$.*

The proof is given in Appendix A.7. Note that Theorem 4.2 establishes the consistency for parameter estimation. When the sample size goes to infinity, the nonzero regression coefficients retained in the model should converge to the same set of true nonzero regression coefficients. At the same time, parameter estimates of zero regression coefficients converge to zero as sample size goes to infinity, thus all the zero coefficients are excluded from the final model.

We now discuss whether the resulting penalized Gaussian process regression estimator can asymptotically perform as well as the one assuming that $w_{\mathcal{B}}^{(0)} = \mathbf{0}$ was known in advance. Such property is commonly referred to as the *oracle property* of estimators (see, e.g., Donoho and Johnstone, 1994; Fan and Li, 2001; Zou, 2006). The sparsity of the oracle estimator based on the penalized Gaussian process regression model is given by the following theorem.

Theorem 4.3 (Sparsity)**.** *Let $\hat{\boldsymbol{\theta}}_n = [\hat{\boldsymbol{w}}_{\mathcal{A}}^T, \hat{\boldsymbol{w}}_{\mathcal{B}}^T, \hat{v_0}, \hat{\sigma}_\varepsilon^2]^T$ be the local optimizer of $l_p(\boldsymbol{\theta})$ in Theorem 4.2. Assume the same conditions as defined in Theorem 4.2 and the following additional conditions are held.*

1. $\liminf_{n\to\infty} \liminf_{\theta\to 0_+} \frac{1}{\lambda_n} \frac{\partial P_{\lambda_n}(\hat{\boldsymbol{\theta}})}{\partial w_q} > 0,$

2. $\lambda_n \to 0$ *and* $\frac{n\lambda_n}{r_n} \to \infty$ *as* $n \to \infty$.

Then, with probability tending to 1, model sparsity can be achieved, i.e.,

$$\lim_{n\to\infty} P(\hat{\boldsymbol{w}}_{\mathcal{B}} = \mathbf{0}) = 1.$$

In the above theorem, "liminf" means "limit inferior," the infimum of the limit points. The proof is given in Appendix A.7.

Theorem 4.2 implies that by choosing proper penalty functions P_{λ_n} and regularizer parameters λ_n, we can get a consistent estimator for penalized Gaussian process regression model. Theorem 4.3 indicates that by choosing proper penalty functions P_{λ_n} and regularizer parameters λ_n, the penalized GPR estimator can also be consistent in terms of variable selection by achieving an oracle property.

For the SCAD penalty, we have $\lim_{\lambda_n\to 0} a_n = 0$. Thus, from Theorem 4.3, we see that when $n\lambda_n / r_n \to \infty$, the estimator of the penalized Gaussian process regression model with SCAD penalty can achieve both estimate consistency and sparsity (oracle property). For the Bridge penalty L_γ, $0 < \gamma < 1$, when

the true values of the significant input variables are large enough, we have $\lim_{\lambda_n \to 0} a_n \to 0$. Thus, the Bridge penalized estimators for Gaussian process regression model also achieve the estimate consistency and sparsity.

For the penalized Gaussian process regression model with adaptive LASSO, the key point for Theorem 4.3 is that the weights $(\omega_1, \ldots, \omega_Q)$ depend on the data. As the sample size goes to infinity, the weights for the insignificant predictor variables converge to zero, while the weights for significant predictors converge to a constant value. Therefore, the large regression coefficients can be unbiased estimates and, at the same time, the zero-coefficient input variables can be removed. As a result, oracle properties hold for the penalized Gaussian process regression model with Adaptive LASSO.

4.4 Further reading and notes

We have discussed some basic properties of covariance functions in Section 4.1, under the assumption that the mean of the Gaussian process prior is a constant, specifically a zero function. The discussion for a general case, for example, a stationary Gaussian process with a nonconstant mean, can be found in Abrahamsen (1997). More examples, such as the Matérn class of covariance functions and its special case, the Ornstein-Uhlenbeck process (a popular class used in engineering), can be found in Rasmussen and Williams (2006) and MacKay (1998a) and the references therein.

Bayes factor and other model selection criteria such as BIC are considered for selecting a suitable covariance function in Section 4.2. The calculation of the Bayes factor is difficult for the GPR models when the dimension of $\boldsymbol{\theta}$ is large, and the Laplace approximation and some Monte Carlo methods can be used as discussed in Section 4.2. Some other methods have also been developed, for example, the integrated nested Laplace approximation (Rue et al., 2009) and nested sampling (Skilling, 2006). Asymptotic properties of the Bayes factor for the GPR model and further discussion can also be found in Choi et al. (2010).

Variable selection has been widely investigated and well developed in statistics for many decades. Penalized techniques became more popular after the LASSO was proposed by Tibshirani (1996). Some more developments include the group Lasso (Yuan and Lin, 2006), the Dantzig selector (Zou, 2006; Candes and Tao, 2007) and Bayesian LASSO (Park and Casella, 2008). Some theoretical properties of penalized methods can be found in, e.g., van de Geer (2008) and Fan and Li (2001). Other methods for variable selection include, for example, L_1 minimization (see, e.g., Candes and Plan, 2007) and sure independence screening (Fan and Lv, 2003). These methods and other variants of penalization techniques are becoming more important and challenging, particularly for dealing with the case of high-dimensional covariates.

In the GPR models, although ARD can be used as an empirical method for

variable selection, it is difficult to find the accurate values of thresholds. The Penalized technique seems to be a more efficient method (Yi, 2009; Yi et al., 2011). Some other penalized GPR methods based on other techniques such as those mentioned above may also be useful, but we have yet to wait for more research in this direction.

Chapter 5

Functional regression analysis

So far, we have discussed Gaussian process regression models for a single curve (which can also be regarded as a batch with only one realization). We now turn our attention to consider batch data or data with repeated curves. A typical example of batch data is the paraplegia data discussed in Section 1.3.1, in which we want to model a standing-up maneuver. The response variable is the vertical trajectory of the body's COM that is a continuous functional variable, but there are two types of covariates. One type of covariate is for functional variables such as forces and torques under the patient's feet in the period of standing-up maneuver, and the other type is for scalar variables such as the patient's weight and height. The data collected from repeated experiments form what we call batch data. This is a typical functional regression problem with functional response variable as described in Section 1.1.2.

In general, regression analysis encompasses any techniques to model a response variable by a set of covariates and analyzes their relationships. Thus, functional regression analysis specifically refers to the area of regression analysis that deals with the situation where either the response variable or covariates are functional variables. Ramsay and Silverman (2005) covered thoroughly the many different types of functional regression analysis models. Among them, in this chapter we focus on the problem that the response variable $y(t)$ is a continuous functional variable. However, the covariates may include both functional and scalar variables. The problem with discrete functional response variables—mainly restricted to cases from exponential family distributions—is discussed in Chapter 7. A general functional regression model was briefly formulated in Section 1.1; in this chapter we pay attention to a specific case that considers simultaneously the covariance and the mean structure of the data. In particular, the covariance structure is modeled by a GPR model and the mean structure is modeled by a functional regression model. Logically, we refer to this model as a Gaussian process functional regression (GPFR) model (Shi et al., 2007).

This chapter is organized as follows. As an introduction, Section 5.1 describes a simple linear functional regression model—a model including mean part only. Section 5.2 describes the details of the Gaussian process functional

regression model and Section 5.3 discusses its implementation. GPFR models with mixed-effect structure and with ANOVA structure are discussed in Sections 5.4 and 5.5, respectively.

5.1 Linear functional regression model

We first consider a special case in which response variable is functional while the covariates are scalar. Let $y_m(t)$ be a functional response variable for replications $m = 1,\ldots,M$ and let $\boldsymbol{u}_m = (u_{m1}, u_{m2}, \ldots, u_{mp})^T$ be a set of scalar covariates. A linear functional regression model is defined by (Ramsay and Silverman, 2005):

$$y_m(t) = \mu_m(t) + \varepsilon_m(t) \text{ and } \mu_m(t) = \boldsymbol{u}_m^T \boldsymbol{\beta}(t), \tag{5.1}$$

where $\varepsilon_m(t)$'s are i.i.d. random errors and $\boldsymbol{\beta}(t)$ is a p-dimensional vector of "time-varying" coefficients depending on t. Here, since both $y_m(t)$ and $\boldsymbol{\beta}(t)$ are functional variables, they can be approximated by using a set of basis functions, which are

$$y_m(t) \approx \tilde{y}_m(t) = \boldsymbol{A}_m^T \boldsymbol{\Phi}(t) \text{ and } \boldsymbol{\beta}(t) \approx \boldsymbol{B}^T \boldsymbol{\Phi}(t), \tag{5.2}$$

where $\boldsymbol{\Phi}(t) = (\phi_1(t), \ldots, \phi_H(t))^T$ are a set of H basis functions, $\boldsymbol{A}_m$ is an H-dimensional coefficient vector, and $\boldsymbol{B}$ is a $H \times p$ matrix. For example, $\boldsymbol{\Phi}(t)$ could be a set of B-spline basis functions (see, for example, Rice and Silverman, 1991; Faraway, 1997, 2001; Ramsay and Silverman, 2005). A simple way to estimate the coefficients $\boldsymbol{A}_m$ is to minimize the following objective function:

$$\int ||y_m(t) - \sum_{h=1}^{H} A_{mh}\phi_h(t)||^2 dt. \tag{5.3}$$

If we have observed $y_m(t)$ at points $\{t_{mi}, i = 1,\ldots,n_m\}$ for $m = 1,\ldots,M$, denoted by $\boldsymbol{y}_m = (y_{m1}, \ldots, y_{mn_m})^T$, the minimization of the above objective function is therefore approximately equivalent to estimate $\boldsymbol{A}_m$ by minimizing the following,

$$(\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{A}_m)^T (\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{A}_m),$$

where $\boldsymbol{\Phi}_m$ is an $n_m \times H$ matrix with elements $(\phi_h(t_{mi}))$. In this way, we can obtain the following estimates:

$$\hat{\boldsymbol{A}}_m = (\boldsymbol{\Phi}_m^T \boldsymbol{\Phi}_m)^{-1} \boldsymbol{\Phi}_m^T \boldsymbol{y}_m. \tag{5.4}$$

Similarly, we can estimate $\boldsymbol{B}$ using a least squares criterion, i.e., minimizing the following objective function:

$$\int ||(\boldsymbol{A}\boldsymbol{\Phi}(t) - \boldsymbol{U}\boldsymbol{B}^T \boldsymbol{\Phi}(t))||^2 dt, \tag{5.5}$$

where $\boldsymbol{A}$ is an $M \times H$ matrix formed by $\boldsymbol{A}_m$ row-wise, and $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M)^T$ is an $M \times p$ matrix. As an approximation, we may simply minimize $||\boldsymbol{A} - \boldsymbol{U}\boldsymbol{B}^T||^2$. Consequently, the estimate of $\boldsymbol{B}$ is given by

$$\hat{\boldsymbol{B}}^T = (\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{U}^T\boldsymbol{A}. \tag{5.6}$$

Here, $\boldsymbol{A}$ can be replaced by the estimates given in (5.4). Further discussion on how to select the form and number of the basis functions can be found in, e.g., Faraway (1997, 2001).

More generally, we can consider a roughness penalty in the objective functions (5.3) and (5.5) (see, e.g., Ramsay and Silverman, 2005). This will be discussed in Section 5.3.

In the next section, we will discuss a nonparametric functional regression problem with both functional and scalar covariates.

5.2 Gaussian process functional regression model

Assume that we have a functional response variable $y_m(t)$ for $m = 1, 2, \ldots, M$, and we also have a set of functional covariates $\boldsymbol{x}_m(t)$ and a set of scalar covariates $\boldsymbol{u}_m$, where

$$\boldsymbol{x}_m(t) = (x_{m1}(t), x_{m2}(t), \ldots, x_{mQ}(t))^T \text{ and } \boldsymbol{u}_m = (u_{m1}, u_{m2}, \ldots, u_{mp})^T.$$

Applying the general model in (1.1) to the m-th batch, we have

$$y_m(t) = f(t, \boldsymbol{x}_m(t), \boldsymbol{u}_m) + \varepsilon_m(t), \tag{5.7}$$

where $\varepsilon_m(t)$'s are random errors that are independent at different t's. Here, we indistinctively refer to this m-th *replication* as m-th batch or m-th *curve* if we need to emphasize its different aspects.

A Gaussian process functional regression (GPFR) model is defined by

$$y_m(t) = \mu_m(t) + \tau_m(\boldsymbol{x}_m) + \varepsilon_m(t), \tag{5.8}$$

where $\mu_m(t)$ models the common mean structure across different curves, while $\tau_m(\boldsymbol{x}_m)$ defines the covariance structure of $y_m(t)$ for the different data points within the same curve. We use a Gaussian process regression model to define the covariance structure:

$$\tau_m(\boldsymbol{x}_m) \sim GPR_m[0, k_m(\boldsymbol{\theta}_m)|\boldsymbol{x}_m], \quad m = 1, \ldots, M, \tag{5.9}$$

where $GPR_m[0, k_m(\boldsymbol{\theta}_m)|\boldsymbol{x}_m]$, as defined in (1.11), denotes a Gaussian process regression model with a covariance function k_m and hyper-parameters $\boldsymbol{\theta}_m$. Equations (5.8) and (5.9) jointly define a Gaussian process functional regression model; it is denoted by

$$y_m(t) \sim GPFR[\mu_m(t), k_m(\boldsymbol{\theta}_m)|\boldsymbol{x}_m(t), \boldsymbol{u}_m].$$

Here the covariance kernel $k_m(\boldsymbol{\theta}_m)$ depends on the functional covariates $\boldsymbol{x}_m(t)$, while the mean part $\mu_m(t)$ depends on scalar covariates $\boldsymbol{u}_m$ or $\boldsymbol{x}_m(t)$ as well. This is indeed a very flexible model as it combines the mean and the covariance structure of the functional relationships. For example, if we have little prior knowledge about the physical relationship between the response variable $y(t)$ and the functional covariates $\{x_{m1}(t), x_{m2}(t), \ldots, x_{mQ}(t)\}$, we can use the GPR model to model the regression relationship nonparametrically. However, if we have already had some information that the regression relationship between $y(t)$ and some of the functional covariates can be described by a parametric model, we can always incorporate this parametric part to model (5.8); see examples in Sections 5.4 and 5.5.

Another important feature of the GPFR model is that it reveals both model structures from the data collected from all batches (or subjects), and predicts each individual curve based on the common mean structure and the data collected from that particular individual. This can largely improve the accuracy of the prediction for each individual curve, and is particularly useful in applications such as the construction of individual dose-response curves as discussed in Section 5.4. A detailed discussion of this feature is given in Section 5.3.2.

In functional regression analysis, all the functional variables in the same batch are usually observed at the same data points $\{t_{mi}, i = 1, \ldots, n_m\}$ for $m = 1, \ldots, M$. Let us recall the notation given in (1.2) for the data available in each batch:

$$\mathcal{D}_m = \{(y_{mi}, t_{mi}, x_{m1i}, \ldots, x_{mQi}) \text{ for } i = 1, \ldots, n_m; \text{ and } (u_{m1}, \ldots, u_{mp})\}, \quad (5.10)$$

where $y_{mi} = y(t_{mi})$ is the observation of $y_m(t)$ at t_{mi} and $x_{mqi} = x_q(t_{mi})$ is the measurement of the q-th input variable for $q = 1, \ldots, Q$. Likewise, the observations of the scalar variables for the m-th batch are given by $\boldsymbol{u}_m = (u_{m1}, \ldots, u_{mp})^T$. If any of the functional variables have not been recorded at the same data points, they can still be used in the analysis but they need to be treated beforehand; see Ramsay and Silverman (2005) for further details.

The mean model $\mu_m(t)$ in (5.8) stands for the common mean structure across all M batches. Either parametric or nonparametric models can be adopted here, depending on our prior knowledge about the physical system. One example is the linear functional regression model defined in (5.1). Several other specific cases are discussed in the remainder of this chapter.

5.3 GPFR model with a linear functional mean model

In this section, we consider a special case of a Gaussian process functional regression model by using a linear functional mean model. We assume that the mean model $\mu_m(t)$ depends on the scalar covariates $\boldsymbol{u}_m$ and t only. A linear functional regression model has been defined by (5.1) in Section 5.1, which is $\mu_m(t) = \boldsymbol{u}_m^T \boldsymbol{\beta}(t)$. Thus, the Gaussian process functional regression model with

a linear functional regression mean is given by

$$y_m(t) = \boldsymbol{u}_m^T \boldsymbol{\beta}(t) + \tau_m(\boldsymbol{x}_m) + \varepsilon_m(t). \tag{5.11}$$

As discussed in (5.2), both $y_m(t)$ and $\boldsymbol{\beta}(t)$ can be approximated by using a set of basis functions. We recall them here:

$$y_m(t) \approx \tilde{y}_m(t) = \boldsymbol{A}_m^T \boldsymbol{\Phi}(t), \text{ and } \boldsymbol{\beta}(t) \approx \boldsymbol{B}^T \boldsymbol{\Phi}(t).$$

Based on the data $\{\mathcal{D}_m\}$ given in (5.10), it is possible to evaluate the likelihood for the model (5.11), but the direct computation of the estimates for all unknown parameters becomes tedious.

Alternatively, we may use a two-stage approach as suggested in Shi et al. (2007), which estimates the mean structure and the covariance structure sequentially. In the first stage we estimate the mean structure. To estimate the coefficients $\boldsymbol{A}_m$, instead of using a simple objection function as given in (5.3), we consider a general one which is defined as follows:

$$\int ||y_m(t) - \sum_{h=1}^{H} A_{mh}\phi_h(t)||^2 dt + \lambda Pen(\tilde{y}_m), \tag{5.12}$$

where $\tilde{y}_m$ is given in (5.2) and $Pen(\tilde{y}_m)$ is a *roughness penalty* (see, e.g., Ramsay and Silverman, 2005). For instance, one choice of the roughness penalty can be based on the second derivative, which is given by

$$Pen(\tilde{y}_m) = \int [D^2 \tilde{y}_m(t)]^2 dt = \int [D^2 \boldsymbol{A}_m^T \boldsymbol{\Phi}(t)]^2 dt = \boldsymbol{A}_m^T \boldsymbol{R} \boldsymbol{A}_m,$$

where $D^2 \tilde{y}_m(t)$ is the second derivative of $\tilde{y}_m(t)$ in terms of t, and the $H \times H$ matrix $\boldsymbol{R}$ is

$$\boldsymbol{R} = \int D^2 \boldsymbol{\Phi}(t) D^2 \boldsymbol{\Phi}^T(t) dt. \tag{5.13}$$

Once the basis functions are selected, we can calculate $\boldsymbol{R}$ directly. For example, if we use B-spline smoothing, we can choose B-spline basis functions and calculate their second derivatives. $\boldsymbol{R}$ is therefore calculated from the integration in (5.13). The B-spline basis functions are a popular choice of spline functions in functional data analysis, and have a wide applicability with the help of common statistical software packages such as *R* and MATLAB.

The coefficients $\boldsymbol{A}_m$ are estimated by minimizing the penalized objection function (5.12). Based on the data $\mathcal{D}_m$, this minimization procedure is equivalent to minimize the following:

$$(\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{A}_m)^T \boldsymbol{\Psi}_m (\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{A}_m) + \lambda \boldsymbol{A}_m^T \boldsymbol{R} \boldsymbol{A}_m,$$

where $\boldsymbol{\Psi}_m$ is an $n_m \times n_m$ covariance matrix calculated from the covariance function $k_m(\boldsymbol{\theta}_m)$ and the other quantities have been defined in Section 5.1. In this way, we can obtain the following estimates:

$$\hat{\boldsymbol{A}}_m = (\boldsymbol{\Phi}_m^T \boldsymbol{\Psi}_m \boldsymbol{\Phi}_m + \lambda \boldsymbol{R})^{-1} \boldsymbol{\Phi}_m^T \boldsymbol{\Psi}_m \boldsymbol{y}_m. \tag{5.14}$$

The smoothing parameter λ involved in (5.14) can be chosen by a cross-validation or generalized cross-validation method; see the detailed discussion in Chapter 5 in Ramsay and Silverman (2005).

We next need to estimate the coefficients $\boldsymbol{B}$ based on the whole dataset $\mathcal{D} = \{\mathcal{D}_m, m = 1,2,\ldots,M\}$. Similar to (5.12), we can use a penalized least squares criterion

$$\int (\boldsymbol{A}\boldsymbol{\Phi}(t) - \boldsymbol{U}\boldsymbol{B}^T\boldsymbol{\Phi}(t))^T(\boldsymbol{A}\boldsymbol{\Phi}(t) - \boldsymbol{U}\boldsymbol{B}^T\boldsymbol{\Phi}(t))dt + \lambda Pen(\boldsymbol{\beta}), \tag{5.15}$$

where $Pen(\boldsymbol{\beta})$ is a roughness penalty for $\boldsymbol{\beta}$. If we carry on using the one based on the second derivatives, we have the following penalty which is analogous to the one in (5.12) for $\tilde{y}_m$.

$$Pen(\boldsymbol{\beta}) = \mathrm{tr}(\boldsymbol{B}^T\boldsymbol{R}\boldsymbol{B}),$$

where $\mathrm{tr}(\cdot)$ represents the trace of a matrix, i.e., the sum of all diagonal elements, and $\boldsymbol{R}$ is given by (5.13). The penalized least squares criterion in (5.15) can therefore be re-expressed by

$$\mathrm{tr}(\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{J}) + \mathrm{tr}(\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{B}^T\boldsymbol{J}\boldsymbol{B}) - 2\mathrm{tr}(\boldsymbol{B}^T\boldsymbol{J}\boldsymbol{A}^T\boldsymbol{U}) + \lambda\mathrm{tr}(\boldsymbol{B}^T\boldsymbol{R}\boldsymbol{B}), \tag{5.16}$$

where

$$\boldsymbol{J}_{H\times H} = \int \boldsymbol{\Phi}(t)\boldsymbol{\Phi}^T(t)dt.$$

Differentiating the objective function in (5.16) with respect to the matrix $\boldsymbol{B}$ and setting it to zero, we obtain the following equations:

$$(\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{B}^T\boldsymbol{J} + \lambda\boldsymbol{B}^T\boldsymbol{R}) = \boldsymbol{U}^T\boldsymbol{A}\boldsymbol{J}.$$

The solution of the above equations provides the estimate of $\boldsymbol{B}$, which is given by

$$\mathrm{vec}(\hat{\boldsymbol{B}}^T) = [\boldsymbol{J}\otimes(\boldsymbol{U}^T\boldsymbol{U}) + \lambda\boldsymbol{R}\otimes\boldsymbol{I}]^{-1}\mathrm{vec}(\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{J}), \tag{5.17}$$

where $\mathrm{vec}(\boldsymbol{B})$ denotes the vector of length $p\times H$ formed by stacking the columns of the matrix $\boldsymbol{B}$, and $\otimes$ denotes the Kronecker product of two matrices (see the definition in, for example, Appendix A.6 in Ramsay and Silverman, 2005). Note that the smoothing parameter λ in (5.17) needs to be distinguished from the smoothing parameter in equation (5.14); although we use the same notation, it can also be selected by using cross-validation or generalized cross-validation methods; see the details in Ramsay and Silverman (2005).

The second stage is to estimate the parameters involved in the covariance structure in (5.9). We replace $\boldsymbol{\beta}(t)$ in equation (5.11) by its estimate $\hat{\boldsymbol{\beta}}(t) = \hat{\boldsymbol{B}}^T\boldsymbol{\Phi}(t)$ from the first stage. Let $\tilde{\tau}_m(t) = y_m(t) - \boldsymbol{u}_m^T\hat{\boldsymbol{B}}^T\boldsymbol{\Phi}(t)$. Then, we have

$$\tilde{\tau}_m(t) = \tau_m(\boldsymbol{x}_m) + \varepsilon_m(t), \tag{5.18}$$

where $\tau_m(\boldsymbol{x}_m)$ has a Gaussian process regression model as specified in (5.9). Thus, based on the dataset $\mathcal{D}_m$ and the estimates of the mean, it can be understood that we have the following observations:

$$\tilde{\tau}_{mi} = y_{mi} - \boldsymbol{u}_m^T \hat{\boldsymbol{B}}^T \boldsymbol{\Phi}(t_{mi}).$$

From (5.18) and (5.9), we have

$$\tilde{\boldsymbol{\tau}}_m = (\tilde{\tau}_{m1}, \tilde{\tau}_{m2}, \ldots, \tilde{\tau}_{mn_m})^T \sim N(\boldsymbol{0}, \boldsymbol{\Psi}_m),$$

where the (i,j)-th element of $\boldsymbol{\Psi}_m$ is given by

$$\Psi_{ij} = k_m(\boldsymbol{x}_{mi}, \boldsymbol{x}_{mj}; \boldsymbol{\theta}_m) + \sigma_\varepsilon^2 \delta_{ij},$$

with $\boldsymbol{x}_{mi}^T = (x_{mi1}, x_{mi2}, \ldots, x_{imQ})$. By treating (5.18) as a Gaussian process regression model, we can therefore use the methods discussed in Chapter 3 to estimate the values of parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_m, m = 1, \ldots, M\}$. Here, $\boldsymbol{\theta}_m$ denotes all parameters involved in $\boldsymbol{\Psi}_m$ including σ_ε. As an example, the empirical Bayes estimate of $\boldsymbol{\Theta}$ can be calculated by maximizing the following marginal likelihood:

$$l(\boldsymbol{\Theta}|\mathcal{D}) = \sum l_m(\boldsymbol{\Theta}|\mathcal{D}), \tag{5.19}$$

where

$$l_m(\boldsymbol{\Theta}|\mathcal{D}) = -\frac{1}{2}\log|\boldsymbol{\Psi}_m(\boldsymbol{\theta}_m)| - \frac{1}{2}\tilde{\boldsymbol{\tau}}_m^T \boldsymbol{\Psi}_m(\boldsymbol{\theta}_m)^{-1}\tilde{\boldsymbol{\tau}}_m - \frac{n}{2}\log 2\pi.$$

However, the covariance matrix or the weight matrix $\boldsymbol{\Psi}_m$ used in (5.14) depends on $\boldsymbol{\theta}_m$, which is unknown before we estimate it in the second stage. Consequently, we need an iterative algorithm as follows.

Algorithm 5.1 (Empirical Bayesian learning for the GPFR model with a linear functional mean model)**.** *An iterative method includes the following steps:*

1. *Start with an initial value of* $\boldsymbol{\theta}_m$*, and calculate* $\boldsymbol{\Psi}_m$*;*
2. *Calculate the estimates* $\hat{\boldsymbol{A}}_m$ *by (5.14) and* $\hat{\boldsymbol{B}}$ *by (5.17);*
3. *Calculate the value of hyper-parameter* $\boldsymbol{\Theta}$ *by maximizing (5.19);*
4. *Repeat steps 2 and 3 until the iteration converges.*

The final estimates of $\boldsymbol{A}_m$, $\boldsymbol{B}$, and $\boldsymbol{\Theta}$ in this iterative approach are obtained when the algorithm satisfies a suitable convergence criterion.

In practice, if our main purpose is to calculate prediction, and the accuracy is not the first priority, we may use a fast approximating approach. This approach removes the roughness penalties in (5.12) and (5.15) and ignores the covariance structure in (5.14), and therefore results in the estimate of coefficients $\boldsymbol{A}_m$ given by (5.4) and the estimate of $\boldsymbol{B}$ given by (5.6) in Section 5.1. Consequently, $\hat{\boldsymbol{\Theta}}$ can be calculated directly by maximizing the marginal likelihood given in (5.19). This approximation does not need to use the iterative

Algorithm 5.1 to update $(\boldsymbol{A},\boldsymbol{B})$ and $\boldsymbol{\Theta}$; in turn, it is therefore expected to be computationally very efficient and convenient. These kinds of fast approximating methods are particularly useful for the problems involved in online learning and monitoring where computation speed is a necessity.

5.3.1 Prediction

We now consider how to calculate the prediction $y^* = y(t^*)$ at a new point $(t^*, \boldsymbol{x}^*, \boldsymbol{u}^*)$ with $\boldsymbol{x}^* = \boldsymbol{x}(t^*)$. From (5.11) and (5.2), the mean is estimated by

$$\hat{\mu}(t) = \boldsymbol{u}^T \hat{\boldsymbol{B}}^T \boldsymbol{\Phi}(t), \tag{5.20}$$

and the prediction of y^* includes two parts,

$$\hat{y}^* = \hat{\mu}(t^*) + \hat{\tau}(\boldsymbol{x}^*), \tag{5.21}$$

where $\tau^* = \tau(\boldsymbol{x}^*)$ is predicted by its conditional mean $\mathrm{E}(\tau^*|\mathcal{D})$ of the Gaussian process regression model defined in (5.9). The formulae are given in (2.7) and (2.8) for a single curve. In the case of batch data, however, we are dealing with repeated curves with the peculiarity that a GPR model is assumed for each one of them. The way we proceed from here about predicting τ^* (or y^*) depends on whether we have information available for the curve (i.e., whether we have observed some data). Let us recall the dose-response example in Section 1.3.2: there are two kinds of patients, the ones who have visited the hospital before for whom we have built a medical file and have recorded some of their data, and the new patients who come to the hospital for the first time. Future predictions for the first kind of patients can be treated similarly to the case for single curves, i.e., τ^* can be calculated by using formulae similar to (2.7) and (2.8). For complete new patients, however, no previous information is available and new methods for prediction are needed.

We first suppose that we have already observed some data for a batch and want to predict $y(t^*)$ at a new point, denoting it as the $(M+1)$-th curve or batch. In addition to the training data observed in the first M batches, assume that n observations have also been obtained in the new curve at $\boldsymbol{t} = (t_1, t_2, \ldots, t_n)^T$, providing the data

$$\mathcal{D}_{M+1} = \{(y_{M+1,i}, t_{M+1,i}, x_{M+1,1,i}, \ldots, x_{M+1,Q,i}),\ i = 1, \ldots, n;\ \boldsymbol{u}_{M+1}\}.$$

Thus the training data are $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_M, \mathcal{D}_{M+1}\}$, which is used to estimate the parameters using the methods discussed earlier. To predict y^* at a new data point t^*, we assume that y^* and the observed data $\{y_{M+1,i}, i = 1, \ldots, n\}$ have the same model (5.11), and thus τ^* and $\{\tau_{mi}, i = 1, \ldots, n\}$ have the same GPR model structure. From (5.18), let $\tilde{\tau}(t) = y(t) - \hat{\mu}(t)$, and we have

$$(\tilde{\tau}_{M+1,1}, \ldots, \tilde{\tau}_{M+1,n}, \tilde{\tau}^*)^T \sim N(0,\ \boldsymbol{\Omega}), \tag{5.22}$$

where $\boldsymbol{\Omega}$ is an $(n+1)\times(n+1)$ covariance matrix, defined by

$$\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{\Psi} & \boldsymbol{\Psi}(\boldsymbol{x}^*, \boldsymbol{x}_{M+1}) \\ \boldsymbol{\Psi}^T(\boldsymbol{x}^*, \boldsymbol{x}_{M+1}) & \Psi(\boldsymbol{x}^*, \boldsymbol{x}^*) \end{bmatrix} \tag{5.23}$$

where $\boldsymbol{\Psi}(\boldsymbol{x}^*, \boldsymbol{x}_{M+1}) = (\Psi(\boldsymbol{x}^*, \boldsymbol{x}_{M+1,1}), \ldots, \Psi(\boldsymbol{x}^*, \boldsymbol{x}_{M+1,n}))^T$ is the covariance vector between y^* and $\boldsymbol{y}_{M+1} = (y_{M+1,1}, \ldots, y_{M+1,n})^T$, and $\boldsymbol{\Psi}$ is the $n \times n$ covariance matrix of $\boldsymbol{y}_{M+1}$ or $\tilde{\boldsymbol{\tau}}_{M+1}$, which depends on $\boldsymbol{x}_{M+1}$. Note that the covariance is calculated by the covariance function defined in (5.9). Since we assumed a joint normal distribution in (5.22), the predictive distribution of y^*, given the training data $\mathcal{D}$ and the mean $\boldsymbol{\mu}(t)$, is also a Gaussian distribution. Using equations (2.7) and (2.8), we obtain its mean and variance as follows:

$$\begin{aligned} \mathrm{E}(y^*|\mathcal{D}, \boldsymbol{\mu}) &= \mu_{M+1}(t^*) + \boldsymbol{H}^T(\boldsymbol{y}_{M+1} - \boldsymbol{\mu}_{M+1}(\boldsymbol{t})), & (5.24) \\ \sigma^{*2}_{GP} = \mathrm{Var}(y^*|\mathcal{D}, \boldsymbol{\mu}) &= \Psi(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{H}^T \boldsymbol{\Psi} \boldsymbol{H}, & (5.25) \end{aligned}$$

where $\boldsymbol{\mu}_{M+1}(\boldsymbol{t}) = (\mu_{M+1}(t_1), \ldots, \mu_{M+1}(t_n))^T$ is the vector of means at the data points $\boldsymbol{t} = (t_1, \ldots, t_N)$, and

$$\boldsymbol{H}^T = [\boldsymbol{\Psi}(\boldsymbol{x}^*, \boldsymbol{x}_{M+1})]^T \boldsymbol{\Psi}^{-1}, \text{ or } \boldsymbol{\Psi}(\boldsymbol{x}^*, \boldsymbol{x}_{M+1}) = \boldsymbol{\Psi}\boldsymbol{H}.$$

Therefore, the prediction is given by

$$\hat{y}^*_{M+1} = \hat{\mu}_{M+1}(t^*) + \boldsymbol{H}^T(\boldsymbol{y}_{M+1} - \hat{\boldsymbol{\mu}}_{M+1}(\boldsymbol{t})), \tag{5.26}$$

where $\hat{\mu}_{M+1}(t) = \boldsymbol{u}^T_{M+1} \hat{\boldsymbol{B}}^T \boldsymbol{\Phi}(t)$. For convenience, we call the above prediction *Type I prediction* and it consists of two parts. The first part is associated with the common mean, estimated by the data collected from all the batches, whereas the second part is linked to the GPR covariance model, estimated mainly by the data collected from the $(M+1)$-th curve—the one the new data point belongs to. On the other hand, the predictive variance is more complicated than before; it is given by the following theorem.

Theorem 5.1. *If we further assume that $\hat{\boldsymbol{B}}$ is given by (5.6), the predictive variance is given by*

$$\hat{\sigma}^{*2}_{M+1} = \hat{\sigma}^{*2}_{GP}\left(1 + \boldsymbol{u}^T_{M+1}(\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{u}_{M+1}\right), \tag{5.27}$$

where $\hat{\sigma}^{*2}_{GP}$ is given by (5.25) and all the parameters are evaluated by their estimators.

Proof. From (5.24), we have

$$\begin{aligned} \mathrm{Var}[\mathrm{E}(y^*|\mathcal{D}, \boldsymbol{\mu})] &= \mathrm{Var}[\mu_{M+1}(t^*)|\mathcal{D}] + \boldsymbol{H}^T \mathrm{Var}[\boldsymbol{\mu}_{M+1}(\boldsymbol{t})|\mathcal{D}]\boldsymbol{H} \\ &\quad - 2\boldsymbol{H}^T \mathrm{Cov}[\boldsymbol{\mu}_{M+1}(\boldsymbol{t}),\, \mu_{M+1}(t^*)|\mathcal{D}]. \end{aligned} \tag{5.28}$$

From the results around equation (5.6), the estimator of the functional mean at a single data point t is given by

$$\begin{aligned}\hat{\mu}_{M+1}(t) &= \boldsymbol{u}_{M+1}^T \hat{\boldsymbol{B}}^T \boldsymbol{\Phi}(t) = \boldsymbol{u}_{M+1}^T (\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{\Phi}(t) \\ &= \boldsymbol{u}_{M+1}^T (\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{U}^T\boldsymbol{Y}(t), \end{aligned} \tag{5.29}$$

where $\boldsymbol{Y}(t) = (y_1(t),\ldots,y_M(t))^T$, and $y_m(t)$ is the output response variable at data point t in the m-th batch for $m = 1,\ldots,M$. Since the data in different batches are independent, it follows that

$$\mathrm{Var}[\boldsymbol{Y}(t)] = \mathrm{Var}[(y_1(t),\ldots,y_M(t))^T] = \Psi(\boldsymbol{x},\boldsymbol{x})\boldsymbol{I}_M,$$

where $\boldsymbol{I}_M$ is an $M \times M$ identity matrix, and $\Psi(\boldsymbol{x},\boldsymbol{x})$ is the variance of $y(t)$ at $\boldsymbol{x}(t) = \boldsymbol{x}$. Applying the above equation to (5.29) gives the variance of $\hat{\mu}_{M+1}(t)$ as

$$\mathrm{Var}[\hat{\mu}_{M+1}(t)|\mathcal{D}] = \Psi(\boldsymbol{x},\boldsymbol{x})\gamma, \text{ with } \gamma = \boldsymbol{u}_{M+1}^T(\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{u}_{M+1}. \tag{5.30}$$

Similarly, the covariance between the estimated means at the two data points t and t^* is given by

$$\mathrm{Cov}[\hat{\mu}_{M+1}(t^*),\hat{\mu}_{M+1}(t)|\mathcal{D}] = \Psi(\boldsymbol{x}^*,\boldsymbol{x})\gamma. \tag{5.31}$$

Applying equations (5.30) and (5.31) to data points $(t_1,\ldots,t_N)$ and t^* and using the notation around (5.24) and (5.25), we have

$$\begin{aligned}\boldsymbol{H}^T\mathrm{Var}[\boldsymbol{\mu}_{M+1}(\boldsymbol{t})|\mathcal{D}]\boldsymbol{H} &= \gamma\boldsymbol{H}^T\boldsymbol{\Psi}\boldsymbol{H}, \\ \boldsymbol{H}^T\mathrm{Cov}[\boldsymbol{\mu}_{M+1}(\boldsymbol{t}),\, \mu_{M+1}(t^*)|\mathcal{D}] &= \gamma\boldsymbol{H}^T\boldsymbol{\Psi}(\boldsymbol{x}^*,\boldsymbol{x}_{M+1}) = \gamma\boldsymbol{H}^T\boldsymbol{\Psi}\boldsymbol{H}.\end{aligned}$$

Using the above results, we have

$$\begin{aligned}\mathrm{Var}(y^*|\mathcal{D}) &= \mathrm{E}\{\mathrm{Var}(y^*|\mathcal{D},\boldsymbol{\mu})\} + \mathrm{Var}\{\mathrm{E}(y^*|\mathcal{D},\boldsymbol{\mu})\} \\ &= \hat{\sigma}^{*2}_{GP} + \Psi(\boldsymbol{x}^*,\boldsymbol{x}^*)\gamma + \gamma\boldsymbol{H}^T\boldsymbol{\Psi}\boldsymbol{H} - 2\gamma\boldsymbol{H}^T\boldsymbol{\Psi}\boldsymbol{H} \\ &= \hat{\sigma}^{*2}_{GP} + \gamma(\Psi(\boldsymbol{x}^*,\boldsymbol{x}^*) - \boldsymbol{H}^T\boldsymbol{\Psi}\boldsymbol{H}) = \hat{\sigma}^{*2}_{GP}(1+\gamma).\end{aligned}$$

This proves (5.27). □

The predictive variance can be calculated similarly when penalized least squares criteria—as shown in (5.12) and (5.15)—are used, although the computation is much more complicated.

The second case that can arise, as discussed at the beginning of this section, is to calculate prediction for a completely new curve. We call it *Type II prediction*. Notationally, we still refer to the new curve as the $(M+1)$-th curve, with scalar covariate $\boldsymbol{u}_{M+1}$. Our objective is to predict y^* at $(t^*,\boldsymbol{x}^*)$ in the $(M+1)$-th batch. In this case, there are no data observed in the $(M+1)$-th batch, and thus

the training data are $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$. One simple method is to predict it using the mean part only, so that

$$\hat{y}^* = \hat{\mu}_{M+1}(t^*) = \boldsymbol{u}_{M+1}^T \hat{\boldsymbol{B}}^T \boldsymbol{\Phi}(t^*). \tag{5.32}$$

An alternative way is to assume that curves $1, 2, \ldots, M$ provide an empirical distribution of the set of all possible curves (Shi et al., 2005a), considering that

$$P(y^* \text{ belongs to the } m\text{-th curve}) = \frac{1}{M}, \tag{5.33}$$

for $m = 1, 2, \ldots, M$. To say that y^* is generated from the m-th curve means that

$$(\tilde{\tau}_{m,1}, \ldots, \tilde{\tau}_{m,n_m}, \tilde{\tau}^*)^T \sim N(0, \boldsymbol{\Omega}_m), \tag{5.34}$$

where $\tilde{\tau}(t) = y(t) - \hat{\mu}(t)$, and $\boldsymbol{\Omega}_m$ is an $(n_m + 1) \times (n_m + 1)$ covariance matrix that is defined similarly to (5.23). We can therefore calculate $\hat{y}_m^*$ and $\hat{\sigma}_m^{*2}$ from (5.26) and (5.27), respectively, as if the new data belong to the m-th batch. Since the empirical distribution is relevant to the Gaussian process component only, $\hat{y}_m^*$ is given by

$$\hat{y}_m^* = \hat{\mu}_{M+1}(t^*) + \boldsymbol{H}^T (\boldsymbol{y}_m - \hat{\boldsymbol{\mu}}_m(\boldsymbol{t})). \tag{5.35}$$

The value of $\hat{\sigma}_m^{*2}$ is given by (5.27), but the related covariance matrices are calculated at $\boldsymbol{x}^*$ and $(\boldsymbol{x}_{m,1}, \ldots, \boldsymbol{x}_{m,n_m})$.

Based on the above assumption for the empirical distribution, a prediction for the response associated with a new input $\boldsymbol{x}^*$ at t^* in a completely new curve can be calculated by

$$\hat{y}^* = \sum_{m=1}^{M} \hat{y}_m^* / M, \tag{5.36}$$

and the related predictive variance is

$$\hat{\sigma}^{*2} = \sum_{m=1}^{M} \hat{\sigma}_m^{*2} / M + \left(\sum_{m=1}^{M} \hat{y}_m^{*2} / M - \hat{y}^{*2} \right). \tag{5.37}$$

When there exists heterogeneity among different curves, the equal empirical probabilities given in (5.33) may result in a large bias. In this case, it is desirable to use a mixture model with a so-called allocation model as discussed in the next chapter.

5.3.2 Consistency

We now discuss briefly the problem of consistency in Gaussian process functional regression. There are two consistency issues to be considered here. One

is related to the common mean in (5.11) and the other is related to the prediction of $y_{M+1}(t^*)$ in the $(M+1)$-th curve. Since the common mean structure is estimated from the data collected from all M batches, it is a consistent estimator of the true mean structure under some regularity conditions (see, e.g., Ramsay and Silverman, 2005; Fan et al., 2003, and references therein).

If we have not observed any data in the $(M+1)$-th curve, we can only predict $y_{M+1}(t^*)$ using the common mean $\hat{\mu}_{M+1}(t^*)$ in (5.32) or the average prediction in (5.36) (i.e., the Type II prediction discussed previously), which is obviously not a consistent estimate of $y_{M+1}(t^*)$. This can be illustrated by looking at Figure 5.1. In this figure, the solid line stands for the true common mean curve, and the dotted line stands for the mean curve by adding independent random errors; the linear functional regression model (5.1) can be used to fit such data. However, most real functional data such as the paraplegia data presented in Figure 1.1 would look like the dashed line presented in Figure 5.1, which is usually systematically far apart from the common mean curve (solid line), although the average of all M repeated curves might close to the common mean curve. The prediction $\hat{y}_{M+1}(t^*)$ is close to the true value only when enough data from the $(M+1)$-th curve are available and the Type I prediction is used.

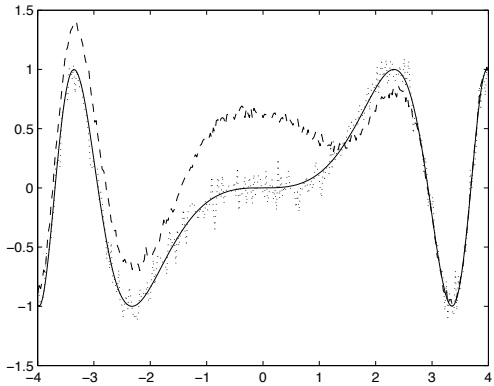


Figure 5.1 *Dashed line—the actual new curve; solid line—the true common mean curve; dotted line—the mean curve plus independent random errors.*

For the Type I prediction in (5.26), the first part $\hat{\mu}_{M+1}(t^*)$ is a consistent estimate of $\mu_{M+1}(t^*)$ when the training data $\mathcal{D}$ are large enough and some regularity conditions are satisfied. The second part is to estimate the difference between the curve and the common mean (i.e., the difference between the dashed line and the solid line in Figure 5.1). Based on the results given in

Chapter 2, $\hat{y}_{M+1}(t^*)$ is a consistent estimate of $y_{M+1}(t^*)$ when n, the size of the sample observed in the $(M+1)$-th curve, tends to infinity and when the regularity conditions are satisfied.

Seeger et al. (2008) discussed the information consistency for GPR models and also gave the error bound. Their results can also be extended to the case of Gaussian process functional regression model (Wang and Shi, 2011).

5.3.3 Applications

We now discuss some applications of the Gaussian process functional regression model based on two examples. One example concerns curve prediction using simulated data, and the other example is based on the paraplegia data discussed in Chapter 1.

Example 5.1 (curve prediction)**.** This example uses simulated data for illustrating curve predictions. The true model used to generate the data is

$$y_{mi}(x_{mi}) = u_m + \sin(0.5x_{mi})^3 + \tau_{mi} + \varepsilon_{mi}, \tag{5.38}$$

where, for each m, $x_{mi} \in (-4,4)$ and $\{\tau_{mi}\}$ is a Gaussian process with zero mean and covariance function $k(x_{mi},x_{mj}) = v_0 \exp\{-\frac{1}{2}w_0(x_{mi}-x_{mj})^2\}$, with $v_0 = 0.1$, $w_0 = 1.0$, and $\sigma_\varepsilon^2 = 0.0025$. In this example, x_{mi} is the same as t_{mi}. Thirty independent curves are generated, where $u_m = 0$ for curves 1 to 10, $u_m = -1$ for curves 11 to 20, and $u_m = 1$ for the remaining 10 curves. The sample curves are presented in Figure 5.2. In each curve, 100 data points are generated; we randomly select half of them as training data.

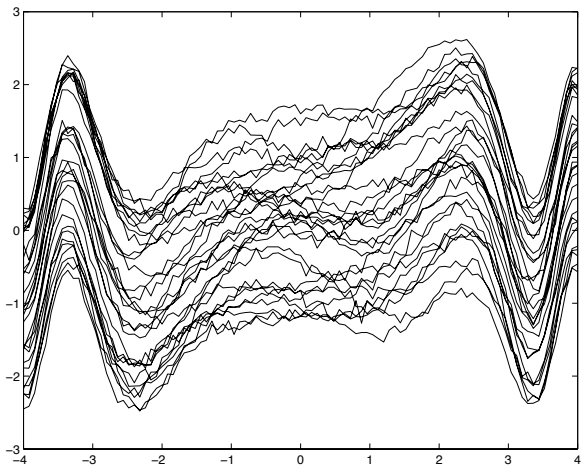


Figure 5.2 *Plots of the 30 sample curves generated from model* (5.38).

In this example, only one scalar categorical covariate u_m is used, and thus the mean structure is given such that $\mu_m(t) = \beta_1(t)$ when $u_m = -1$, $\mu_m(t) = \beta_2(t)$ when $u_m = 0$, and $\mu_m(t) = \beta_3(t)$ when $u_m = 1$. We use cubic B-spline smoothing with 18 knots equally spaced in (-4,4) and the approach discussed in Section 5.1 to estimate $\beta_i(t)$, for $i = 1,2,3$. We then apply GPR models to $\tilde{\tau}_m(t) = y_m(t) - \hat{\mu}_m(t)$ with the squared exponential covariance function (2.4). The prediction can then be calculated using (5.26).

In a new batch, we also generate 100 data points with $u_m = 1$. Under two different scenarios, half of the data is selected as training data and the other half is used as test data, i.e., predictions are calculated for the test data. We first consider an interpolation problem, i.e., the training data are selected randomly from the whole set of 100 points; second, consider an extrapolation problem, i.e., data points corresponding to $x \in (-4,0)$ are used as training data and the rest in $x \in (0,4)$ are used as test data. The second problem, the extrapolation, is obviously a more difficult issue.

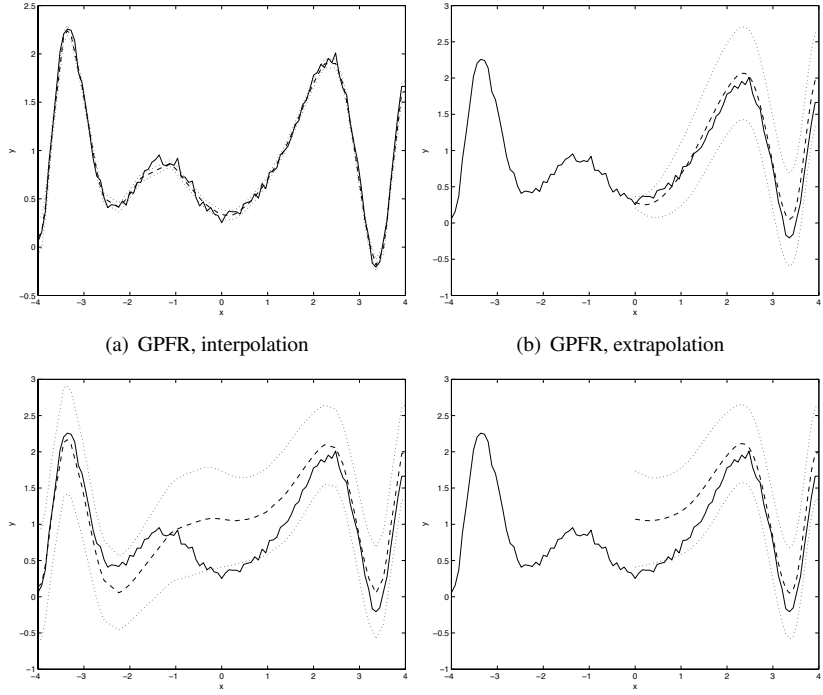
Type I predictions under two different models are calculated. The first case corresponds to the full GPFR model given in (5.11) and the solution is shown in Figure 5.3, panels (a) and (b). The second case corresponds to the linear functional regression model given in (5.1) in which we consider only the mean part, i.e., without considering the covariance part $\tau_m(\boldsymbol{x}_m)$ in (5.11); the solution is shown in Figure 5.3, panels (c) and (d). Since the LFR model takes into account the common mean structure only, the prediction is quite distant from the true curve, although it is a consistent estimate of the common mean structure for the curves with $u_m = 1$ (see the discussion given in the previous subsection).

The GPFR model, however, gives very precise results for interpolation; it actually gives consistent prediction to $y_{M+1}(t^*)$ if we have observed sufficient data, as discussed in Section 5.3.2. For the problem of extrapolation shown in panel (b), when the test data are "close" to the training data, the GPFR model uses both the mean structure and the covariance structure to calculate predictions, which results in a very precise result (similar to the interpolation in this case); when the test data are distant from the training data, the GPFR model relies mainly on the mean structure, and the fit is still reasonably good (similar to the linear functional regression model in this case).

We next apply the Gaussian process functional regression model to the example of standing-up maneuvers by paraplegic patients as discussed in Section 1.1.2.

Example 5.2 (Paraplegia data)**.** The response variable is the vertical trajectories of the body COM, denoted by *comz*. Forty curves for 40 standings-up, 5 for each of the 8 patients, are presented in Figure 1.1.

To model the mean structure in the GPFR model, we use three covariates for $\boldsymbol{u}_m$, namely, the patient's height, weight, and sex. These are natural covariates for the mean structure model, since the trajectory of the body COM



(a) GPFR, interpolation (b) GPFR, extrapolation

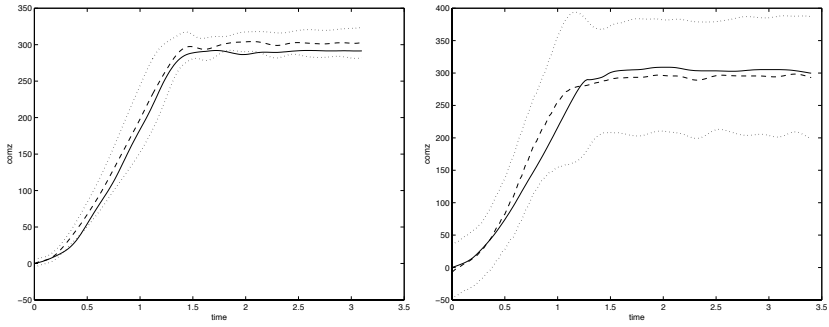
(c) LFR, interpolation (d) LFR, extrapolation

Figure 5.3 *The predictions obtained by using the different models, where the solid line represents the true curve, the dashed line represents the predictions, and the dotted line represents 95% prediction intervals.*

obviously depends on the patient's height and weight. Also, the standing-up strategy adopted may be different for male and female patients and, therefore, the shape of the trajectory is likely to differ. Figure 1.1 indicates that the time scale may be different for different standings-up. The data need to be aligned before estimating the mean structure. This step is also known as *data registration*; see, e.g., Ramsay and Silverman (2005), Chapter 7, and Ramsay (1998). We then estimate the mean structure and covariance structure by the method discussed early in this section.

Due to the nature of the example (i.e., no data recorded for new patients), we consider Type II predictions only, using the data collected from the other batches to predict $y(t)$ in a new batch. Specifically, we use the average prediction given in (5.36), and two cases are considered for this purpose. In the first case, we use the data that have already been observed for one patient to predict a new standing-up for the same patient. The results are presented in Figure

5.4(a). Since the data collected from the same patient should have the same model structure, the GPFR model provides a very good prediction as shown in the figure. In the second case, we predict *comz* for a new patient using the data collected from the other patients. The prediction and its actual curve for one standing-up are presented in Figure 5.4(b). For this standing-up, the GPFR model still predicts *comz* quite well, although the uncertainty in this case is much larger (with a wider prediction interval) than the one in the first case.



(a) GPFR, same patient (b) GPFR, new patient

Figure 5.4 *Paraplegia data: the true test data (solid line), the prediction (dashed line), and the 95% prediction intervals (dotted line). Prediction of a new standing-up using the data from (a) the same patient, and from (b) different patients.*

The average RMSE between the prediction and the actual values for the five standings-up for the patient associated with Figure 5.4(b) is 14.67. However, when we consider an atypical new female patient (59 kg and 178 cm, thin and tall), the average value of RMSE for the five standings-up is 62.8, a case of failure. We therefore need a different model structure to capture such an anomaly and this is discussed in the next chapter.

5.4 Mixed-effects GPFR models

A mixed-effects GPFR model is defined as

$$y_m(t) = \mu_m(t) + \boldsymbol{v}_m^T(t)\boldsymbol{\gamma} + \boldsymbol{w}_m^T(t)\boldsymbol{b}_m + \tau_m(\boldsymbol{x}_m(t)) + \varepsilon_m(t), \tag{5.39}$$

with

$$\boldsymbol{b}_m \sim N(0, \boldsymbol{\Sigma}), \;\; \varepsilon_m(t) \sim N(0, \sigma_\varepsilon^2), \tag{5.40}$$

where $\varepsilon_m(t)$ are random errors that are independent at different times; $\boldsymbol{v}_m(t)$, $\boldsymbol{w}_m(t)$, and $\boldsymbol{x}_m(t)$ are functional covariates with dimension r, k, and Q, respectively; $\tau_m(\cdot)$ has a GPR model that is defined in (5.9). The first term $\mu_m(t)$ in (5.39) depends on the scalar covariates $\boldsymbol{u}_m$; we may use a linear functional

regression model defined in (5.1), i.e., $\mu_m(t) = \boldsymbol{u}_m^T \boldsymbol{\beta}(t)$, where $\boldsymbol{\beta}(t)$ may be approximated by B-spline basis functions as shown in (5.2).

Note that the mixed-effect GPFR in (5.39) is an extended form of the GPFR model (5.11), where the two new terms, $\boldsymbol{v}_m^T(t)\boldsymbol{\gamma}$ and $\boldsymbol{w}_m^T(t)\boldsymbol{b}_m$, are added to achieve the same purpose as that in linear mixed-effects models. In the example of the renal data, as discussed in Section 1.3.2, the response curve is the measurement of hemoglobin (Hb) concentration for renal anemia patients, which is a functional response variable changing over time t. The related function-valued covariates are the dosage of the epoetin agent, and other covariates such as iron dosage levels and Hb levels measured in the previous month, which all change with time. A preliminary study shows that there is a linear relationship between $y(t)$ and some function-valued covariates. However, such a linear relationship may vary for different subjects, which can be explained by a linear mixed-effect model. Indeed, the heterogeneity among different subjects is quantified by $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_k^2)$ via the mixed-effect model structure.

As in the Gaussian process functional regression model, the term $\tau_m(\cdot)$ in (5.39) plays a very important role. First, it is used to model the relationship between functional response variables and other functional covariates. Even for the covariates included in $\boldsymbol{v}$ or $\boldsymbol{w}$, $y(t)$ may depend on them nonlinearly apart from the linear part in (5.39). We therefore use a nonparametric GPR model to describe such unknown nonlinearity. Second, it introduces a covariance structure among different data points in each curve. This enables us to explore the individual relationship between response variable and covariates for each individual curve, and thus, it enables us to find, for example, the dose-response curve for each individual patient as we will discuss further in Section 5.4.2.

5.4.1 Model learning and prediction

Similar to previous sections, we also assume that all functional variables are observed at the same data points of $\{t_{mi}, i = 1, \ldots, n_m\}$ for the m-th batch, and denote the data by $\mathcal{D}_m$; this now includes $\boldsymbol{v}_m(t_{mi})$ and $\boldsymbol{w}_m(t_{mi})$ in addition to the variables in (5.10). Let $\boldsymbol{y}_m$ be the vector of $\{y_{mi}, i = 1, \ldots, n_m\}$, $\boldsymbol{t}_m = \{t_{mi}, i = 1, \ldots, n_m\}$, $\boldsymbol{V}_m$ be the $n_m \times r$ matrix with the i-th row $\boldsymbol{v}_m^T(t_{mi})$, and $\boldsymbol{W}_m$ be the $n_m \times k$ matrix with the i-th row $\boldsymbol{w}_m^T(t_{mi})$. Then the mixed-effect GPFR model based on the n_m observations from the m-th batch can be written in matrix form as

$$\boldsymbol{y}_m = \boldsymbol{\Phi}_m \boldsymbol{B} \boldsymbol{u}_m + \boldsymbol{V}_m \boldsymbol{\gamma} + \boldsymbol{W}_m \boldsymbol{b}_m + \boldsymbol{\tau}_m + \boldsymbol{\varepsilon}_m, \tag{5.41}$$

where $\boldsymbol{\Phi}_m$ is an $n_m \times H$ matrix as defined in Section 5.1, $\boldsymbol{\varepsilon}_m \sim N(\boldsymbol{0}, \sigma_\varepsilon^2 \boldsymbol{I})$, and $\boldsymbol{\tau}_m^T = (\tau_1, \ldots, \tau_{n_m})$ follows a GPR model as shown in (5.9).

The unknown parameters involved in the above mixed-effects GPFR models include the B-spline coefficient $\boldsymbol{B}$, the fixed effect coefficient $\boldsymbol{\gamma}$, the random

effect covariance matrix $\boldsymbol{\Sigma}$, the parameters $\boldsymbol{\theta}$ involved in the covariance function, and the measurement error variance σ_ε^2. These parameters are denoted collectively as $\boldsymbol{\Theta} = (\boldsymbol{B}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\Sigma}, \sigma_\varepsilon^2)$. Using the same idea of the empirical Bayesian approach, the hyper-parameters $\boldsymbol{\theta}$ can be treated as other main parameters of interest, and we may then calculate the estimates of $\boldsymbol{\Theta}$ by maximizing the following marginal log-likelihood:

$$\begin{aligned} & l(\boldsymbol{\Theta}) = l(\boldsymbol{B}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\Sigma}, \sigma_\varepsilon^2) \\ = & \sum_{m=1}^{M} \Big\{ -\frac{1}{2} n_m \log(2\pi) - \frac{1}{2} \log |\boldsymbol{\Psi}_m| \\ & -\frac{1}{2} (\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B} \boldsymbol{u}_m - \boldsymbol{V}_m \boldsymbol{\gamma})^T \boldsymbol{\Psi}_m^{-1} (\boldsymbol{y}_m - \boldsymbol{\Phi}_m \boldsymbol{B} \boldsymbol{u}_m - \boldsymbol{V}_m \boldsymbol{\gamma}) \Big\}, \end{aligned} \quad (5.42)$$

where

$$\boldsymbol{\Psi}_m = \boldsymbol{W}_m \boldsymbol{\Sigma} \boldsymbol{W}_m^T + \boldsymbol{K}_m + \sigma_\varepsilon^2 \boldsymbol{I} \text{ and } \boldsymbol{K}_m = (k_m(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

for $i, j = 1, \ldots, n_m$. $\boldsymbol{K}_m$ is the covariance matrix of $\boldsymbol{\tau}_m$ with $k_m(\cdot, \cdot)$ as its covariance function. As shown in Shi et al. (2010), it is more efficient to use an iterative method than using a direct maximization procedure for $l(\boldsymbol{\Theta})$ in (5.42) to calculate the estimate of $\boldsymbol{\Theta}$. The iterative method is given as follows.

Algorithm 5.2 (Iterative ML approach)**.** *Each iteration includes two steps:*

1. *Update* $\boldsymbol{B}$ *and* $\boldsymbol{\gamma}$ *given the current values of* $(\boldsymbol{\theta}, \boldsymbol{\Sigma}, \sigma_\varepsilon^2)$*;*
2. *Update* $(\boldsymbol{\theta}, \boldsymbol{\Sigma}, \sigma_\varepsilon^2)$ *by maximizing* $l(\boldsymbol{\Theta})$ *in* (5.42) *given* $\boldsymbol{B}$ *and* $\boldsymbol{\gamma}$*.*

In step 2, since $(\boldsymbol{\theta}, \boldsymbol{\Sigma}, \sigma_\varepsilon^2)$ are included in the covariance matrix $\boldsymbol{\Psi}_m$, the log-likelihood in (5.42) given $\boldsymbol{B}$ and $\boldsymbol{\gamma}$ is analogous to the log-likelihood (3.7) we have discussed in Section 3.1. We can update them by maximizing the log-likelihood directly.

Given $(\boldsymbol{\theta}, \boldsymbol{\Sigma}, \sigma_\varepsilon^2)$, there are analytic forms to update $\boldsymbol{B}$ and $\boldsymbol{\gamma}$ in step 1. We listed the formulae here without giving the proof; further details can be found in (Shi et al., 2010).

$$\text{vec}(\boldsymbol{B}) = (\boldsymbol{A}_{11} - \boldsymbol{A}_{12} \boldsymbol{A}_{22}^{-1} \boldsymbol{A}_{21})^{-1} (\boldsymbol{y}_{(1)} - \boldsymbol{A}_{12} \boldsymbol{A}_{22}^{-1} \boldsymbol{y}_{(2)}), \quad (5.43)$$

$$\boldsymbol{\gamma} = (\boldsymbol{A}_{22} - \boldsymbol{A}_{21} \boldsymbol{A}_{11}^{-1} \boldsymbol{A}_{12})^{-1} (\boldsymbol{y}_{(2)} - \boldsymbol{A}_{21} \boldsymbol{A}_{11}^{-1} \boldsymbol{y}_{(1)}), \quad (5.44)$$

where

$$\boldsymbol{y}_{(1)} = \sum_{m=1}^{M} (\boldsymbol{u}_m \otimes \boldsymbol{\Phi}_m^T)\boldsymbol{\Psi}_m^{-1}\boldsymbol{y}_m,$$
$$\boldsymbol{y}_{(2)} = \sum_{m=1}^{M} \boldsymbol{V}_m^T\boldsymbol{\Psi}_m^{-1}\boldsymbol{y}_m,$$
$$\boldsymbol{A}_{11} = \sum_{m=1}^{M} (\boldsymbol{u}_m \otimes \boldsymbol{\Phi}_m^T)\boldsymbol{\Psi}_m^{-1}(\boldsymbol{u}_m^T \otimes \boldsymbol{\Phi}_m),$$
$$\boldsymbol{A}_{21}^T = \boldsymbol{A}_{12} = \sum_{m=1}^{M} (\boldsymbol{u}_m \otimes \boldsymbol{\Phi}_m^T)\boldsymbol{\Psi}_m^{-1}\boldsymbol{V}_m,$$
$$\boldsymbol{A}_{22} = \sum_{m=1}^{M} \boldsymbol{V}_m^T\boldsymbol{\Psi}_m^{-1}\boldsymbol{V}_m.$$

After we estimate the parameter $\boldsymbol{\Theta}$, we can calculate the prediction. We first re-express the mixed-effects GPFR model (5.39) and (5.40) by

$$y_m(t) = \tilde{\mu}_m(t) + \tilde{\tau}_m(\boldsymbol{z}_m, \boldsymbol{w}_m, t) + \varepsilon_m(t), \tag{5.45}$$

where

$$\tilde{\mu}_m(t) = \boldsymbol{u}_m^T\boldsymbol{\beta}(t) + \boldsymbol{v}_m^T(t)\boldsymbol{\gamma}$$

and

$$\tilde{\tau}_m = \boldsymbol{W}_m\boldsymbol{b}_m + \boldsymbol{\tau}_m.$$

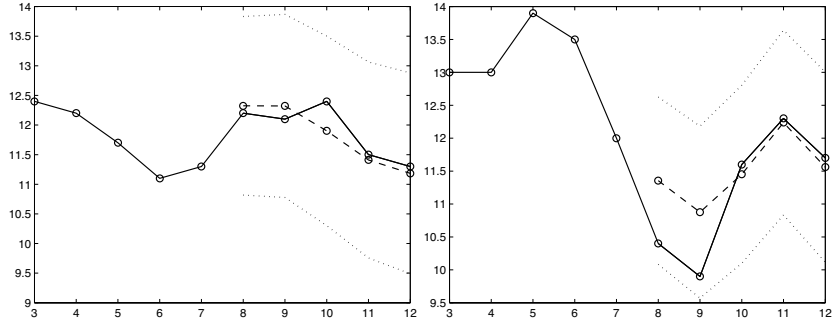
Thus, $\tilde{\tau}_m$ still has a Gaussian process regression model with the following covariance function:

$$k_m(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}) + \sum_{q=1}^{k} \sigma_q^2 w_{iq} w_{jq}. \tag{5.46}$$

We therefore can use the formulae given in Section 5.3.1 to calculate the prediction by replacing the mean model and covariance functions by the ones given above.

5.4.2 Application: dose-response study

We consider the example of the renal data introduced in Section 1.3.2. A detailed discussion can be found in Shi et al. (2010); here we only use this example to illustrate how to apply the mixed-effects GPFR model. Let us use a liner mixed-effects model as given in (5.39), where the fixed-effects part involves the covariates $\boldsymbol{v}_m(t) = \{1, t, d_m(t-2), d_m^2(t-2)\}$ and the random-effects part involves the covariates $\boldsymbol{w}_m(t) = \{1, d_m(t-2)\}$. Here, t is the time and $d_m(t-2)$ is the dose level for subject m at time $t-2$ (i.e., the dosage of the drug taken



(a) Patient 1 (b) Patient 2

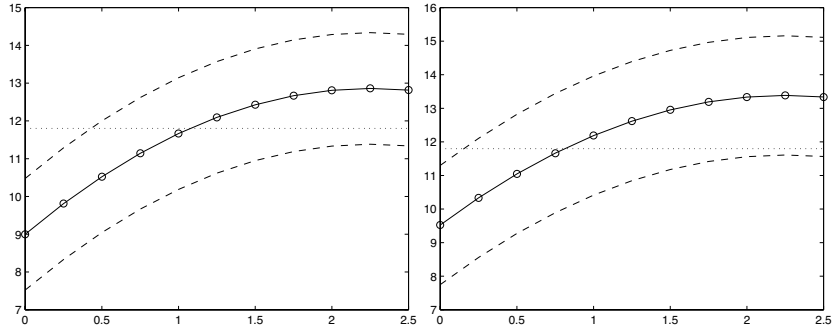
Figure 5.5 *Prediction of Hb for two patients: the solid line stands for the actual observations, the dashed line stands for the predictions, and the dotted lines stand for the 95% predictive intervals.*

between 30 and 60 days ago). To model the covariance structure τ_m, we use all the functional covariates, including the covariates used in $\boldsymbol{v}_m(t)$.

We first use the data collected from all the patients to estimate the unknown parameters involved in the model by using Algorithm 5.2. For an individual patient, the common mean and the data collected from this patient are used to calculate the prediction. We use the data up to the current month and assume that a dosage level from that specific day to the next 30 days is used to predict the Hb level in 60 days. Based on the discussion given in Section 5.3.2, the prediction is expected to be close to the true value of Hb level for this patient when enough data are collected from the same patient. This enables us to find the individual dose-response relationship for each patient if we can collect enough data for the patient. Figure 5.5 shows the predictions and the actual observations for two typical patients. The prediction is shown to be very accurate and would be even more accurate if more data are recorded.

Dose-response relationship describes the change in effect on an organism or patient caused by differing doses. In this example, the aim is to study how the Hb level changes by using different doses. The mixed-effects GPFR model can be used to construct a dose-response curve for each *individual* patient.

To obtain a dose-response curve, we need to calculate predictions of responses for different dose levels. We take 11 different doses from 0 to 2.5, and then calculate the predicted values for each of those dose levels. A specific predictive dose-response curve is given in Figure 5.6 for a representative patient. Panel (a) presents the predicted values of Hb for Month 8 based on the data collected during the first 6 months. The dotted line is the target control level of Hb. From the figure it seems that a dose of around 1.1 units would be the most appropriate. Panel (b) presents a similar dose-response curve but for



(a) Month 8 (b) Month 12

Figure 5.6 *Hb response for different dose levels: the solid line stands for predictions with different dose levels, the dashed line stands for their 95% predictive intervals, and the dotted line stands for the target control level of* $Hb = 11.8$.

Month 12 based on the data collected during the first 10 months. Notice that the recommended dose for this specific patient should be around 0.8 units.

As we have mentioned, the mixed-effects GPFR model considers both the global information available from all the patient data and also the special features or peculiarities of each individual. In this respect, Figure 5.6 is just a snapshot of that fact: the prediction is capturing the common model structure learned from all the data but also patients' specific information, which are the data recorded in the first 6 months for Panel (a) and 10 months for Panel (b), learned via the model covariance structure. These dose-response curves are very useful in practice, as clinicians may use them to prescribe the most suitable dosage on a patient-by-patient basis. This allows a so-called *patient-specific treatment regime*.

5.5 GPFR ANOVA model

A one-way functional ANOVA (analysis of variance) model is defined by (Ramsay and Silverman, 2005, Chapter 13)

$$y_{mj}(t) = \mu(t) + \alpha_j(t) + \varepsilon_{mj}(t), \;\; \varepsilon_{mj}(t) \sim N(0, \sigma_\varepsilon^2), \tag{5.47}$$

where $y_{mj}(t)$ denotes a functional response from batch m under level j of the factor (or j-th group) for $j = 1, \ldots, J$ and $m = 1, \ldots, M_j$. Here, $\mu(t)$ is the grand mean function, and the term $\alpha_j(t)$ is the specific effect for level j. They are required to satisfy the constraint

$$\sum_j \alpha_j(t) = 0, \;\; \text{for all } \; t, \tag{5.48}$$

or equivalently

$$\alpha_1(t) = 0, \text{ for all } t. \tag{5.49}$$

Under the constraint (5.48), $\mu(t)$ is the average effect across all levels; while under the constraint (5.49), $\mu(t)$ is the effect for level 1 and $\alpha_j(t)$ ($j \neq 1$) is the difference of the effects comparing level j with level 1.

In model (5.47), $\mu(t)$ and $\alpha_j(t)$'s are functions of one-dimensional covariate t. We can use, for example, B-spline basis functions to approximate these functions as we discussed earlier in this chapter. Other methods such as using regularized basis expansions can be found in Chapter 13 of Ramsay and Silverman (2005).

For spatial data, the response variable may depend on multidimensional spatial covariates $\boldsymbol{x}$. Thus, model (5.47) becomes

$$y_{mj}(\boldsymbol{x}) = \mu + \alpha_j(\boldsymbol{x}) + \varepsilon_{mj}(\boldsymbol{x}), \ \ \varepsilon_{mj}(\boldsymbol{x}) \sim N(0, \sigma_\varepsilon^2). \tag{5.50}$$

This is a *spatial* ANOVA model. The structure μ is meant to capture patterns in the common response. We can usually find a parametric mean model depending on scalar variables, using, for example, model (5.1). The individual effect can be modeled by a Gaussian process regression model as follows:

$$\alpha_j(\boldsymbol{x}) \sim GPR[\boldsymbol{0}, k_j(\boldsymbol{\theta}_j)|\boldsymbol{x}] \ \text{ for } j = 2, \ldots, J, \tag{5.51}$$

where $\boldsymbol{\theta}_j$ are hyper-parameters involved in the covariance functions. In general, as suggested by Kaufman and Sain (2010), we can also use a Gaussian process regression model to fit the effect for the baseline level 1. The model (5.50) is therefore modified by

$$y_{mj}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \alpha_j(\boldsymbol{x}) + \varepsilon_{mj}(\boldsymbol{x}), \tag{5.52}$$

where $\mu(\boldsymbol{x}) \sim GPR[\mu, k_\mu(\boldsymbol{\theta}_\mu)|\boldsymbol{x}]$ and $\boldsymbol{\theta}_\mu$ are hyper-parameters. Here μ can be a constant but can also be modeled by a parametric model. In the above model, we assume that $\tau(\boldsymbol{x})$, $\alpha_j(\boldsymbol{x})$, and $\varepsilon_{mj}(\boldsymbol{x})$ are independent.

Assume that we have observed the data

$$\mathcal{D} = \{(y_{mji}, \boldsymbol{x}_i), \ i = 1, \ldots, n, j = 1, \ldots, J, m = 1, \ldots, M_j\}, \tag{5.53}$$

where $y_{mji} = y_{mj}(\boldsymbol{x}_i)$. For simplicity, we assume that the observed covariate values $\boldsymbol{x}$ are identical for all levels, although model (5.52) can be applied to a general case without introducing any more difficulties.

The main aim of functional ANOVA is to compare the effects for different levels, i.e., to compare the posterior mean of $E(\alpha_j(\boldsymbol{x})|\mathcal{D})$ for different j's. This can be achieved by using a numerical scheme such as the empirical Bayesian approach, the MAP approach, or the MCMC algorithm as

we have discussed before. Among them, we briefly introduce an MCMC algorithm augmented with latent variables $\boldsymbol{\alpha}_j = (\alpha_j(\boldsymbol{x}_1), \ldots, \alpha_j(\boldsymbol{x}_n))^T$. Letting $\boldsymbol{y}_{mj} = (y_{mj}(\boldsymbol{x}_1), \ldots, y_{mj}(\boldsymbol{x}_n))^T$, we have

$$\boldsymbol{y}_{mj}|\boldsymbol{\theta}, \mu, \boldsymbol{\alpha}_j \sim N(\mu + \boldsymbol{\alpha}_j, \boldsymbol{\Psi}_\mu(\boldsymbol{\theta})), \tag{5.54}$$

where $\boldsymbol{\Psi}_\mu(\boldsymbol{\theta}) = \boldsymbol{K}_\mu(\boldsymbol{\theta}_\mu) + \sigma_\varepsilon^2 \boldsymbol{I}$, in which the first term is calculated from the covariance function in (5.52) at n values of $\boldsymbol{x} = \{\boldsymbol{x}_i, i = 1, \ldots, n\}$ and $\boldsymbol{\theta} = (\boldsymbol{\theta}_\mu, \sigma_\varepsilon^2)$. In the second term, $\boldsymbol{I}$ stands for an identity matrix. We also have

$$\boldsymbol{\alpha}_j \sim N(\boldsymbol{0}, \boldsymbol{\Psi}_j(\boldsymbol{\theta}_j)), \ j = 2, \ldots, J, \tag{5.55}$$

where $\boldsymbol{\Psi}_j(\boldsymbol{\theta}_j)$ is calculated similarly from the covariance function in (5.51) evaluated at n values of $\boldsymbol{x}$.

We then use a Gibbs sampler to generate random variates. Letting

$$\boldsymbol{\Theta} = \{\mu, \boldsymbol{\theta}, \boldsymbol{\theta}_j, j = 2, \ldots, J\} \ \text{ and } \ \boldsymbol{\alpha} = \{\boldsymbol{\alpha}_j, j = 2, \ldots, J\},$$

the algorithm is given as follows.

Algorithm 5.3 (Gibbs sampler for the GPFR ANOVA model)**.** *A sweep of each iteration includes the following steps:*

1. *Generate* $\boldsymbol{\theta}$ *from* $p(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\alpha}, \mu)$*;*
2. *Generate* μ *from* $p(\mu|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\theta})$*;*
3. *Generate* $\boldsymbol{\theta}_j$ *from* $p(\boldsymbol{\theta}_j|\boldsymbol{\alpha}_j)$ *for* $j = 2, \ldots, J$*;*
4. *Generate* $\boldsymbol{\alpha}$ *from* $p(\boldsymbol{\alpha}|\mathcal{D}, \boldsymbol{\Theta})$.

In step 1, we use the fact that $\boldsymbol{y}_{mj}$ have a conditional multivariate normal distribution in (5.54), where the mean is given. The conditional density of $\boldsymbol{\theta}$ is therefore given in (3.11), where the likelihood is calculated from the multivariate normal distribution in (5.54). We can therefore use the hybrid Monte Carlo method discussed in Section 3.2 and Appendix A.4 in this step.

Step 2 depends on what mean model is used. Also from (5.54), when the covariance matrix and $\boldsymbol{\alpha}_j$ are given and if we use a parametric model as a mean model, a routine Bayesian method for Gaussian data can be used to find the conditional density of μ (or the parameters involved in μ).

In step 3, if we assume independent priors for $\boldsymbol{\theta}_j$, the random variates of $\boldsymbol{\theta}_j$ can also be generated independently for each of j in $\{2, \ldots, J\}$. We can also use a hybrid Monte Carlo method similar to the one used in step 1.

Finally, we look at step 4. We use the fact that

$$\begin{aligned} p(\boldsymbol{\alpha}|\mathcal{D}, \boldsymbol{\Theta}) &= p(\boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_J|\mathcal{D}, \boldsymbol{\Theta}) \\ &\propto \prod_{j=2}^{J} \left[\prod_{m=1}^{M_j} p(\boldsymbol{y}_{mj}|\boldsymbol{\theta}, \mu, \boldsymbol{\alpha}_j) p(\boldsymbol{\alpha}_j|\boldsymbol{\theta}_j) \right], \end{aligned}$$

thus, $\boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_J$ are conditional independent given $(\mathcal{D}, \boldsymbol{\Theta})$, and

$$p(\boldsymbol{\alpha}_j|\mathcal{D}, \boldsymbol{\Theta}) \propto \prod_{m=1}^{M_j} p(\boldsymbol{y}_{mj}|\boldsymbol{\theta}, \mu, \boldsymbol{\alpha}_j) p(\boldsymbol{\alpha}_j|\boldsymbol{\theta}_j).$$

Since $p(\boldsymbol{y}_{mj}|\boldsymbol{\theta}, \mu, \boldsymbol{\alpha}_j)$ is the normal density function given in (5.54) and $p(\boldsymbol{\alpha}_j|\boldsymbol{\theta}_j)$ is the normal density funciton given in (5.55), it is easy to prove that the conditional distribution of $\boldsymbol{\alpha}_j$ is also a normal distribution:

$$\boldsymbol{\alpha}_j|\mathcal{D}, \boldsymbol{\Theta} \sim N\left(\boldsymbol{\Omega}_{\alpha,j}\boldsymbol{\Psi}_\mu^{-1}\left[\sum_{m=1}^{M_j}(\boldsymbol{y}_{mj} - \mu)\right], \boldsymbol{\Omega}_{\alpha,j}\right) \tag{5.56}$$

where

$$\boldsymbol{\Omega}_{\alpha,j}^{-1} = M_j\boldsymbol{\Psi}_\mu^{-1} + \boldsymbol{\Psi}_j^{-1}.$$

After a burn-in period, we can select a set of random covariates to carry out posterior inference. For example, the posterior mean $E(\alpha_j(\boldsymbol{x}_i)|\mathcal{D})$ can be approximated simply by the sample mean in the selected set of random variates for $\alpha_j(\boldsymbol{x}_i)$. The values of $\alpha_j(\boldsymbol{x})$ at any other $\boldsymbol{x}$ can be calculated from the GPR model (5.51), using formula (2.7).

In (5.51), we assumed independent Gaussian process regression model for each $\alpha_j(\boldsymbol{x})$. We could also use a multivariate Gaussian process regression model. One example suggested in Kaufman and Sain (2010) is to use a multivariate GPR model with zero mean and the following covariance function:

$$\mathrm{Cov}(\alpha_j(\boldsymbol{x}), \alpha_{j'}(\boldsymbol{x}')) = \begin{cases} (1-\frac{1}{J})\sigma_\alpha^2 k(\boldsymbol{x}, \boldsymbol{x}'), & \text{if } j = j', \\ -\frac{1}{J}\sigma_\alpha^2 k(\boldsymbol{x}, \boldsymbol{x}'), & \text{if } j \neq j'. \end{cases}$$

If we used this model we would need to resort to a sub-Gibbs sampler in step 4, and generate each of the $\boldsymbol{\alpha}_j$ conditional on the other $\boldsymbol{\alpha}_{j'}$'s for $j' \neq j$ (see the details in Kaufman and Sain, 2010).

A two-way functional ANOVA model can be defined similarly:

$$y_{mjk}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \alpha_j(\boldsymbol{x}) + \beta_k(\boldsymbol{x}) + (\alpha\beta)_{jk}(\boldsymbol{x}) + \varepsilon_{mj}(\boldsymbol{x}), \tag{5.57}$$

where $\beta_k(\boldsymbol{x})$ is the effect under level k for the second factor, and $(\alpha\beta)_{jk}(\boldsymbol{x})$ is the interaction between the two factors. Effects $\beta_k(\boldsymbol{x})$ satisfy constraints similar to (5.49), i.e., $\beta_1(\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$ and $(\alpha\beta)_{jk}(\boldsymbol{x})$ satisfy the constraints that $(\alpha\beta)_{jk}(\boldsymbol{x}) = 0$ for either $j = 1$ or $k = 1$. The inference for the two-way functional ANOVA model is quite similar to the one-way functional ANOVA model although the model structure is more complicated. Detailed discussions can be found in Kaufman and Sain (2010).

5.6 Further reading and notes

The main purpose of this chapter is to model the regression relationship between a functional response variable and a set of functional covariates for repeated curves using a GPFR model. Gaussian process plays a key role in the model for explaining the covariance structure. A wide class of nonlinear regression functions can be accommodated by the GPFR model via a proper choice of a covariance function. In addition, the Gaussian process regression can be used to address the regression problem with a large number of functional covariates, while most traditional nonparametric methods are limited to problems with functional covariates of small dimension (usually less than three) due to the *curse of dimensionality*. This enables GPFR models to treat *spatial* data easily; for example, the GPFR ANOVA model discussed in Section 5.4.

It is a difficult problem to fit or predict a curve or a functional response variable with high-dimensional input functional variables. Neural network models are another popular approach (at least in the engineering community); see, for example, Cheng and Titterington (1994) and Neal (1996). Alternative approaches are to impose special structures to the model. Commonly used models include the additive model (Breiman and Friedman, 1985; Hastie and Tibshirani, 1990); the varying-coefficient model (see, e.g., Hastie and Tibshirani, 1993; Fan and Zhang, 1999; Fan et al., 2003), and dimension reduction methods which include the projection pursuit (Huber, 1985), the sliced inverse regression (Li, 1991), and the single index models (Härdle and Stoker, 1989; Choi et al., 2011).

Another feature of the GPFR model is that it models both mean structure and covariance structure; the mean structure can capture the common pattern among all the curves and the covariance structure can cope with the characteristics of each individual curve. This opens up the possibility of carrying out research such as *patient-specific treatment regimes* based on individual dose-response relationships as discussed in Section 5.3. Rice and Silverman (1991) used the idea of modeling the mean and covariance structure simultaneously, in which the mean function is approximated by a cubic spline and the covariance structure is estimated via a smooth nonparametric estimate of the relevant eigenfunctions (see also Hall et al., 2008). But the method is limited to the case of a one-dimensional input functional variable. It is more efficient to model the covariance by a Gaussian process model although we need to select a covariance function carefully. There are other methods to model the covariance structure, for instance, the method based on a modified Cholesky decomposition (see, e.g., Pourahmadi, 1999; Leng et al., 2009).

This page intentionally left blank

Chapter 6

Mixture models and curve clustering

As discussed in Example 5.2 in Chapter 5, the paraplegia data of standing-up maneuvers are collected from different standings-up among different patients, and thus the regression relationship may depend on patients' personal characteristics such as their height, weight, the tactics used in each standing-up, and other circumstantial factors in performing the experiment. In many cases, these features result in functional clustering curves or so-called "spatially" indexed data, representing that the actual regression models may be the same for some batches of data, but may be different for others. This is so-called *heterogeneity* among curves, and such a heterogeneity is usually hard to be captured by standard functional regression models such as the GPFR model that we described in the previous chapter.

In this chapter, we use a mixture model of Gaussian process functional regression models to deal with this type of data, and describe how to use the mixture models to address the problem of heterogeneity (see, e.g., Shi and Wang, 2008). Specifically, in Section 6.1 we describe a mixture GPR model and an MCMC algorithm to implement thereof. A mixture of GPFR models is described in Section 6.2. How to use an EM algorithm to learn the model and how to calculate predictions are discussed, respectively, in Sections 6.2.1 and 6.2.2. The problem of curve clustering is discussed in Section 6.3. Further reading and notes are given in Section 6.4.

6.1 Mixture GPR models

Mixture models have been studied for many decades in their two main roles in addition to others. One is to model *heterogeneity* for data coming from different populations and the other is that it can be used as a convenient form of flexible population density (see, e.g., Titterington et al., 1985; McLachlan and Peel, 2000). A mixture model is defined by

$$Y \sim p(y) = \sum_{k=1}^{K} \pi_k p_k(y), \tag{6.1}$$

where $0 \leq \pi_k \leq 1, \pi_1 + \cdots + \pi_K = 1$ and $p_k(y)$ is a probability density function for a continuous random variable or a probability mass function for a discrete random variable. An alternative way to define a mixture model is to use a latent indicator random variable, say, Z. Then, the mixture model defined in (6.1) is equivalent to the following models:

$$Y|Z=k \sim p_k(y), \tag{6.2}$$
$$P(Z=k) = \pi_k, \ k=1,\ldots,K. \tag{6.3}$$

From the alternative definition in (6.2) and (6.3), the mixture model can be interpreted as follows: given that $Z=k$, Y belongs to the k-th component or k-th "cluster" and has a distribution of $p_k(y)$. The latent variable Z has a multinomial distribution with parameters $(\pi_1,\ldots,\pi_K)$.

We now define a mixture Gaussian process regression model based on the general concept of the mixture model defined above. For each curve $y_m(t)$, we define an indicator variable z_m for $m=1,\ldots,M$. Then, mixtures of GPR models are defined by

$$\begin{aligned} y_m(t) &= f_m(t) + \varepsilon_m(t), \ \ \varepsilon_m(t) \sim N(0,\sigma_\varepsilon^2), \\ f_m(t)|z_m = k &\sim GPR_k[\mathbf{0}, k(\boldsymbol{\theta}_k)|\boldsymbol{x}_m(t)]. \end{aligned} \tag{6.4}$$

Here, the first k represents the k-th component but $k(\cdot)$ represents a covariance function. The definition of *GPR* has already been given in (1.11). Consequently, if we have observed data

$$\mathcal{D} = \{\boldsymbol{y}_m = (y_{m1},\ldots,y_{mn_m}), \boldsymbol{x}_{mi} = (x_{mi1},\ldots,x_{miQ}), i=1,\ldots,n_m, m=1,\ldots,M\},$$

we have

$$\boldsymbol{y}_m|z_m = k \sim N(0, \Psi_{n_m}(\boldsymbol{\theta}_k)),$$

where Ψ_{n_m} is defined in (2.6) but with covariance function $k(\boldsymbol{\theta}_k)$.

6.1.1 MCMC algorithm

The Expectation Maximization (EM) algorithm is certainly the most popular choice for mixture models implementation within a non-Bayesian context; for a fully Bayesian approach, MCMC methods are commonly preferred (see, e.g., Robert and Casella, 2004). We explain the EM algorithm in the next section and focus on an MCMC algorithm, in particular the Gibbs sampler in this section.

Let $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_K)$ and $\boldsymbol{\pi} = (\pi_1,\ldots,\pi_K)$. To use a fully Bayesian approach, we need to define a prior distribution for both $\boldsymbol{\Theta}$ and $\boldsymbol{\pi}$. The hyper-prior distribution for the hyper-parameter $\boldsymbol{\theta}_k$ has been discussed in Section 3.2, and we use the same hyper-prior as before. For the mixture weights, we assume

that $(\pi_1,\ldots,\pi_K)$ has a Dirichlet distribution $D(\delta_1,\ldots,\delta_K)$ with the following density function:

$$p(\pi_1,\ldots,\pi_K) = \frac{\prod_{k=1}^K \Gamma(\delta_k)}{\Gamma(\sum_{k=1}^K \delta_k)} \prod_{k=1}^K \pi_k^{\delta_k - 1}, \tag{6.5}$$

where $\Gamma(\cdot)$ is a Gamma function. We may simply take $\delta_k = 1$.

We now discuss an MCMC algorithm, i.e., we want to sample a set of random variates from the posterior distribution of $\boldsymbol{\Theta}$ and $\boldsymbol{\pi}$. As is common in the Bayesian analysis of mixture models (Robert and Casella, 2004), this is implemented using a Gibbs sampler (Geman and Geman, 1984) where we draw from the distribution of $(\boldsymbol{\Theta},\boldsymbol{\pi})$ augmented with the latent variables $\boldsymbol{z} = (z_1,\ldots,z_M)$. Sampling from this augmented distribution is much easier than sampling from $(\boldsymbol{\Theta},\boldsymbol{\pi})$ alone. The algorithm is described as follows.

Algorithm 6.1 (Gibbs sampler for the mixture GPR models)**.** *One sweep of the algorithm includes:*

1. *sample $\boldsymbol{z}$ from $p(\boldsymbol{z}|\boldsymbol{\Theta},\mathcal{D})$ given the current value of $\boldsymbol{\Theta}$;*
2. *sample $\boldsymbol{\Theta}$ from $p(\boldsymbol{\Theta}|\boldsymbol{z},\mathcal{D})$ given the current value of $\boldsymbol{z}$.*

In the first step, let c_k be the number of observations for which $z_m = k$ over all $m = 1,\cdots,M$. Then

$$p(z_1,\cdots,z_M|\pi_1,\cdots,\pi_K) = \prod_{k=1}^K \pi_k^{c_k},$$

and

$$\begin{aligned} p(z_1,\cdots,z_M) &= \int p(z_1,\cdots,z_M|\pi_1,\cdots,\pi_K)p(\pi_1,\cdots,\pi_K)d\pi_1\cdots d\pi_K \\ &= \frac{\Gamma(K\delta)}{\Gamma(M+K\delta)} \prod_{k=1}^K \frac{\Gamma(c_k+\delta)}{\Gamma(\delta)}. \end{aligned}$$

Here we have assumed that $\delta_k = \delta$ in the prior distribution of $\boldsymbol{\pi}$ in (6.5). The conditional density function of z_m is

$$p(z_m = k|\boldsymbol{z}_{-m}) = \frac{c_{-m,k}+\delta}{M-1+K\delta},$$

where $\boldsymbol{z}_{-m} = (z_1,\ldots,z_{m-1},z_{m+1},\ldots,z_M)$, i.e., a subvector of $\boldsymbol{z}$ excluding the element z_m, and $c_{-m,k}$ is the number of observations for which $z_i = k$ for all $i \neq m$. A Gibbs subalgorithm is used to update z_m by sampling from the following density:

$$\begin{aligned} p(z_m = k|\boldsymbol{z}_{-m},\boldsymbol{y},\boldsymbol{\Theta}) &\propto p(z_m = k|\boldsymbol{z}_{-m})p(\boldsymbol{y}|\boldsymbol{\Theta},\boldsymbol{z}) \\ &\propto p(z_m = k|\boldsymbol{z}_{-m})p(\boldsymbol{y}_m|\boldsymbol{\theta}_k). \end{aligned} \tag{6.6}$$

In the above equation, we need to use the result that, if $z_m = k$, then $p(\boldsymbol{y}_m|\boldsymbol{\theta}, z_m)$ has a Gaussian distribution with the zero mean and the covariance matrix $\boldsymbol{\Psi}(\boldsymbol{\theta}_k)$.

In the second step, if we assume that the prior distributions of $\boldsymbol{\theta}_k$ are independent for $k = 1, \cdots, K$, then the conditional density function of $\boldsymbol{\Theta}$ is

$$\begin{aligned} p(\boldsymbol{\Theta}|\mathcal{D}, \boldsymbol{z}) &\propto \prod_{k=1}^{K} p(\boldsymbol{\theta}_k) \prod_{m=1}^{M} p(\boldsymbol{y}_m|\boldsymbol{\Theta}, \boldsymbol{z}) \\ &= \prod_{k=1}^{K} \left[p(\boldsymbol{\theta}_k) \prod_{m \in \{z_m = k\}} p(\boldsymbol{y}_m|\boldsymbol{\theta}_k) \right]. \end{aligned} \tag{6.7}$$

Thus $\boldsymbol{\theta}_k, k = 1, \cdots, K$ are conditionally independent given $(z_1, \cdots, z_M)$, and we can deal with each $\boldsymbol{\theta}_k$ separately. Thus, step 2 in Algorithm 6.1 is specified as sampling $\boldsymbol{\theta}_k$ from

$$p(\boldsymbol{\theta}_k|\mathcal{D}, \boldsymbol{z}) \propto p(\boldsymbol{\theta}_k) \prod_{m \in \{z_m = k\}} p(\boldsymbol{y}_m|\boldsymbol{\theta}_k)$$

independently for $k = 1, \ldots, K$. A hybrid MC algorithm, as discussed in Section 3.2 and Appendix A.4, can be used here.

6.1.2 Prediction

From the MCMC algorithm we described in the previous subsection, we could collect T samples $\{\boldsymbol{\theta}_1^{(t)}, \ldots, \boldsymbol{\theta}_K^{(t)}, \boldsymbol{z}^{(t)}, t = 1, \ldots, T\}$ after the burn-in period. In order to calculate statistical quantities involving the posterior distribution, we consider the Bayesian sampling-based approach using the set of posterior samples we have generated. Thus, the posterior predictive distribution, i.e., the conditional distribution of a future observation given the current observation, can also be obtained by the sampling-based Monte Carlo methods (see, e.g., Robert and Casella, 2004). Then, we could easily obtain the prediction, which is often the posterior predictive mean, from the posterior predictive density. Specifically, the posterior predictive density of $f_m(\boldsymbol{x})$ at $\boldsymbol{x}^*$ conditional on $\mathcal{D}$ is given by

$$\begin{aligned} p(f_m(\boldsymbol{x})|\mathcal{D}, \boldsymbol{x}^*) &= \int p(f_m(\boldsymbol{x})|\mathcal{D}, \boldsymbol{x}^*, \boldsymbol{\theta}, z_m) p(\boldsymbol{\theta}, z_m|\mathcal{D}_m) d\boldsymbol{\theta} dz_m \\ &\simeq \frac{1}{T} \sum_{t=1}^{T} p(f_m(\boldsymbol{x})|\mathcal{D}_m, \boldsymbol{x}^*, \boldsymbol{\theta}^{(t)}, z_m^{(t)}). \end{aligned} \tag{6.8}$$

The posterior distribution $p(f_m(\boldsymbol{x})|\mathcal{D}_m, \boldsymbol{x}^*, \boldsymbol{\theta}^{(t)}, z_m^{(t)})$ is Gaussian with mean and variance

$$\hat{y}_m^{*(t)} = \mathrm{E}[f_m(\boldsymbol{x}^*)|\mathcal{D}_m, \boldsymbol{\theta}^{(t)}, z_m^{(t)}], \quad \hat{\sigma}_m^{*2(t)} = \mathrm{Var}[f_m(\boldsymbol{x}^*)|\mathcal{D}_m, \boldsymbol{\theta}^{(t)}, z_m^{(t)}],$$

where $\hat{y}_m^{*(t)}$ is calculated by the formula in (2.7) but using the covariance function $k(\boldsymbol{\theta}_k^{(t)})$ if $z_m^{(t)}$ takes value of k. Similarly, we can calculate $\hat{\sigma}_m^{*2(t)}$ by using formula (2.8).

If we use the posterior predictive mean of (6.8) as a prediction, it is calculated by

$$\begin{aligned} \hat{y}_m^* &= \mathrm{E}(f_m(\boldsymbol{x})|\mathcal{D},\boldsymbol{x}^*) = \mathrm{E}\left[\mathrm{E}(f_m(\boldsymbol{x})|\mathcal{D},\boldsymbol{x}^*,\boldsymbol{\theta},z_m)\right] \\ &\simeq \frac{1}{T}\sum_{t=1}^{T}\mathrm{E}[f_m(\boldsymbol{x}^*)|\mathcal{D}_m,\boldsymbol{\theta}^{(t)},z_m^{(t)}] \\ &= \frac{1}{T}(\hat{y}_m^{*(1)}+\cdots+\hat{y}_m^{*(T)}). \end{aligned} \tag{6.9}$$

Similarly, the variance associated with the prediction can be calculated by

$$\hat{\sigma}_m^{*2} = \frac{1}{T}\sum_{t=1}^{T}\hat{\sigma}_m^{*2(t)} + \frac{1}{T}\sum_{t=1}^{T}(\hat{y}_m^{*(t)})^2 - (\hat{y}_m^*)^2. \tag{6.10}$$

The predictive variance is $(\hat{\sigma}_m^{*2}+\hat{\sigma}_\varepsilon^2)$. Prediction for a complete new batch can be calculated using a method analogous to Type II prediction discussed in Section 5.3.1.

6.2 Mixtures of GPFR models

Based on the idea of mixture GPR models considered in the previous section, a mixture GPFR model for batch data is defined by

$$y_m(t)|z_m = k \sim GPFR_k\left[\mu_{mk}(t), k_k(\boldsymbol{\theta}_k)|\boldsymbol{x}_m(t),\boldsymbol{u}_m\right], \tag{6.11}$$

where $GPFR_k$ denotes the Gaussian process functional regression model as defined in (5.8) and (5.9). Here, both the mean model and the GPR covariance model may be different for different k. For illustrative purposes, let us simply use the linear functional regression model discussed in Section 5.3 as the mean model

$$\mu_{mk}(t) = \boldsymbol{u}_m^T\boldsymbol{\beta}_k(t) = \boldsymbol{u}_m^T\boldsymbol{B}_k^T\boldsymbol{\Phi}(t), \tag{6.12}$$

where $\boldsymbol{B}_k$ is an $H\times p$ unknown B-spline coefficient matrix. Likewise, we use a squared exponential covariance function with a linear part in the GPR model. For the k-th component, $\mathrm{Cov}(y_m(t_i),y_m(t_j)|z_m=k)$ is given by

$$\begin{aligned} \Psi_k(\boldsymbol{x}_m(t_i),\boldsymbol{x}_m(t_j);\boldsymbol{\theta}_k) &= v_0^k\exp\left(-\frac{1}{2}\sum_{q=1}^{Q}w_q^k(x_{mq}(t_i)-x_{mq}(t_j))^2\right) \\ &+ a_1^k\sum_{q=1}^{Q}x_{mq}(t_i)x_{mq}(t_j) + (\sigma_\varepsilon^k)^2\delta_{ij}, \end{aligned} \tag{6.13}$$

where $\boldsymbol{\theta}_k = (w_1^k, \ldots, w_Q^k, v_0^k, a_1^k, (\sigma_\varepsilon^k)^2)^T$ are the hyper-parameters involved in the k-th component of the covariance model.

As before, the latent variable z_m is an indicator of clusters, thus we can simply assume that z_m's are independent and identically distributed with a multinomial distribution as shown in (6.3). However, for real applications such as the paraplegia data in which the patients with similar characteristics may belong to the same cluster, the distribution of z_m may additionally depend on some covariates that can determine which "cluster" the curve belongs to. We denote such covariates as $\boldsymbol{v}_m$ and use a so-called *allocation model* for z_m, where the weight π_{mk}, the probability of z_m being equal to k, is explained by the covariates $\boldsymbol{v}_m$. For example, a logistic allocation model (Shi and Wang, 2008) is defined as

$$P(z_m = k) = \pi_{mk} = \frac{\exp\{\boldsymbol{v}_m^T \boldsymbol{\gamma}_k\}}{1 + \sum_{j=1}^{K-1} \exp\{\boldsymbol{v}_m^T \boldsymbol{\gamma}_j\}}, \quad k = 1, \ldots, K-1. \tag{6.14}$$

Using the fact that $\pi_{m1} + \cdots + \pi_{mK} = 1$, we have

$$P(z_m = K) = \pi_{mK} = 1 - \sum_{j=1}^{K-1} \pi_{mj}.$$

Here $\{\boldsymbol{\gamma}_k, k = 1, \ldots, K-1\}$ are unknown parameters that need to be estimated.

Some other "spatial" allocation models can also be used; examples include the Gaussian Markov random field model and the Potts model (see, e.g., Green and Richardson, 2000; Fernandez and Green, 2002).

6.2.1 Model learning

Among several numerical methods to implement the mixture GPFR model, we focus on an EM algorithm. The methods discussed in this section are based on the work of Shi and Wang (2008).

For the m-th batch, suppose that we have observed $\boldsymbol{v}_m = (v_{m1}, \ldots, v_{mr})^T$ in addition to the data given in (5.10), which are still denoted by $\mathcal{D}_m$. Using the same notation as $\boldsymbol{y}_m = (y_{m1}, \ldots, y_{mn_m})^T$ and the others that were given before, and based on the mixture model defined in (6.11), we describe the model for data $\mathcal{D}_m$ as follows:

$$\boldsymbol{y}_m|_{z_m=k} = \boldsymbol{\mu}_{mk} + \boldsymbol{\tau}_{mk} + \boldsymbol{\varepsilon}_m, \;\; k = 1, \ldots, K, \tag{6.15}$$

where

$$\boldsymbol{\mu}_{mk} = \boldsymbol{\Phi}_m \boldsymbol{B}_k \boldsymbol{u}_m, \;\; \text{and} \;\; \tilde{\boldsymbol{\tau}}_{mk} = \boldsymbol{\tau}_{mk} + \boldsymbol{\varepsilon}_m \sim N(\boldsymbol{0}, \boldsymbol{\Psi}_{mk}(\boldsymbol{\theta}_k)),$$

$\boldsymbol{\Phi}_m$ is an $n_m \times H$ matrix with (i,h)-th element $\phi_h(t_{mi})$, and $\boldsymbol{\Psi}_{mk}(\boldsymbol{\theta}_k)$ is an

$n_m \times n_m$ covariance matrix with the (i,j)-th element $\Psi(\boldsymbol{x}_{mi},\boldsymbol{x}_{mj};\boldsymbol{\theta}_k)$ given by (6.13). Thus, the unknown parameters involved in the above mean, covariance, and allocation models in (6.14) include

$$\boldsymbol{B}=\{\boldsymbol{B}_k,\, k=1,\ldots,K\},\ \boldsymbol{\theta}=\{\boldsymbol{\theta}_k,\, k=1,\ldots,K\},\ \text{and}\ \boldsymbol{\gamma}=\{\boldsymbol{\gamma}_k,\, k=1,\ldots,K-1\}.$$

We denote that $\boldsymbol{\Theta}=(\boldsymbol{B},\boldsymbol{\theta},\boldsymbol{\gamma})$. If we use an empirical Bayesian approach to estimate the values of the hyper-parameters $\boldsymbol{\theta}$, we can estimate them jointly with other parameters by maximizing the following marginal log-likelihood:

$$l(\boldsymbol{\Theta})=\sum_{m=1}^{M}\log\{\sum_{k=1}^{K}\pi_{mk}p(\boldsymbol{y}_m|\boldsymbol{B}_k,\boldsymbol{\theta}_k,\boldsymbol{x}_m)\}, \tag{6.16}$$

where $p(\boldsymbol{y}_m|\boldsymbol{B}_k,\boldsymbol{\theta}_k,\boldsymbol{x}_m)$ is the density function of the n_m-dimensional normal distribution defined in (6.15) and π_{mk} is modeled by the multinomial logistic model in (6.14).

As a consequence of the large number of unknown parameters and the complicated covariance structure, it is very difficult to carry out the estimation directly. Instead, we consider the use of EM algorithms (Dempster et al., 1977) by treating $\boldsymbol{z}=(z_1,\ldots,z_m)$ as missing data. For convenience, we introduce a new variable z_{mk}, which takes the value 1 if $z_m=k$ and 0 otherwise. It is obvious that $\{z_{mk}\}$ and z_m are identical, and we will use $\boldsymbol{z}$ to represent either of them. The log-likelihood for the complete data $(\boldsymbol{y},\boldsymbol{z})$ is

$$l_c(\boldsymbol{\Theta})=\sum_{k=1}^{K}\sum_{m=1}^{M}z_{mk}\{\log\pi_{mk}+\log p(\boldsymbol{y}_m|\boldsymbol{B}_k,\boldsymbol{\theta}_k,\boldsymbol{x}_m)\}. \tag{6.17}$$

The EM algorithm is an iterative approach; its implementation for the mixture GPFR model is described next.

Algorithm 6.2 (EM algorithm for the mixture GPFR models)**.** *The $(i+1)$-th iteration includes the following two steps:*

1. E-step. *Calculate the conditional expectation of the log-likelihood l_c given the current value $\boldsymbol{\Theta}^{(i)}$:*

$$Q(\boldsymbol{\Theta};\boldsymbol{\Theta}^{(i)})\overset{d}{=}E_{\boldsymbol{z}}\{l_c(\boldsymbol{\Theta})|\mathcal{D},\boldsymbol{\Theta}^{(i)}\}; \tag{6.18}$$

2. M-step. *Update $\boldsymbol{\Theta}_k=(\boldsymbol{B}_k,\boldsymbol{\theta}_k,\boldsymbol{\gamma}_k)$ for $k=1,\ldots,K$ by maximizing the above $Q(\boldsymbol{\Theta};\boldsymbol{\Theta}^{(i)})$.*

The conditional log-likelihood (6.18) can be expressed by

$$\begin{aligned}
&Q(\boldsymbol{\Theta};\boldsymbol{\Theta}^{(i)})\\
=\ &\sum_{k=1}^{K}\sum_{m=1}^{M}E(z_{mk}|\mathcal{D},\boldsymbol{\Theta}^{(i)})\{\log\pi_{mk}+\log p(\boldsymbol{y}_m|\boldsymbol{B}_k,\boldsymbol{\theta}_k,\boldsymbol{x}_m)\}\\
=\ &\sum_{k=1}^{K}\sum_{m=1}^{M}\alpha_{mk}(\boldsymbol{\Theta}^{(i)})\{\log\pi_{mk}+\log p(\boldsymbol{y}_m|\boldsymbol{B}_k,\boldsymbol{\theta}_k,\boldsymbol{x}_m)\},
\end{aligned} \tag{6.19}$$

where

$$\alpha_{mk}(\boldsymbol{\Theta}) = E(z_{mk}|\mathcal{D},\boldsymbol{\Theta}) = \frac{\pi_{mk}\ p(\boldsymbol{y}_m|\boldsymbol{B}_k,\boldsymbol{\theta}_k,\boldsymbol{x}_m)}{\sum_{j=1}^{K}\pi_{mj}\ p(\boldsymbol{y}_m|\boldsymbol{B}_j,\boldsymbol{\theta}_j,\boldsymbol{x}_m)}. \tag{6.20}$$

Note that each of the $\{\boldsymbol{B}_k,\boldsymbol{\theta}_k,\boldsymbol{\gamma}_k,k=1,\ldots,K\}$ in (6.19) can be estimated separately and thus the iterative procedure usually runs quite efficiently, which indicates the advantage of using the EM algorithm in the implementation. The details are described as follows.

We first rewrite (6.19) as

$$Q(\boldsymbol{\Theta};\boldsymbol{\Theta}^{(i)}) = l_1(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{K-1}) + \sum_{k=1}^{K} l_{2k}(\boldsymbol{B}_k,\boldsymbol{\theta}_k),$$

where

$$\begin{aligned} & l_1(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{K-1}) \\ = & \sum_{m=1}^{M}\sum_{k=1}^{K}\alpha_{mk}(\boldsymbol{\Theta}^{(i)})\log\pi_{mk} \\ = & \sum_{m=1}^{M}\sum_{k=1}^{K}\alpha_{mk}(\boldsymbol{\Theta}^{(i)})\left\{\boldsymbol{v}_m^T\boldsymbol{\gamma}_k - \log\left[1+\sum_{j=1}^{K-1}\exp\{\boldsymbol{v}_m^T\boldsymbol{\gamma}_j\}\right]\right\} \end{aligned} \tag{6.21}$$

and

$$l_{2k}(\boldsymbol{B}_k,\boldsymbol{\theta}_k) = \sum_{m=1}^{M}\alpha_{mk}(\boldsymbol{\Theta}^{(i)})\log p(\boldsymbol{y}_m|\boldsymbol{B}_k,\boldsymbol{\theta}_k,\boldsymbol{x}_m). \tag{6.22}$$

Thus, maximizing $Q(\boldsymbol{\Theta};\boldsymbol{\Theta}^{(i)})$ with respect to $\boldsymbol{\gamma}_k$ for $k=1,\ldots,K-1$ is equivalent to maximizing l_1 in (6.21), where $\alpha_{mk}(\boldsymbol{\Theta}^{(i)})$ is calculated using the estimates obtained in the previous iteration. This is similar to the calculation of maximum likelihood estimates from a multinomial logistic model and can use, for example, a routine iteratively re-weighted least square algorithm (see, e.g., Green, 1984).

Parameters $(\boldsymbol{B}_k,\boldsymbol{\theta}_k)$ can be updated separately for $k=1,\ldots,K$ by maximizing l_{2k} in (6.22), in which $p(\boldsymbol{y}_m|\boldsymbol{B}_k,\boldsymbol{\theta}_k,\boldsymbol{x}_m)$ is the density function from the model (6.15). Given k and the values of $\alpha_{mk}(\boldsymbol{\Theta}^{(i)})$, the log-likelihood l_{2k} in (6.22) is analogous to the log-likelihood (5.42) for the model (5.41) in Section 5.4.1. We can therefore use a sub-iteration similar to Algorithm 5.2:

(b.1) *Update* $\boldsymbol{B}_k$ *given* $\boldsymbol{\theta}_k$*;*

(b.2) *Update* $\boldsymbol{\theta}_k$ *by maximizing* l_{2k} *given* $\boldsymbol{B}_k$.

In Step (b.1), we can update $\boldsymbol{B}_k$ by

$$\begin{aligned} \mathrm{vec}(\boldsymbol{B}_k) &= \left\{ \sum_{m=1}^{M} \alpha_{mk}(\boldsymbol{\Theta}^{(i)})(\boldsymbol{u}_m \otimes \boldsymbol{\Phi}_m^T)\boldsymbol{\Psi}_{mk}^{-1}(\boldsymbol{u}_m^T \otimes \boldsymbol{\Phi}_m) \right\}^{-1} \\ &\quad \cdot \left\{ \sum_{m=1}^{M} \alpha_{mk}(\boldsymbol{\Theta}^{(i)})(\boldsymbol{u}_m \otimes \boldsymbol{\Phi}_m^T)\boldsymbol{\Psi}_{mk}^{-1}\boldsymbol{y}_m \right\}. \end{aligned} \tag{6.23}$$

The derivation of the above equation is similar to the proof for (5.43); see the details in Shi and Wang (2008). Given $\boldsymbol{B}_k$, maximizing l_{2k} in terms of $\boldsymbol{\theta}_k$ is the same as finding the values of the hyper-parameters using an empirical Bayesian approach. The readers can refer to the details in Section 3.1. The calculation of the standard errors using this EM algorithm also can be found in Shi and Wang (2008).

Another important but difficult issue is to choose the number of components in the mixture, i.e., the value of K. As a model selection tool we could use the Bayesian information criterion (Schwarz, 1978), BIC, which is an approximation of the Bayes Factor (Kass and Raftery, 1995) . As briefly discussed in Chapter 4, the BIC is defined as

$$\mathrm{BIC} = -2l(\hat{\boldsymbol{\Theta}}) + G\log(n), \tag{6.24}$$

where $l(\boldsymbol{\Theta})$ is given by (6.16), G is the total number of parameters, and $n = n_1 + \cdots + n_M$.

A fully Bayesian approach is also often used for mixture models as mentioned in Section 6.1. The MCMC algorithm discussed there can be extended to the mixture GPFR model without much difficulty. A Gibbs sampler can be designed to generate random variates for all the unknown parameters augmented by the latent variable $\boldsymbol{z}$. From (6.17), we know that $\{\boldsymbol{B}_k, \boldsymbol{\theta}_k, \boldsymbol{\gamma}_k\}$ are conditionally independent given $\boldsymbol{z}$; therefore, they can be sampled separately. Here, the conditional density of $\boldsymbol{\gamma}_k$ given the other parameters is similar to the density arising out of a Bayesian analysis in a multinomial logistic regression model (see, e.g., Martin and Quinn, 2007; Martin et al., 2010). The conditional density of $\boldsymbol{B}_k$ also can be easily obtained from the one used in routine Bayesian analysis of a linear model. We can use the hybrid Monte Carlo algorithm discussed in Section 3.2 to generate random variates from the conditional density of $\boldsymbol{\theta}_k$. The conditional density of $\boldsymbol{z}$ (or π_{mk}'s) can be found, for example, in Neal (2000).

6.2.2 Prediction

Similar to Section 5.3.1, we also consider two types of prediction. Let us consider the new $(M+1)$-th batch and keep all the notation used in Section 5.3.1. First, suppose that we have already observed some training data in this new

batch and aim to predict the value y^* at a new data point $\boldsymbol{x}^* = \boldsymbol{x}(t^*)$. If we have already known which component the new curve belongs to, e.g., the k-th, then y^* and the observed data $\{y_{M+1,i}, i = 1,\ldots,n\}$ have the same model $GPFR_k$ as defined in (6.11). This is a single GPFR model, and thus we can use the formulae given in Section 5.3.1 to calculate both the prediction and the predictive variance. Applying the predictive mean given in (5.26) to the model $GPFR_k$, we have

$$\begin{aligned} \hat{y}_k^* &= \mathrm{E}(y^*|\mathcal{D},\hat{\mu},z_{M+1}=k) \\ &= \hat{\mu}_{M+1,k}(t^*) + \boldsymbol{H}_k^T(\boldsymbol{y}_{M+1} - \hat{\mu}_{M+1,k}(\boldsymbol{t})), \end{aligned} \tag{6.25}$$

where all the terms are evaluated at the estimates $\hat{\boldsymbol{B}}_k$ and $\hat{\boldsymbol{\theta}}_k$, i.e., the estimates for the k-th component. The overall prediction is therefore given by

$$\hat{y}^* = \sum_{k=1}^{K} \mathrm{E}(y^*|\mathcal{D},\hat{\mu},z_{M+1}=k)P(z_{M+1}=k|\mathcal{D}) = \sum_{k=1}^{K} \alpha_{M+1,k}\,\hat{y}_k^*, \tag{6.26}$$

where $\alpha_{M+1,k} = P(z_{M+1} = k|\mathcal{D})$ is the conditional probability that the $(M+1)$-th curve belongs to the k-th component. It can be estimated by

$$\hat{\alpha}_{M+1,k} = \mathrm{E}(z_{M+1,k}|\mathcal{D},\hat{\boldsymbol{\Theta}}) = \frac{\hat{\pi}_{M+1,k}p(\boldsymbol{y}_{M+1}|\hat{\boldsymbol{B}}_k,\hat{\boldsymbol{\theta}}_k,\boldsymbol{x}_{M+1})}{\sum_{j=1}^{K}[\hat{\pi}_{M+1,j}p(\boldsymbol{y}_{M+1}|\hat{\boldsymbol{B}}_j,\hat{\boldsymbol{\theta}}_j,\boldsymbol{x}_{M+1})]}, \tag{6.27}$$

with

$$\hat{\pi}_{M+1,k} = \frac{\exp\{\boldsymbol{v}_{M+1}^T\hat{\boldsymbol{\gamma}}_k\}}{1+\sum_{j=1}^{K-1}\exp\{\boldsymbol{v}_{M+1}^T\hat{\boldsymbol{\gamma}}_j\}}. \tag{6.28}$$

From Theorem 5.1, given $z_{M+1} = k$, the predictive variance is given by

$$\begin{aligned} \hat{\sigma}_k^{*2} &= \mathrm{Var}(y^*|\mathcal{D},z_{M+1}=k) \\ &= [\boldsymbol{\Psi}_k(\boldsymbol{x}^*,\boldsymbol{x}^*) - \boldsymbol{H}_k^T\boldsymbol{\Psi}_k\boldsymbol{H}_k](1+\boldsymbol{u}_{M+1}^T(\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{u}_{M+1}). \end{aligned}$$

Thus the overall variance is given by

$$\begin{aligned} \hat{\sigma}^{*2} &= \mathrm{Var}(y^*|\mathcal{D}) \\ &= E[\mathrm{Var}(y^*|\mathcal{D},z_{M+1})] + \mathrm{Var}[E(y^*|\mathcal{D},z_{M+1})] \\ &= \sum_{k=1}^{K}\hat{\alpha}_{M+1,k}\,\hat{\sigma}_k^{*2} + \sum_{k=1}^{K}\hat{\alpha}_{M+1,k}\,\hat{y}_k^{*2} - \hat{y}^{*2}. \end{aligned} \tag{6.29}$$

Second, we discuss Type II predictions, i.e., those assuming that there are no data observed for the new batch. The simplest method is to calculate the prediction using the mean model only. From model (6.15), the mean is given

by $\hat{\mu}^*_{M+1,k} = \boldsymbol{u}^T_{M+1}\hat{\boldsymbol{B}}^T_k\boldsymbol{\Phi}(t^*)$ at a new data point t^* given $z_{M+1} = k$. Thus, the overall prediction is

$$\hat{y}^* = \sum_{k=1}^{K} \hat{\pi}_{M+1,k}\boldsymbol{u}^T_{M+1}\hat{\boldsymbol{B}}^T_k\boldsymbol{\Phi}(t^*), \tag{6.30}$$

where $\hat{\pi}_{M+1,k}$ is given by (6.28).

An alternative method is to use the data observed from other batches. The basic idea is similar to the one we have discussed in Section 5.3.1, i.e., assume that the batches $1,2,\ldots,M$ provide empirical information known about the new batch. However, instead of using a uniform empirical distribution as in (5.33), we define the following one:

$$P(y^* \text{ belongs to } m\text{-th curve}) = a_m. \tag{6.31}$$

This actually defines weights for each batch when we use them to predict the new curve. It is natural that we should give relatively large weights to those that are "close" to the new curve and small weights for others. The closeness of these two curves, i.e., the curve from the current batch and the new curve, can be measured based on the allocation model for z_m and z_{M+1}. Shi and Wang (2008) suggested the use of a Kullback-Leibler divergence for this purpose, which is defined by (2.16) and thus

$$D(z_{M+1}||z_m) = \sum_{k=1}^{K} \hat{\pi}_{M+1,k}\log\frac{\hat{\pi}_{M+1,k}}{\hat{\pi}_{mk}}, \;\; m = 1,\ldots,M,$$

where $\hat{\pi}_{M+1,k}$ is calculated by (6.28), and $\hat{\pi}_{mk}$ is calculated by (6.14) but evaluated at $\hat{\boldsymbol{\gamma}}$. The weights in (6.31) can be simply defined as proportional to the inverse of the Kullback-Leibler divergence:

$$a_m \propto \frac{1}{D(z_{M+1}||z_m)}. \tag{6.32}$$

Note that the Kullback-Leibler divergence is equal to zero for two identical distributions; thus, if the distribution of z_m given in the allocation model (6.14) is the same as the distribution of z_{M+1}, then $D(z_{M+1}||z_m)$ takes the value of zero. The weight a_m will then turn to infinity. In this case, we define constant weights for all batches if the distribution of the related indicator variable z_m is identical to the one for the new batch and set all other weights as zero. Therefore, if there are M_0 such batches, we define the weights for these batches to be $1/M_0$, and the weights for the other batches to be zero.

Let y^*_m and σ^{*2}_m be the prediction and the predictive variance assuming that the new batch belongs to the m-th batch. Then they can be calculated by (6.26) and (6.29), respectively. The overall prediction for y^* is given by

$$\hat{y}^* = \sum_{m=1}^{M} a_m y^*_m, \tag{6.33}$$

and the overall predictive covariance is

$$\hat{\sigma}^{*2} = \sum_{m=1}^{M} a_m \sigma_m^{*2} + \sum_{m=1}^{M} a_m y_m^{*2} - \hat{y}^{*2}. \tag{6.34}$$

Comparing the above two equations with (5.36) and (5.37), we see that the latter used equal weights for all M batches, whereas here we are using unequal weights to cope with the heterogeneity among different batches. This is expected to give a more accurate result.

We now discuss two examples to illustrate the application of the mixture GPFR models. The first example is a simulation study and the second is the paraplegia data revisited.

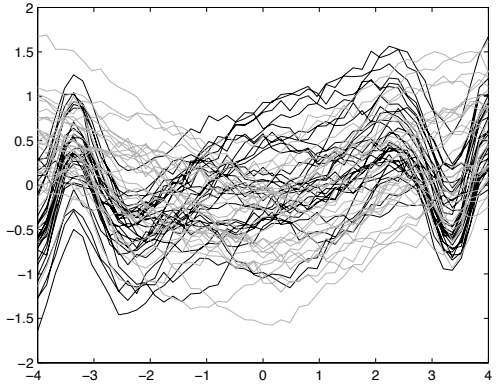


Figure 6.1 *Sixty sample curves mixed with two components (one in black and the other in gray).*

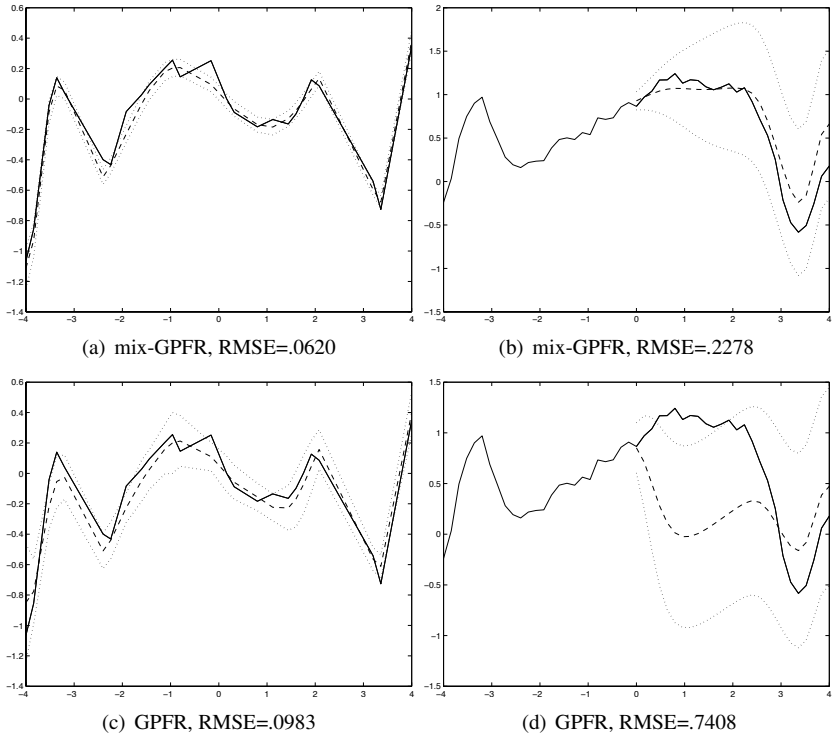
Example 6.1 (Curve prediction with mixture components)**.** We consider data simulated from a mixture model with two components. The true model is given in (6.11) with covariance function (6.13). The data points $t = t_i$ are 50 equally spaced points between -4 and 4. The functional covariate x for each batch is generated from another Gaussian process on $(-4,4)$. The two mean functions are $\mu_{m1}(t) = 0.5\sin((0.5t)^3)$ and $\mu_{m2}(t) = -3.0\exp(-t^2/8)/\sqrt{2\pi} + 0.7$, respectively. The covariance functions are

$$\Psi(x_i, x_j) = v_0 \exp\Big(-\frac{1}{2} w_1 (x_i - x_j)^2\Big) + a_1 x_i x_j + \sigma_\varepsilon \delta_{ij}.$$

Thus the parameters involved in the covariance model are $\boldsymbol{\theta} = (w_1, a_1, v_0, \sigma_\varepsilon)^T$. We take $\boldsymbol{\theta}_1 = (1.0, 0.0, 0.2, 0.0025)^T$ and $\boldsymbol{\theta}_2 = (0.5, 0.0, 0.25, 0.0025)^T$, respectively, for the two components. The allocation model is given by (6.14)

with $\gamma_1 = 2.0$. Sixty curves are generated with $v_m = 2$ for $m = 1,\ldots,30$ and $v_m = -1$ for $m = 31,\ldots,60$. They are presented in Figure 6.1.

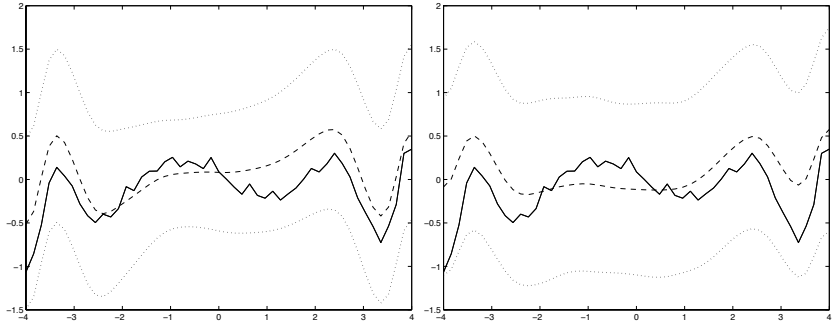
We use the model in (6.14) to analyze the simulated data. The mean model depends on t and the covariance model depends on the bivariate functional covariates (t,x). Similar to Example 5.1, we consider both interpolation and extrapolation problems for type I prediction. The former assumes that the training data are selected randomly from the whole range $t \in (-4,4)$, while the latter selects the training data in $t \in (-4,0)$. The results are presented in Figure 6.2. We also consider type II prediction, i.e., predicting a completely new curve; the results are presented in Figure 6.3. Here, we can draw the same conclusions as those gotten in Example 5.1; the results for interpolation are the best, and the next best are the ones for extrapolation; both are better than the results for the results predicted for completely new curves.



(a) mix-GPFR, RMSE=.0620 (b) mix-GPFR, RMSE=.2278

(c) GPFR, RMSE=.0983 (d) GPFR, RMSE=.7408

Figure 6.2 *Type I prediction: plots of the actual sample curves (the solid lines), the predictions (the dashed lines), and the 95% confidence intervals (the dotted lines), where panels (a) and (b) used the mixture GPFR models while (c) and (d) used a single GPFR model for problems of interpolation (a and c) and extrapolation (b and d), respectively.*



(a) mix-GPFR, RMSE=.2888 (b) GPFR, RMSE=.3655

Figure 6.3 *Type II predictions: plots of the actual sample curves (the solid lines), the predictions (the dashed lines), and the 95% confidence intervals (the dotted lines), where panel (a) used the mixture GPFR models while (b) used a single GPFR model.*

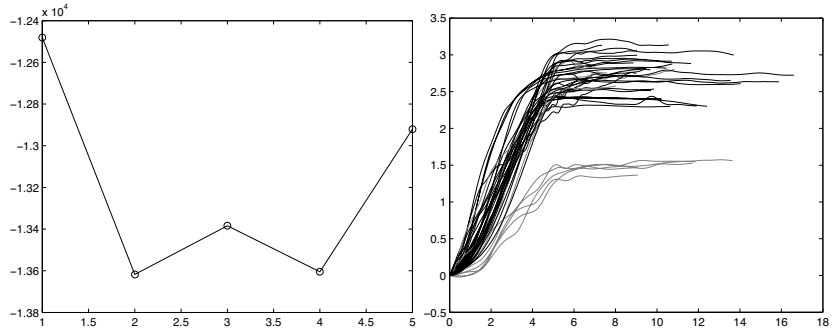
As comparison, the results using a single GPFR model are also presented in Figures 6.2 and 6.3. It is easy to see that the mixture model gives better results, which are expected by the simulation.

In the above example, mixture models give very good results for the type I predictions for both interpolation and extrapolation. This is because the mixture model evaluates the posterior probability of z_{M+1} in (6.27) which depends on the sample size, n, of the $(M+1)$-th batch. The bigger this n is, the better the estimate of the posterior probability for z_{M+1}; in other words, this probability provides accurate information regarding which cluster the new batch belongs to. In contrast, for Type II predictions, the information of the indicator variable z_{M+1} is merely given by $\boldsymbol{\nu}_{M+1}$ and therefore the prediction needs to be calculated by borrowing information from other batches. This insufficient information would undoubtedly lead to a less accurate prediction than the type I prediction. However, mixture models still perform better than a single GPFR model as we have seen before.

In practice, if there are doubts about the homogeneity of the data, we would recommend considering a mixture model. Further details can be found in Shi and Wang (2008).

Example 6.2 (Paraplegia data)**.** We revisit Example 5.2 but apply a mixture GPFR model instead. Patient's personal characteristics—height, weight, and standing-up strategy (one of three strategies is adopted in each standing-up; see Kamnik et al., 1999)—are used as covariates (i.e., $\boldsymbol{\nu}_m$) in the logistic allocation model. We first use the mixture GPFR model and the allocation model for each of the 40 curves. Based on the values of the BIC shown in Figure 6.4(a), we

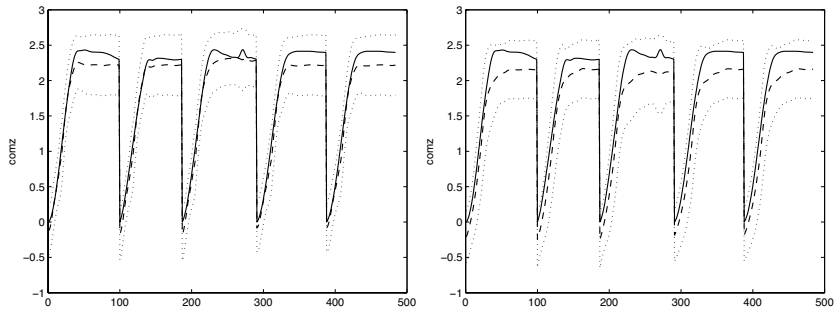
select a model with two components. As shown in Figure 6.4(b), 9 curves are in one cluster while the remaining curves are in the other cluster.



(a) (b)

Figure 6.4 *Paraplegia data. (a) The values of BIC; (b) two clusters: all curves in black belong to one cluster and the others (in gray) belong to another cluster.*

We use the data collected from seven of the eight patients as training data, and then predict the five standing-up curves for the eighth patient. This is the Type II prediction. The results for one patient are presented in Figure 6.5 as an illustration. For the selected patient, the results obtained from the mixture model are much better than the results using a single GPFR model.



(a) mix-GPFR, RMSE=0.1324 (b) GPFR, RMSE=0.2601

Figure 6.5 *Paraplegia data. Predictions for patient "mk" by using a mixture GPFR model and a single GPFR model: the solid lines stand for the real observations, the dashed lines stand for the predictions, and the dotted lines stand for the 95% confidence bands.*

Shi and Wang (2008) reported similar results but using a slightly different model. Their allocation model is based on each patient rather than each

individual curve, which seems more reasonable. They also selected a mixture model with two GPFR models. Four patients are clustered in one group and the other four are in the other group. On average, the mixture model reduces the values of RMSE by approximately 13% compared with a single GPFR model. Better results are expected if data from a wider range of patients are available.

6.3 Curve clustering

Curve clustering has been widely studied, and various methods have been proposed (see, e.g., James and Sugar, 2003; Müller, 2005; Fraley and Raftery, 2006). Most methods concern the clustering of longitudinal data or time-varying functional data, and they are basically based on the shape of the curves. Compared to those methods, we would like to discuss a different approach based on GPFR models in this section. As explained before, one advantage of the GPFR model is that it can model the regression relationship of y in terms of multidimensional spatial functional covariates $\boldsymbol{x}$. We can therefore take advantage of this and cluster the functional data based on the surface shape of y against $\boldsymbol{x}$.

In particular, we can use the mixture GPFR models to cluster curves. Suppose we have a set of data with M batches $\mathcal{D} = \{\mathcal{D}_m, m = 1,\ldots,M\}$ as described at the beginning of Section 6.2.1. We use the mixture model (6.11) to fit the data. If we fixed the number of cluster K we could use the EM algorithm discussed in Section 6.2.1 to learn about the model. Assume now we have obtained the estimates of all unknown parameters $\hat{\boldsymbol{\Theta}} = \{\hat{\boldsymbol{B}}_k, \hat{\boldsymbol{\theta}}_k, k = 1,\ldots,K;\ \hat{\boldsymbol{\gamma}}_k,\ k = 1,\ldots,K-1\}$. We can then use the following method to cluster curves.

Assume that we have observed a new batch of data (this also could be one of the M batches in the training data). The data consists of $\boldsymbol{y}^*$, the n observations of the response variable y, the corresponding functional input covariates $\boldsymbol{x}^*$, and the batch-based covariates $\boldsymbol{v}^*$ and $\boldsymbol{u}^*$. From (6.20), the posterior distribution of the latent variable z^* is given by

$$P(z^* = k|\boldsymbol{y}^*, \mathcal{D}) = \frac{\pi_k^* p(\boldsymbol{y}^*|\hat{\boldsymbol{B}}_k, \hat{\boldsymbol{\theta}}_k, \boldsymbol{x}^*)}{\sum_{j=1}^K \pi_j^* p(\boldsymbol{y}^*|\hat{\boldsymbol{B}}_j, \hat{\boldsymbol{\theta}}_j, \boldsymbol{x}^*)}, \tag{6.35}$$

where

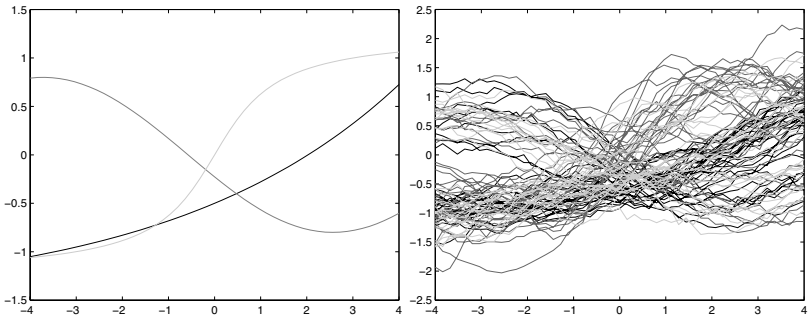
$$\pi_k^* = \frac{\exp\{\boldsymbol{v}^{*T}\hat{\boldsymbol{\gamma}}_k\}}{1 + \sum_{j=1}^{K-1} \exp\{\boldsymbol{v}^{*T}\hat{\boldsymbol{\gamma}}_j\}}.$$

For curve clustering, a curve is classified into the k^*-th cluster if $P(z^* = k|\boldsymbol{y}^*)$ takes its maximum value at $k = k^*$ for $k = 1,\ldots,K$.

The number of clusters K could be selected by the BIC as defined in (6.24). This approach usually works well. Other approaches can be found in the references given in Section 6.4.

Example 6.3 (Curve clustering)**.** Ninety curves belonging to three known classes are generated by using the mixture model (6.11). Three mean functions are given by $\mu_1(t) = \exp(t/5) - 1.3$, $\mu_2(t) = 0.8\text{atan}(0.6t)$ and $\mu_3(t) = -0.3\cos(0.8t + 4.5) - 0.2$, respectively. They are presented in Figure 6.6(a). We use the same covariance function as the one used in Example 6.1 but taking different values of $\boldsymbol{\theta}_k$ for different k.

Ninety sample curves are presented in Figure 6.6(b). Since the first two mean functions are similar to each other, it becomes very difficult to differentiate between the three clusters based merely on the shape of the curves as shown in Figure 6.6(b). Actually, if we use a mixture model including just the mean part in (6.15), i.e., if we cluster the curves based on the shape of y against t, the clustering error rate is 35.5%. This means that the curves in one of the clusters cannot be distinguished from the other two clusters at all. But if we use a mixture model of GPFR models based on the surface shape of y against the bivariate functional covariates (t,x), the clustering error rate is reduced down to just 0.5%.



(a) 3 means curves (b) 90 sample curves

Figure 6.6 *The mean curves and the sample curves of three clusters (in black, gray, and light gray, respectively).*

More discussion including comparisons with some other commonly used statistical methods can be found in Shi and Wang (2008).

6.4 Further reading and notes

How to select the number of clusters is a major issue in mixture modeling. It is particularly important in clustering analysis. Many methods have been developed to address this difficult problem. A simple way is to use a statistical quantity such as the BIC or Bayes factor; an alternative way is to assume that the number of components K is a random parameter, whose value is estimated from its posterior distribution. In this case, a reverse jump algorithm

(Green, 1995; Richardson and Green, 1997) or a birth-death MCMC algorithm (Stephens, 2000; Cappé et al., 2003) can be used. A detailed exposition of various Monte Carlo methods also can be found in Robert and Casella (2004), including the Bayesian mixture approach and its computational aspects. Other methods include Dirichlet mixture models with infinite mixture components (see, e.g., Jain and Neal, 2004) and hypothesis tests (see, e.g., Chen and Li, 2009). In addition, a Dirichlet mixture of Gaussian process regression models has been developed in machine learning (see, e.g., Görür and Rasmussen, 2010), as well as in functional data analysis (see, e.g., Petrone et al., 2009).

Chapter 7

Generalized Gaussian process regression for non-Gaussian functional data

Up until now, our discussion on Gaussian process regression and Gaussian process functional regression modeling have been based on Gaussian data. That is, the response variable y is explained by the regression function of covariates $f(\boldsymbol{x})$ along with a normal noise term. Thus, the distribution of the response variable, given $f(\boldsymbol{x})$, is normal with mean $f(\boldsymbol{x})$ and variance σ_ε^2. Based on the normal distribution assumption on the response, we treated the unknown regression function as a Gaussian process with suitably chosen mean and covariance function. This Gaussian process regression framework can be extended to non-Gaussian responses in principle, and their modeling can be done in a similar fashion to the Gaussian process regression problems discussed so far. Thus, we have the general form of the Gaussian process regression model given in (3.2) and (3.3). For exponential family distributions, we re-write the models as

$$y|f(\boldsymbol{x}) \sim g(h(f(\boldsymbol{x})), \boldsymbol{\theta}_g) \tag{7.1}$$

and

$$f(\boldsymbol{x}) \sim GPR(\boldsymbol{0}, k(\boldsymbol{\theta})|\boldsymbol{x}), \tag{7.2}$$

where g is an appropriate density function belonging to the exponential family distribution depending on parameters $\boldsymbol{\theta}_g$, $h^{-1}(\cdot)$, the inverse of $h(\cdot)$ is a link function, and *GPR* is a Gaussian process regression model as defined in (1.11). For example, in the case of a normal response, g can be a normal distribution function, and $h(\cdot)$ is just an identity function. Equations (7.1) and (7.2) describe a general form for non-Gaussian data based on a nonparametric Gaussian process regression model structure.

A typical example of non-Gaussian response is binary data, in which the response variable y takes the values of either 0 or 1; this is also closely related to the problem of classification. Suppose we have measured multidimensional functional covariates $\boldsymbol{x}$ which are associated with a binary re-

sponse y; then the regression problem is to estimate the response probability $P(y=1|\boldsymbol{x})=\pi(\boldsymbol{x})$. Mathematically, this can be formulated by g in (7.1) with a binomial distribution and a suitably chosen link function. For example, a convenient link function for binomial data can be a logit function that is defined as $\text{logit}(\pi(\boldsymbol{x}))=f(\boldsymbol{x})$, where $f(\cdot)$ can take any real values. The idea of Gaussian process binary regression is to model $f(\cdot)$ by a Gaussian process regression model as described in (7.2).

This chapter starts from a GP binary regression model. This model is defined in Section 7.1 along with suitable approaches for model learning, prediction, and variable selection. In addition, we will briefly discuss some asymptotic properties in terms of consistency at the end of this section. This idea is extended to other non-Gaussian regression models with responses from the exponential family distribution in Section 7.2. Section 7.3 describes a generalized Gaussian process functional regression model for batch data, and last, Section 7.4 focuses on a mixture model.

7.1 Gaussian process binary regression model

A generalized Gaussian process regression model can be written in terms of the general form in (7.1) and (7.2). Specifically for binary functional data, the model is given by

$$y(t)|\pi(t) \sim Bin(1,\pi(t)), \tag{7.3}$$

where $y(t)$ is a binary functional response variable that has a binomial distribution with parameters 1 and $\pi(t)$. If we use a logit link, $\text{logit}[\pi(t)]=f(t)$, then the latent variable $f(t)$ can be modeled by the GPR model given in (7.2), or the following form in the case of functional data:

$$f(t) \sim GPR[0,k(\boldsymbol{\theta})|\boldsymbol{x}(t)],$$

where $\boldsymbol{x}(t)$ are Q-dimensional functional covariates. This defines a nonparametric binary regression model, called a *GP binary regression model*, between the binary response variable $y(t)$ and the functional covariates $\boldsymbol{x}(t)$ through a functional latent variable $f(t)$. If we observe data $(y(t),\boldsymbol{x}(t))$ at data points $t=1,\ldots,n$, we can write the discrete form of the model as

$$y_i|\pi_i \overset{\text{ind}}{\sim} Bin(1,\pi_i), \;\; i=1,\ldots,n. \tag{7.4}$$

Using a logit link, the latent variable f_i is defined by $f_i=f(\boldsymbol{x}_i)=\text{logit}[\pi_i(\boldsymbol{x}_i)]$. From the Gaussian process binary regression model (7.2), we have

$$\boldsymbol{f}=(f_1,\ldots,f_n) \sim N(\boldsymbol{0},\boldsymbol{K}(\boldsymbol{\theta})), \tag{7.5}$$

where $\boldsymbol{K}$ is defined by a covariance function $k(\boldsymbol{\theta})$ and the (i,j)-th element of $\boldsymbol{K}$ is given by $\text{Cov}(f_i,f_j)=k(\boldsymbol{x}_i,\boldsymbol{x}_j;\boldsymbol{\theta})$. When f_i or π_i is given, the density

function of $\boldsymbol{y}$ can be written by

$$p(\boldsymbol{y}|\boldsymbol{f}) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i} = \prod_{i=1}^{n} \frac{\exp(f_i y_i)}{1+\exp(f_i)}. \tag{7.6}$$

Thus, the marginal distribution of $\boldsymbol{y}$ is given by

$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{x},\boldsymbol{\theta})d\boldsymbol{f}. \tag{7.7}$$

In this section, we discuss two approaches on model learning and predicting. One is based on the empirical Bayesian method discussed in Section 3.1 and a Laplace approximation (Williams, 1998; Rasmussen and Williams, 2006). The other is based on a fully Bayesian procedure as discussed in Section 3.2 through an MCMC algorithm (c.f. Neal, 1999; Shi et al., 2003; Choudhuri et al., 2007). Comments on other methods can be found in Section 7.5.

7.1.1 Empirical Bayesian learning and Laplace approximation

We have discussed the empirical Bayesian learning method in Section 3.1. The idea is to find the values of the hyper-parameters $\boldsymbol{\theta}$ by maximizing the marginal likelihood. From (7.7), the marginal log-likelihood is

$$l(\boldsymbol{\theta}) = \log(p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})) = \log\left(\int \exp(\gamma(\boldsymbol{f}))d\boldsymbol{f}\right), \tag{7.8}$$

where

$$\gamma(\boldsymbol{f}) = \log(p(\boldsymbol{y}|\boldsymbol{f})) + \log(p(\boldsymbol{f}|\boldsymbol{x},\boldsymbol{\theta})). \tag{7.9}$$

It is easy to calculate the first two derivatives of $\gamma(\boldsymbol{f})$ in terms of $\boldsymbol{f}$. For the model with the logit link function, $p(\boldsymbol{y}|\boldsymbol{f})$ is given by (7.6), and thus we have

$$\frac{\partial\gamma(\boldsymbol{f})}{\partial\boldsymbol{f}} = \boldsymbol{y} - \boldsymbol{\pi} - \boldsymbol{K}_n^{-1}\boldsymbol{f}, \tag{7.10}$$

$$\frac{\partial^2\gamma(\boldsymbol{f})}{\partial\boldsymbol{f}\partial\boldsymbol{f}^T} = -\boldsymbol{A} - \boldsymbol{K}_n^{-1}, \tag{7.11}$$

where $\boldsymbol{\pi} = (\pi_1,\ldots,\pi_n)^T$ and $\boldsymbol{A} = \mathrm{diag}(\pi_1(1-\pi_1),\ldots,\pi_n(1-\pi_n))$. Using the Laplace method, the integral in (7.8) can be approximated by (see Appendix A.9)

$$l(\boldsymbol{\theta}) = \log\int \exp(\gamma(\boldsymbol{f}))d\boldsymbol{f} \approx \gamma(\hat{\boldsymbol{f}}) + \frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{K}_n^{-1} + \boldsymbol{A}|, \tag{7.12}$$

where $\hat{\boldsymbol{f}}$ is the maximizer of $\gamma(\boldsymbol{f})$. Empirical Bayes estimate of $\boldsymbol{\theta}$ can therefore be calculated by maximizing (7.12). However, $\gamma(\boldsymbol{f})$ in (7.9) depends on $\boldsymbol{\theta}$. So we need an iterative method, starting with initial values of $\boldsymbol{\theta}$ and then updating $\boldsymbol{f}$ and $\boldsymbol{\theta}$ in turn until both converge.

Algorithm 7.1 (Empirical Bayesian learning for the GP binary regression model)**.** *Each iteration includes the following two steps:*

1. *Update* $\boldsymbol{f}$ *by maximizing* $\gamma(\boldsymbol{f})$ *in* (7.9) *given* $\boldsymbol{\theta}$*;*
2. *Update* $\boldsymbol{\theta}$ *by maximizing* $l(\boldsymbol{\theta})$ *in* (7.12) *given* $\boldsymbol{f}$*.*

The first two derivatives of $\gamma(\boldsymbol{f})$ in terms of $\boldsymbol{f}$ have been given in (7.10) and (7.11), respectively. The first two derivatives of $l(\boldsymbol{\theta})$ can be obtained easily from (3.8) and (3.9). Thus, we can use an efficient Newton-Raphson method in each step.

7.1.2 Prediction

The calculation of the prediction from the Gaussian process binary regression model is much more complicated than from the Gaussian process regression model since the analytical form no longer exists. Here we provide an estimate based on a Laplace approximation. Let $\boldsymbol{x}^*$ be a new point and $\mathcal{D}$ be the training data. We need to calculate the value of $\pi(\boldsymbol{x}^*)$ or $f^* = f(\boldsymbol{x}^*)$. Based on the assumption that $(f_1 \ldots, f_n, f^*)$ come from the same Gaussian process in (7.2), we have the following results:

$$\begin{aligned} \mathrm{E}(f^*|\boldsymbol{f}) &= \boldsymbol{K}^{*T}\boldsymbol{K}_n^{-1}\boldsymbol{f}, & (7.13)\\ \mathrm{Var}(f^*|\boldsymbol{f}) &= k(\boldsymbol{x}^*,\boldsymbol{x}^*) - \boldsymbol{K}^{*T}\boldsymbol{K}_n^{-1}\boldsymbol{K}^*, & (7.14) \end{aligned}$$

where $\boldsymbol{K}^* = (k(\boldsymbol{x}^*,\boldsymbol{x}_1),\ldots,k(\boldsymbol{x}^*,\boldsymbol{x}_1))^T$. The above equations are analogous to (2.7) and (2.8) but there is no error term here. Thus, we have

$$\begin{aligned} \mathrm{E}(f^*|\mathcal{D},\boldsymbol{x}^*) &= \mathrm{E}_{\boldsymbol{f}}\left[\mathrm{E}(f^*|\boldsymbol{f},\mathcal{D},\boldsymbol{x}^*)\right] \\ &= \boldsymbol{K}^{*T}\boldsymbol{K}_n^{-1}\mathrm{E}(\boldsymbol{f}|\mathcal{D}). & (7.15) \end{aligned}$$

It is not possible to get an analytical form of $\mathrm{E}(\boldsymbol{f}|\mathcal{D})$, but it can be estimated numerically, for instance, a Laplace approximation similar to (7.12) or via an MCMC approach as discussed in the next section. An alternative way is to use the posterior mode $\hat{\boldsymbol{f}}$ as a replacement for the posterior mean in the prediction, where $\hat{\boldsymbol{f}}$ can be obtained using the iterative approach specified in Algorithm 7.1. This leads to the following result:

$$\hat{f}^* = \boldsymbol{K}^{*T}\boldsymbol{K}_n^{-1}\hat{\boldsymbol{f}}. \quad (7.16)$$

In addition, replacing the posterior mean with the posterior mode, the predictive variance of f^* can be approximated by

$$\begin{aligned} & \mathrm{Var}(f^*|\mathcal{D},\boldsymbol{x}^*) \\ = \; & \mathrm{E}_{\boldsymbol{f}}\left[\mathrm{Var}(f^*|\boldsymbol{f},\mathcal{D},\boldsymbol{x}^*)\right] + \mathrm{Var}_{\boldsymbol{f}}\left[\mathrm{E}(f^*|\boldsymbol{f},\mathcal{D},\boldsymbol{x}^*)\right] \\ \approx \; & k(\boldsymbol{x}^*,\boldsymbol{x}^*) - \boldsymbol{K}^{*T}\boldsymbol{K}_n^{-1}\boldsymbol{K}^* + \boldsymbol{K}^{*T}\boldsymbol{K}_n^{-1}\left[\mathrm{Var}(\hat{\boldsymbol{f}})\right]\boldsymbol{K}_n^{-1}\boldsymbol{K}^*. \end{aligned}$$

Using the fact that $\mathrm{Var}(\hat{\boldsymbol{f}}) \approx (A + \boldsymbol{K}_n^{-1})^{-1}$ from (7.11) (when sample size is large), we have

$$\begin{aligned} & \mathrm{Var}(f^* | \mathcal{D}, \boldsymbol{x}^*) \\ = \; & k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{K}^{*T} \boldsymbol{K}_n^{-1} \boldsymbol{K}^* + \boldsymbol{K}^{*T} \boldsymbol{K}_n^{-1} (\boldsymbol{A} + \boldsymbol{K}_n^{-1})^{-1} \boldsymbol{K}_n^{-1} \boldsymbol{K}^* \\ = \; & k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{K}^{*T} (\boldsymbol{A}^{-1} + \boldsymbol{K}_n)^{-1} \boldsymbol{K}^*. \end{aligned} \tag{7.17}$$

Note that in the matrix inversion in (7.17) we used the equation (A.16) in Appendix A.8 to simplify the formula.

Once $\hat{f}^*$ is calculated, we can estimate the value of π^*, the predictive probability. Consequently, the prediction of y^* takes either the value of 1 if $\pi^* \geq 0.5$, or 0 otherwise.

7.1.3 Variable selection

Methods of model selection discussed in Chapter 4 can be applied and extended to the GP binary regression model. Here, we provide a description of the method due to its ever-growing importance in many applications. Specifically, due to the fast development of high technology, many problems involve massive datasets in areas such as computational biology, health studies, and financial engineering. Variable selection is therefore becoming more and more important for regression and classification; see, e.g., Example 7.1.

The penalized technique discussed in Section 4.3 can be applied to the GP binary regression model. As before, a penalty function for hyper-parameters $\boldsymbol{\theta}$ is imposed on the negative log-likelihood, that is,

$$l_p(\boldsymbol{\theta}) = -l(\boldsymbol{\theta}) + \sum_{q=1}^{Q} P_{\lambda_n}(w_q), \tag{7.18}$$

where $l(\boldsymbol{\theta})$ is given in (7.8), and $P_{\lambda_n}(w_q)$ is a general term for penalties. Let us now select a stationary covariance function; specifically, let us focus on the squared exponential kernel given in (4.6). The coefficient w_q therefore corresponds to the q-th covariate x_q. If $w_q = 0$, the covariate x_q has no influence on the response variable and, hence, can be removed from the model. The penalty $P_{\lambda_n}(w_q)$ in the above penalized log-likelihood forces those w_q's of small values to be zero. All the penalty functions discussed in Section 4.3 can also be used here.

This penalized framework can be applied to the GP binary regression model in two steps. First, a generalized cross-validation (GCV) method is used to find the optimal value of the tuning parameter (regularizer), λ_n. And second, once λ_n is fixed, the estimates of $\boldsymbol{\theta}$ are determined by minimizing the penalized log-likelihood in (7.18). To use GCV, we need to calculate the error rate, defined as the proportion of incorrect predictions, rather than the RMSE

used in Section 4.3, since y takes the values of either 0 or 1. Predictions are calculated using the methods given in the previous subsection. Error rate is the proportion of the predictions that are classified wrongly compared with the true observations.

In model learning, the first term of the penalized log-likelihood in (7.18), i.e., the log-likelihood $l(\boldsymbol{\theta})$, is approximated by (7.12). Thus, to obtain a penalized estimate of $\boldsymbol{\theta}$, we also need an iterative method. In this case, we can still apply Algorithm 7.1, but the second step should be replaced by the following one:

2. (modified step) *Update* $\boldsymbol{\theta}$ *by minimizing* $l_p(\boldsymbol{\theta})$ *in* (7.18) *given* $\boldsymbol{f}$.

We now provide a real application example with leukemia patients to demonstrate this variable selection procedure.

Example 7.1 (Leukemia cancer data)**.** The leukemia cancer data (Golub et al., 1999) is a typical example with a binary response variable and a very large number of covariates. This example uses gene microarray expressions as predictors to classify two types of leukemia cancer. The dataset contains 7129 genes (input covariates), and just 72 samples. The whole dataset is split into two parts. One part is used to learn about the model and has 38 samples, of which 27 cases are Acute Lymphoblastic Leukemia (ALL) and 11 cases are Acute Myeloid Leukemia (AML); the other part is used as the test dataset and has 34 samples, of which 20 cases are ALL and 14 cases are AML. The response variable was coded as 1 and 0, denoting the two types of leukemia cancer ALL and AML, respectively.

In this example, we have high-dimensional input covariates, 7129 gene microarray expressions, and it is essential to select suitable input covariates or reduce the number of covariates that characterize the types of leukemia cancer without loss of much information. However, since the dimension of input covariates is too high (7129), numerical problems are expected to occur. Therefore, in order to make the optimization of the penalized likelihood function more stable, we use a prescreened dataset that includes 350 genes (Golub et al., 1999). The heat map for the training samples can be found in Figure 7.1.

We first consider the penalized GP binary regression model with the LASSO penalty function; a fivefold generalized cross-validation is used to select the value of the tuning parameter and then estimate the hyper-parameters following the procedure discussed earlier in this subsection. Each w_q is corresponding to a covariate, which is a gene in this example. The covariates or the genes with the zero estimates will be removed. By using a LASSO penalty function, 30 covariates remain in the model, meaning the associated 30 genes are selected. Using those 30 genes to classify those two types of leukemia cancers, the error rate is 4 out of 38 for the training data and 3 out of 34 for the test data, a pretty good result.

By making use of the Elastic NET penalty (Zou and Hastie, 2005), we just

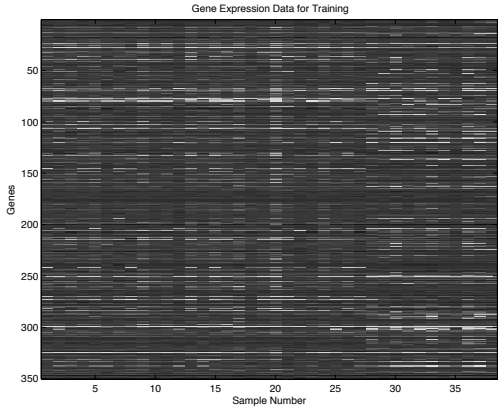


Figure 7.1 *Heat map for leukemia cancer training samples of prescreened data with 350 genes (27 cases of ALL and 11 cases of AML).*

need 22 genes and obtain improved results. The error rate is 2 out of 38 for the training data using fivefold GCV and 1 out of 34 for the test data. The interested reader can refer to Yi (2009) and Yi et al. (2011) for further details.

Figure 7.2 shows the heat map for leukemia cancer data with selected genes by using the penalized GP binary regression with the LASSO and elastic NET penalty functions. Both look much more obvious on distinguishing two classes than the prescreened data with 350 genes shown in Figure 7.1.

7.1.4 MCMC algorithm

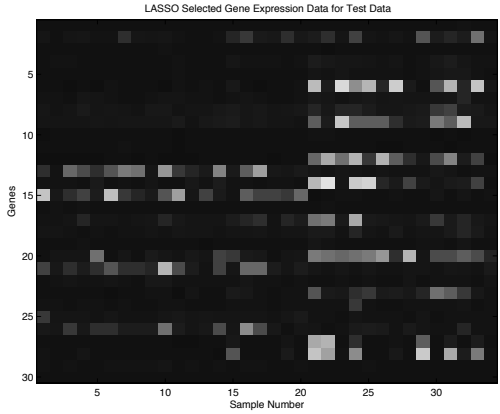
To use a fully Bayesian approach in order to fit a GP binary regression model, we need to assume a hyper-prior $p(\boldsymbol{\theta})$ for the hyper-parameters $\boldsymbol{\theta}$. As before, we use the same hyper-prior discussed in Sections 3.2 and 6.1. For the GP binary regression models, the key issue to design an efficient MCMC algorithm is how to deal with the latent variable $\boldsymbol{f} = (f_1, \ldots, f_n)$. A Gibbs sampler is designed to augment the parameters by the above latent variables. The algorithm is specified as follows.

Algorithm 7.2 (Gibbs sampler for the GP binary regression model)**.** *One sweep of the algorithm includes the following steps:*

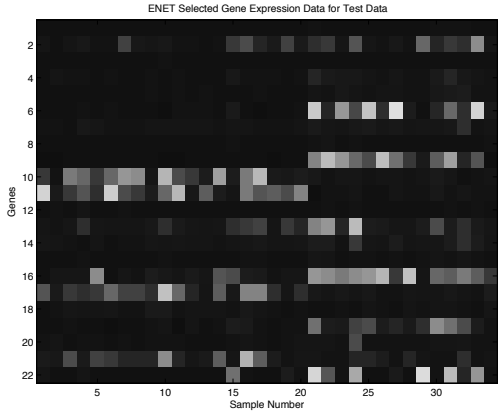
1. *Sample* $\boldsymbol{\theta}$ *from* $p(\boldsymbol{\theta}|\boldsymbol{f}, \mathcal{D})$*;*
2. *Sample* $\boldsymbol{f}$ *from* $p(\boldsymbol{f}|\boldsymbol{\theta}, \mathcal{D})$.

The conditional density function of $\boldsymbol{\theta}$ is given by

$$p(\boldsymbol{\theta}|\boldsymbol{f}, \mathcal{D})) \propto p(\boldsymbol{\theta})p(\boldsymbol{f}|\boldsymbol{\theta}). \tag{7.19}$$



(a)



(b)

Figure 7.2 *Heat map for leukemia cancer test data (20 cases of ALL and 14 cases of AML): (a) 30 genes selected by penalized GP binary regression with the LASSO penalty and (b) 22 genes selected by the method with the elastic NET penalty.*

Here, $p(\boldsymbol{\theta})$ is the prior density of $\boldsymbol{\theta}$. The latent vector $\boldsymbol{f}$ comes from a noise-free Gaussian process, i.e., the normal distribution given by (7.5). Thus, the above density function is similar to the density function in (3.11). We can therefore use the hybrid MC algorithm given in Appendix A.4 to sample $\boldsymbol{\theta}$ from its conditional distribution (7.19) in step 1.

If a logit link function is used, the conditional density function of $\boldsymbol{f}$ is

expressed by

$$\begin{aligned} p(\boldsymbol{f}|\boldsymbol{\theta},\mathcal{D}) &= p(\boldsymbol{f}|\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{f}) \\ &= \varphi_n(\boldsymbol{f};\boldsymbol{0},\boldsymbol{K}_n(\boldsymbol{\theta}))\prod_{i=1}^{n}\frac{\exp(f_i y_i)}{1+\exp(f_i)}. \end{aligned} \tag{7.20}$$

This is log-concave for f_i (i.e., the log-density is concave); it has been shown that a Gibbs sampler with adaptive rejection sampling (Gilks and Wild, 1997) is a very efficient way to generate random variates from a log-concave density function. This can be used in step 2 of Algorithm 7.2.

After the Gibbs sampler in Algorithm 7.2 converges within a burn-in period, we take a set of random variates

$$\{\boldsymbol{\theta}^{(t)},\boldsymbol{f}^{(t)},t=1,\ldots,T\}.$$

For each pair of $(\boldsymbol{\theta}^{(t)},\boldsymbol{f}^{(t)})$, we can calculate the prediction and the predictive variance at a new data point $\boldsymbol{x}^*$ from (7.16) and (7.17), which are

$$\hat{f}^{*(t)} = \boldsymbol{K}^{*T}\boldsymbol{K}_n^{-1}\boldsymbol{f}^{(t)}, \tag{7.21}$$

$$\hat{\sigma}^{*(t)2} = \left[k(\boldsymbol{x}^*,\boldsymbol{x}^*)-\boldsymbol{K}^{*T}(\boldsymbol{A}^{-1}+\boldsymbol{K}_n)^{-1}\boldsymbol{K}^*\right]_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(t)}}. \tag{7.22}$$

The covariance function depends on the parameter $\boldsymbol{\theta}$ which is evaluated at $\boldsymbol{\theta}^{(t)}$ in the above equations. Thus, the overall prediction and predictive variance are given by

$$\begin{aligned} \hat{f}^* &= \frac{1}{T}\sum_{t=1}^{T}\hat{f}^{*(t)}, \\ \hat{\sigma}^{*2} &= \frac{1}{T}\sum_{t=1}^{T}\hat{\sigma}^{*(t)2}+\frac{1}{T}\sum_{t=1}^{T}(\hat{f}^{*(t)})^2-\hat{f}^{*2}. \end{aligned}$$

The prediction of π^* and t^* can be calculated accordingly.

If we use a probit link function, the model is defined by

$$y_i=\begin{cases} 0 & \text{if } f_i<0 \\ 1 & \text{if } f_i\geq 0 \end{cases},$$

and $\pi_i=P(y_i=1)=P(f_i\geq 0)$. We can still use Gibbs sampler Algorithm 7.2 for the model with the probit link function. However, the conditional density function $p(\boldsymbol{y}|\boldsymbol{f})$ in (7.20) needs to be replaced by the following equation:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{f}) &= \prod_{i=1}^{n}p(y_i|f_i)=\prod_{i=1}^{n}[P(f_i\geq 0)y_i+P(f_i<0)(1-y_i)] \\ &= \prod_{i=1}^{n}[(1-y_i)+(2y_i-1)P(f_i\geq 0)], \end{aligned}$$

where $P(f_i\geq 0)$ is calculated from the normal distribution (7.5). Further discussion about the model with the probit link is given in Choudhuri et al. (2007).

7.1.5 *Posterior consistency*

Posterior consistency in GP binary regression models has been studied in Ghosal and Roy (2006) and in Choi (2007). As discussed in Chapter 2, sufficient conditions to achieve posterior consistency are described in two respects: one with the prior positivity on a suitable neighborhood of the true parameter, usually the Kullback-Leibler neighborhood, and the other with the existence of uniformly consistent tests. These two basic conditions are derived from the Schwartz's theorem (Schwartz, 1965), and there are variants of Schwartz's theorem, depending on the model structure as well as topology for the parameter space under consideration. For example, Choudhuri et al. (2004) provided a general theorem for in-probability consistency of the posterior distribution based on independent but nonidentically distributed observations. Ghosal and Roy (2006) applied the result of Choudhuri et al. (2004) to the GP binary regression model by verifying two sufficient conditions. Specifically, Ghosal and Roy (2006) showed that the posterior distribution is consistent in the L_1-distance between two response probability functions in the binary regression model if the covariance function has derivatives up to a desired order and the bandwidth parameter satisfies the following condition:

$$\Pi\left(\pi : \int |\pi(x) - \pi_0(x)|dx > \varepsilon \middle| y_1, \ldots, y_n\right) \to 0,$$

where $y_i \sim \text{Bin}(1, \pi(x_i))$ for $i = 1, \ldots, n$. Here $\pi(x)$ is induced by a Gaussian process prior in that $h^{-1}(\pi(x)) = f(x)$, where h^{-1} is a known link function and $f(x)$ is a Gaussian process. Detailed assumptions and the technical statement of the theorem can be found in Ghosal and Roy (2006).

Alternatively, extending the work of Ghosal and Roy (2006), Choi (2007) also established consistency of the posterior distribution under modified assumptions on the smoothness of the covariance function from Ghosal and Roy (2006). Note that the condition on the prior positivity is independent of the model structure, and it only involves the small ball probabilities of the process. Thus, the extension of Choi (2007) was mainly involved with the construction of the uniformly consistent test for the binary regression model with a different technique and a modified condition. Also note that although it is plausible to assume additional conditions on the hyper-parameters of the covariance function (see, e.g., Choi, 2007), posterior consistency has been mainly achieved with isotropic covariance functions when all of the hyper-parameters other than the bandwidth parameter are assumed to be fixed.

Furthermore, general results about rates of contraction of posterior distributions have been studied in van der Vaart and van Zanten (2008a); they also considered smooth classification problems under logistic or probit link functions combined with various Gaussian process priors.

7.2 Generalized Gaussian process regression

Let $\{y(t), t \in \mathcal{T}\}$ be functional or longitudinal data that have a distribution from the exponential family with the following density function:

$$p(y(t)|\alpha,\phi) = \exp\left\{\frac{y(t)\alpha - b(\alpha)}{a(\phi)} + c(y(t),\phi)\right\}, \tag{7.23}$$

where α and ϕ are canonical and dispersion parameters, respectively. From known properties of the exponential family distribution, we have

$$\begin{aligned} \mathrm{E}(y(t)) &= b'(\alpha), \\ \mathrm{Var}(y(t)) &= b''(\alpha)a(\phi), \end{aligned}$$

where $b'(\alpha)$ and $b''(\alpha)$ are the first two derivatives of $b(\alpha)$ in terms of α. For functional variable $y(t)$, the parameters (α,ϕ) are usually also dependent on t; thus, we have $(\alpha(t),\phi(t))$. We denote the exponential family distribution in (7.23) as

$$y(t) \sim EF(\alpha(t),\phi(t)). \tag{7.24}$$

Suppose that $\boldsymbol{x}(t)$ is a vector of Q-dimensional functional covariates. We are interested in finding the regression relationship between $y(t)$ and $\boldsymbol{x}(t)$. Let $h^{-1}(\cdot)$ be a link function; we have

$$\mathrm{E}(y(t)) = h(f(t)), \text{ or } h^{-1}(\mathrm{E}(y(t))) = f(t) \tag{7.25}$$

where $h^{-1}(\cdot)$ is the inverse function of $h(\cdot)$. A generalized regression model usually consists of modeling $f(\cdot)$ by $\boldsymbol{x}(t)$. Once the transformation with a link function is carried out, $f(\boldsymbol{x}(t))$ can take any real values. Thus, many regression models can be used to model $f(t)$. For example, a linear regression model (McCullagh and Nelder, 2000) is defined by $f(t) = \boldsymbol{x}^T(t)\boldsymbol{\beta}$. A generalized linear mixed model (Breslow and Clayton, 1993) is defined by

$$f(t) = \boldsymbol{x}_1^T(t)\boldsymbol{\beta} + \boldsymbol{x}_2^T(t)\boldsymbol{\gamma},$$

where $\boldsymbol{\gamma}$ is a random vector that takes into account the random effects. A generalized Gaussian process regression model is defined by using the Gaussian process regression model in (7.2), i.e.,

$$f(t) \sim GPR(\boldsymbol{0}, k(\boldsymbol{\theta})|\boldsymbol{x}(t)).$$

Notice that the GP binary regression is a special case of the generalized GPR we considered in this section. Indeed, the binomial distribution is only one of the exponential family distributions defined in (7.24). Other commonly used distributions belonging to the exponential family include the Poisson distribution, the normal distribution, and the exponential distribution. We have already discussed the normal distribution with Gaussian process regression, and the binomial distribution with GP binary regression. Here, we show two additional examples.

Example 7.2 (Poisson distribution)**.** Suppose that $y(t) \sim Poisson(\lambda(t))$, and we have a set of observations $y_1, \ldots, y_n$. They have the distribution

$$y_i \sim Poisson(\lambda_i), \;\; \lambda_i = \lambda(t_i), \; i = 1, \ldots, n.$$

If we use a log link function $\log(\lambda(t)) = f(t)$, and model $f(t)$ by the Gaussian process regression model in (7.2). Then, $\boldsymbol{f} = (f_1, \ldots, f_n)^T$ has a multivariate normal distribution as given in (7.5). The density function of $p(\boldsymbol{y}|\boldsymbol{f})$ is given by

$$p(\boldsymbol{y}|\boldsymbol{f}) = \prod_{i=1}^{n} \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!} = \frac{\exp[\sum_{i=1}^{n}(f_i y_i - e^{f_i})]}{\prod_{i=1}^{n}(y_i!)}.$$

By replacing the above density function with $p(\boldsymbol{y}|\boldsymbol{f})$ in (7.7) for the GP binary regression model, we can use the empirical Bayesian learning method described in Sections 7.1.1 and 7.1.2 to estimate the values of the hyperparameters $\boldsymbol{\theta}$ and calculate predictions. Variable selection and the MCMC algorithm discussed in the previous section can also be modified and applied to the GP Poisson model.

Example 7.3 (Gaussian process ordinal regression)**.** Suppose that $y(t)$ is an ordinal response variable, taking values from $\{1, 2, \ldots, r\}$ of r ordered categories. If we use a probit link function, we can define a Gaussian process ordinal regression model (Chu and Ghahramani, 2005) as follows:

$$Y = j \quad \text{if} \;\; b_{j-1} < f(\boldsymbol{x}) \leq b_j,$$

where $b_0 = -\infty$, $b_r = \infty$, and $b_j \in \mathcal{R}$ for $j = 1, \ldots, r-1$ are the thresholds; $f(\boldsymbol{x})$ has a GPR model given in (7.2). Thus, if we have observed the data $\{(y_i, \boldsymbol{x}_i) \; i = 1, \ldots, n\}$, we have

$$p(\boldsymbol{y}|\boldsymbol{f}) = \prod_{i=1}^{n} p(y_i|f_i) = \prod_{i=1}^{n} P(b_{y_i-1} < f_i \leq b_{y_i})$$

and $\boldsymbol{f} = (f_1, \ldots, f_n) \sim N(0, \boldsymbol{K}(\boldsymbol{\theta}))$ where $\boldsymbol{K}(\boldsymbol{\theta})$ is the covariance matrix of $\boldsymbol{f}$ calculated from the covariance function $k(\cdot, \cdot)$ in (7.2).

When $r = 2$, this is actually the case of binary data we have discussed in Section 7.1. The methods discussed there can be extended to this example for the cases of $r > 2$. The detailed results can be found in Chu and Ghahramani (2005).

Inference for a generalized GP regression model with other distributions from the exponential family can be treated similarly.

7.3 Generalized GPFR model for batch data

We now consider the case of repeated measurements in which the observations are arranged into M batches. Based on the generalized GPR model discussed

in the previous section, we simply extend the method to cover functional data analysis and represent the model structure in the format of batches. This is called the generalized Gaussian process functional regression model (Wang and Shi, 2011); it is defined by

$$\begin{aligned} y_m(t)|\alpha_m(t),\phi_m(t) &\sim EF(\alpha_m(t),\phi_m(t)), && (7.26)\\ \mathrm{E}(y_m(t)) &= h(f_m(t)), && (7.27)\\ f_m(t) &\sim GPFR[\mu,k(\boldsymbol{\theta})|\boldsymbol{x}_m(t),\boldsymbol{u}_m], && (7.28) \end{aligned}$$

for $m = 1,\ldots,M$. Here, (7.26) defines a distribution from the exponential family with functional response variables $y_m(t)$ and whose density is given in (7.24). A link function $h^{-1}(\cdot)$ is used to define a transformation of the mean such that $f(t)$ can take any real values. The GPFR model, defined in (5.8), can be used to model $f_m(t)$ by functional covariates $\boldsymbol{x}_m(t)$ and scalar covariates $\boldsymbol{u}_m$. In the above model, we assume that the $y_m(t)$'s are independent for $m = 1,\ldots,M$, and so are the $f_m(t)$'s. The generalized GPFR model is a nature extension of the GPFR model we have discussed in Section 5.2 when the response variable $y_m(t)$ has a non-Gaussian distribution from the exponential family.

Suppose that we have observed a set of data for each batch. In the m-th batch, the response variable $y_m(t)$ and the covariates $\boldsymbol{x}_m(t)$ are observed at data points $t = t_{mi}, i = 1,\ldots,n_m$. We denote the data collected in the m-th batch as $\mathcal{D}_m$ and the whole batch data as $\mathcal{D}$, where the format of $\mathcal{D}_m$ is the same as the one defined in (5.10). Let $f_{mi} = f_m(t_{mi})$, the latent variable at t_{mi}, and $\boldsymbol{f}_m = (f_{m1},\ldots,f_{mn_m})$. We can easily write the density function $p(\boldsymbol{y}_m|\boldsymbol{f}_m)$. Note that the model-fitting procedure is similar to the case of the generalized GPR model except for the batch structure. Hence, the same principle can be applied in order to implement the generalized GPFR model. For the sake of clarity, we focus on the case of a binary GPFR model below. It can be extended to other cases such as Poisson regression.

Similar to (7.6), with a logit link function for binary data, we have

$$p(\boldsymbol{y}_m|\boldsymbol{f}_m) = \prod_{i=1}^{n_m} \frac{\exp(f_{mi}y_{mi})}{1+\exp(f_{mi})}.$$

The marginal density of $\boldsymbol{y}_m$ is given by

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

The density function $p(\boldsymbol{f}_m|\boldsymbol{x}_m,\boldsymbol{\theta})$ is derived from the GPFR model in (7.28) and will depend on which model we choose to use; refer to Chapter 5 for further details. We can therefore calculate the above marginal density by using, for example, the Laplace approximation similar to the ones we have used in

Section 7.1. The overall marginal log-likelihood for $\boldsymbol{\theta}$ is expressed by

$$l(\boldsymbol{\theta}|\mathcal{D}) = \sum_{m=1}^{M} \log p(\boldsymbol{y}_m|\boldsymbol{x}_m, \boldsymbol{\theta}). \tag{7.29}$$

The iterative method in Algorithm 7.1 can be modified to calculate the empirical Bayes estimates of $\boldsymbol{\theta}$ and $\boldsymbol{f}_m$.

For a new data point $\boldsymbol{x}_m^*$ in the m-th batch, the prediction can be calculated by using formulae in (7.16) and (7.17), as long as all the quantities are evaluated by the data and the estimates corresponding to the m-th batch.

The MCMC algorithm discussed in Section 7.1.4 can also be extended to the generalized GPFR models. Note the fact that

$$\begin{aligned} p(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M|\boldsymbol{\theta}, \mathcal{D}) &= p(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M|\boldsymbol{\theta}) p(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M|\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M) \\ &= \prod_{m=1}^{M} [p(\boldsymbol{f}_m|\boldsymbol{\theta}) p(\boldsymbol{y}_m|\boldsymbol{f}_m)], \end{aligned}$$

meaning that $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_M$ are conditional independent given $(\boldsymbol{\theta}, \mathcal{D})$. This leads to an efficient MCMC algorithm, since we can sample each $\boldsymbol{f}_m$ separately. A Gibbs sampler for the generalized GPFR model is given as follows.

Algorithm 7.3 (Gibbs sampler for the generalized GPFR models)**.** *One sweep of the algorithm includes the following steps:*

1. *Sample* $\boldsymbol{\theta}$ *from* $p(\boldsymbol{\theta}|\boldsymbol{f}, \mathcal{D}) \propto p(\boldsymbol{\theta}) \prod_{m=1}^{M} p(\boldsymbol{f}_m|\boldsymbol{\theta})$*;*
2. *For* $m = 1, \ldots, M$*, sample* $\boldsymbol{f}_m$ *from* $p(\boldsymbol{f}_m|\boldsymbol{\theta}, \mathcal{D}) = p(\boldsymbol{f}_m|\boldsymbol{\theta}) p(\boldsymbol{y}_m|\boldsymbol{f}_m)$.

In the first step, the dimension of $\boldsymbol{\theta}$ is usually not very large. Sampling $\boldsymbol{\theta}$ in this step is quite straightforward. In the second step, although we can sample $\boldsymbol{f}_m$ separately, the dimension of $\boldsymbol{f}_m$ is n_m, the sample size of the data collected in the m-th batch, which is usually quite large. Thus, we need to find an efficient algorithm in this step. Specifically, following the structure of binary data we described, we have

$$\begin{aligned} p(\boldsymbol{f}_m|\boldsymbol{\theta}, \mathcal{D}) &= p(\boldsymbol{f}_m|\boldsymbol{\theta}) p(\boldsymbol{y}_m|\boldsymbol{f}_m) \\ &= \varphi_n(\boldsymbol{f}_m; \boldsymbol{0}, \boldsymbol{K}_{n_m}(\boldsymbol{\theta})) \prod_{i=1}^{n_m} \frac{\exp(f_{mi} y_{mi})}{1 + \exp(f_{mi})}, \end{aligned}$$

where $\boldsymbol{K}_{n_m}(\boldsymbol{\theta})$ is the covariance matrix of $\boldsymbol{f}_m$ with (i, j)-th element $k(\boldsymbol{x}_{mi}, \boldsymbol{x}_{mj}; \boldsymbol{\theta})$. From (7.20), we know that the above density function is log-concave, and so a Gibbs sampler with adaptive rejection sampling can be used (Gilks and Wild, 1997).

Similar procedure can be applied to the GPFR model with other types of non-Gaussian data; further details can be found in Wang and Shi (2011).

7.4 Mixture models for multinomial batch data

A multinomial distribution is used in problems of multiclass classification. Multinomial data $y(t)$ takes values from $\{v = 0, 1, \ldots, V\}$, corresponding to $V+1$ classes. When $V = 1$, we are referring to the special case of binary data discussed in Section 7.1. Using V latent variables $f_v(t)$ for $v = 1, \ldots, V$, a multinomial logistic regression model is defined by

$$P(y(t) = v | f_v(t)) = \frac{\exp(f_v(t))}{1 + \sum_{a=1}^{V} \exp(f_a(t))}, \; v = 0, 1, \ldots, V, \tag{7.30}$$

where we define $f_0(t) = 0$ for notational convenience. A Gaussian process multinomial regression model, an extension of GP binary regression, aims to model $f_v(t)$ by a Gaussian process regression model

$$f_v(t) \sim GPR[0, k(\boldsymbol{\theta}_v) | \boldsymbol{x}(t)], \;\; v = 1, \ldots, V, \tag{7.31}$$

where *GPR* is defined by (1.11). For each class of $f_v(t)$, we use a GPR model with the same type of covariance function $k(\boldsymbol{\theta}_v)$ but with the different hyperparameters $\boldsymbol{\theta}_v$. The reader should note that, in principle, we could use different GPR models, each one of them with a different covariance function $k_v(\cdot)$, for different classes.

We now consider batch data with M batches. Denote the data collected in the m-th batch as $\mathcal{D}_m = \{(y_{mi}, \boldsymbol{x}_{mi}), i = 1, \ldots, n_m\}$, and the whole batch dataset as $\mathcal{D}$. For each pair $(y_{mi}, \boldsymbol{x}_{mi})$, the latent variable defined in (7.30) is denoted by $f_{mi,v}$. For $m = 1, \ldots, M$,

$$P(y_{mi} = v | f_{mi,v}) = \frac{\exp(f_{mi,v})}{1 + \sum_{a=1}^{V} \exp(f_{mi,a})}, \; v = 0, 1, \ldots, V, \; i = 1, \ldots, n_m, \tag{7.32}$$

where $f_{mi,0} = 0$. If we use a single GPR model given in (7.31), the latent variable $\boldsymbol{f}_{m,v} = (f_{m1,v}, \ldots, f_{mn_m,v})^T$ has an n_m-variate normal distribution

$$\boldsymbol{f}_{m,v} \sim N(0, \boldsymbol{K}_{n_m}(\boldsymbol{\theta}_v)),$$

where $\boldsymbol{K}_{n_m}(\boldsymbol{\theta}_v)$ is the covariance matrix of $\boldsymbol{f}_{m,v}$ calculated from the covariance kernel $k(\boldsymbol{\theta}_v)$ in (7.31).

If we recall the discussion in Chapter 6, (7.31) could also be replaced with a mixture GPR model to address the problem of heterogeneity. Then, $f_v(t)$ would be defined as

$$f_v(t) \sim \sum_{k=1}^{K} \pi_k GPR[0, k(\boldsymbol{\theta}_{k,v}) | \boldsymbol{x}(t)],$$

or equivalently, if we use an indicator variable $\boldsymbol{z} = (z_1, \ldots, z_M)^T$,

$$f_v(t) | z_m = k \sim GPR[0, k(\boldsymbol{\theta}_{k,v}) | \boldsymbol{x}(t)].$$

Note that the hyper-parameter $\boldsymbol{\theta}_{k,v}$ is different for each of the mixture components k. The indicator variable is modeled by

$$P(z_m = k) = \pi_k, \quad k = 1, \ldots, K. \tag{7.33}$$

We may assume a Dirichlet prior distribution $D(\delta, \ldots, \delta)$ for $(\pi_1, \ldots, \pi_K)$ as defined in (6.5). Consequently, a mixture GPR model for multinomial batch data is defined as follows:

$$\boldsymbol{f}_{m,v} | z_m = k \sim N(0, \boldsymbol{K}_{n_m}(\boldsymbol{\theta}_{k,v})), \tag{7.34}$$

where $\boldsymbol{K}_{n_m}(\boldsymbol{\theta}_{k,v})$ is the covariance matrix of $\boldsymbol{f}_{m,v}$ evaluated at $\boldsymbol{\theta}_{k,v}$, the hyper-parameters for class v, and mixture component k.

The hyper-parameters involved in the above model are

$$\boldsymbol{\Theta} = \{\boldsymbol{\theta}_{k,v},\ k = 1, \ldots, K,\ v = 1, \ldots, V\}.$$

There are two types of latent variables involved in this model. One is the predictor $\boldsymbol{f} = \{\boldsymbol{f}_{m,v},\ m = 1, \ldots, M,\ v = 1, \ldots, V\}$, and the other is the indicator variable $\boldsymbol{z} = (z_1, \ldots, z_M)$. The complicated structure of multicategory GPR models makes the empirical Bayes method intractable (with multimodality problems, for example); in these cases, a fully Bayesian approach with an MCMC algorithm is preferred. We use a Gibbs sampler, which allows us to sample separately $\boldsymbol{\Theta}$ as well as the latent variables $\boldsymbol{z}$ and $\boldsymbol{f}$. The algorithm is given as follows.

Algorithm 7.4 (Gibbs sampler for mixture GPR models with multinomial batch data)**.** *One sweep of the Gibbs sampler includes the following steps:*

1. *sample* $\boldsymbol{z}$ *from* $p(\boldsymbol{z}|\boldsymbol{f}, \boldsymbol{\Theta}, \mathcal{D})$*;*
2. *sample* $\boldsymbol{\Theta}$ *from* $p(\boldsymbol{\Theta}|\boldsymbol{f}, \boldsymbol{z}, \mathcal{D})$*;*
3. *sample* $\boldsymbol{f}$ *from* $p(\boldsymbol{f}|\boldsymbol{\Theta}, \boldsymbol{z}, \mathcal{D})$.

To sample $\boldsymbol{z}$ from $p(\boldsymbol{z}|\boldsymbol{f}, \boldsymbol{\Theta}, \mathcal{D})$, we let c_k be the number of observations for which $z_m = k$ over all $m = 1, \ldots, M$. We then use a sub-Gibbs sampler as discussed in the first step of Algorithm 6.1. Similar to (6.6), the density function used in the subalgorithm is

$$\begin{aligned} p(z_m = k|\boldsymbol{z}_{-m}, \boldsymbol{f}, \boldsymbol{\Theta}, \mathcal{D}) &\propto p(z_m = k|\boldsymbol{z}_{-m}) p(\boldsymbol{f}|\boldsymbol{\Theta}, \boldsymbol{z}) \\ &\propto p(z_m = k|\boldsymbol{z}_{-m}) p(\boldsymbol{f}_m|\boldsymbol{\Theta}, z_m = k) \\ &\propto p(z_m = k|\boldsymbol{z}_{-m}) \prod_{v=1}^{V} p(\boldsymbol{f}_{m,v}|\boldsymbol{\theta}_k). \end{aligned} \tag{7.35}$$

In the second step, we use the fact that if the prior distributions of $\boldsymbol{\theta}_k$ are independent for different k, the conditional density function of $\boldsymbol{\Theta}$ is

$$p(\boldsymbol{\Theta}|\boldsymbol{f}, \boldsymbol{z}, \mathcal{D}) = \prod_{k=1}^{K} p(\boldsymbol{\theta}_k|\boldsymbol{f}, \boldsymbol{z}, \mathcal{D})$$

with

$$p(\boldsymbol{\theta}_k|\boldsymbol{f},\boldsymbol{z},\mathcal{D}) \propto p(\boldsymbol{\theta}_k) \prod_{m\in\{z_m=k\}} \prod_{v=1}^{V} p(\boldsymbol{f}_{m,v}|\boldsymbol{\theta}_k);$$

therefore, the $\boldsymbol{\theta}_k$'s are conditionally independent and we can deal with each of them separately. For a particular k, the hybrid MC algorithm given in Appendix A.4 can be used.

In the third step, we have

$$p(\boldsymbol{f}|\boldsymbol{\Theta},\boldsymbol{z},\mathcal{D}) = \prod_{m=1}^{M} p(\boldsymbol{f}_m|\boldsymbol{\theta}_{z_m},\mathcal{D}) \propto \prod_{m=1}^{M} p(\boldsymbol{f}_m|\boldsymbol{\theta}_{z_m}) p(\boldsymbol{y}_m|\boldsymbol{f}_m). \tag{7.36}$$

The conditional distributions of $\boldsymbol{f}_m$ are therefore independent for $m = 1,\ldots,M$. Within the batch m,

$$\begin{aligned} p(\boldsymbol{f}_m|\boldsymbol{\Theta},\boldsymbol{z},\mathcal{D}) &\propto p(\boldsymbol{f}_m|\boldsymbol{\theta}_{z_m}) p(\boldsymbol{y}_m|\boldsymbol{f}_m) \\ &\propto \prod_{v=1}^{V} p(\boldsymbol{f}_{m,v}|\boldsymbol{\theta}_{z_m}) \prod_{i=1}^{n_m} \frac{\exp(f_{mi,y_{mi}})}{1+\sum_{a=1}^{V}\exp(f_{mi,a})}. \end{aligned}$$

Sampling $\boldsymbol{f}_m$ from the above density is similar to that of (7.20).

After the burn-in period, we take a set of samples $\{\boldsymbol{z}^{(t)},\boldsymbol{\Theta}^{(t)},\boldsymbol{f}^{(t)},t = 1,\ldots,T\}$. As it is usual within a Bayesian sampling-based approach, we carry out posterior inference using this set of samples. For example, let us now consider the prediction problem. At a new data point $\boldsymbol{x}^*_m$ in m-th batch, we can calculate a prediction as

$$\hat{f}^{*(t)}_{m,v} = \boldsymbol{K}^{*T}_m \boldsymbol{K}^{-1}_{n_m} \boldsymbol{f}^{(t)}_{m,v}. \tag{7.37}$$

Here the covariance function is calculated based on the data $\mathcal{D}_m$ and evaluated at $\boldsymbol{\theta}^{(t)}_{k,v}$ if $z^{(t)}_m$ takes the value of k. This is similar to the formula given in (7.21). The sample mean of $\{\hat{f}^{*(t)}_{m,v},t = 1,\ldots,T\}$ is used as an overall prediction of $f^*_{m,v}$. The probability $P(y^*_m = v|f^*_{m,v})$ is calculated using (7.32) for $v = 0,1,\ldots,V$. The new data point is allocated to class v^* if the probability takes the maximum value at $v = v^*$.

Example 7.4 (Three-way classification)**.** We consider a three-way classification problem in this example. A nonlinear model is used to generate the data. We consider a two-dimensional covariate $\boldsymbol{x}$, all the elements of which are generated independently from a uniform distribution over $(0,1)$. The class of the item y_i is selected as follows:

$$y_i = \begin{cases} 0 & \text{if } ||\boldsymbol{x}_i - \boldsymbol{x}_0)|| \le .35; \\ 1 & \text{otherwise, if } 0.8x_{i1} + 1.8x_{i2} \le c_0; \\ 2 & \text{otherwise.} \end{cases}$$

where $||\cdot||$ is the Euclidean distance. We construct a model with two mixture components with slightly different classification mechanisms—one corresponding to $\boldsymbol{x}_0 = (0.4, 0.5)$ and $c_0 = 0.6$, and the other corresponding to $\boldsymbol{x}_0 = (0.5, 0.4)$ and $c_0 = 1.0$. Figure 7.3 shows two groups of data, each corresponding to one mixture component of classifications; readers should notice the significant variation between the two groups. Ten batches of data are generated, five from each of the two mixture components. For each batch, 100 observations are produced, of which 40 are used as training data and 60 as test data.

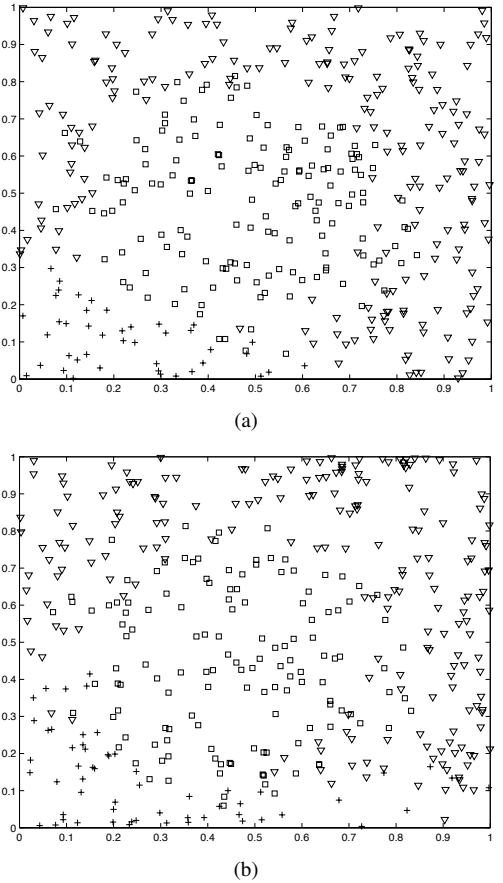


Figure 7.3 *Two groups of classification data. Each case is plotted according to its values for x_{i1} and x_{i2}. The three different symbols stand for three different classes. The data in panel (a) are generated from the first mixture component, while the data in panel (b) are generated from the second mixture component.*

We use the MCMC Algorithm 7.4 and collect 100 samples after the burn-in period to carry out posterior inference. The prediction of $f^*_{m,v}$ is calculated from the sample mean of the predictions in (7.37). They are used to classify the test data in each batch. Finally, the prediction of y^*_m is compared with its actual class value, which allows us to work out a classification error rate. For the 10 batches used in this example, the error rates range from 5% to 17% with an overall error rate 12%. These are very good results taking into account the complicated nature of the classification problem.

In classification, in addition to classification error rates, some other quantities can be used to measure the performance. Examples include the Rand index and the adjusted Rand index (Rand, 1971; Hubert and Arabie, 1985).

7.5 Further reading and notes

Inference for generalized Gaussian process regression models involves the calculation of a multi-dimensional integration. The direct calculation is often difficult and usually requires intensive computation. Two numerical schemes discussed in this chapter, namely, a Laplace approximation (empirical Bayes approach) and an MCMC algorithm (fully Bayesian approach), are a way of tackling this problem. Alternatively, other procedures have been developed in the literature. One approximation method that is popular in machine learning is the Expectation Propagation (EP; Minka, 2001); see, for example, Kuss and Rasmussen (2005) and Rasmussen and Williams (2006) for applications to the Gaussian process classification model. The other approximation approach is the variational Bayes (VB) method; see, for example, Hinton and van Camp (1993), MacKay (1995), Attias (1999, 2000), and Wang and Titterington (2006). For extensive reviews, the interested reader should refer to Titterington (2004) and Jordan (2004). There are some other recent developments, including the integrated nested Laplace approximations proposed by Rue et al. (2009).

Most current research about Gaussian process regression applied to non-Gaussian data concentrates on classification problems, including the generalized GPR model with binomial and multinomial response variables; see, for example, Williams (1998), Neal (1999), Shi et al. (2003), Kuss and Rasmussen (2005), Lawrence and Jordan (2005), and Nickisch and Rasmussen (2008). This chapter has discussed a general framework for generalized GPR models with distributions from the exponential family. It has also shown how such framework can be applied to the important case of functional batch data (repeated measurements). Indeed, this research direction is worth further development in both theory and practical applications.

In statistics, there are wide variations of the methods used to model response variables belonging to the exponential family. For instance, structured additive regression models are yet another alternative commonly used model;

references include Fahrmeir and Tutz (2001), Fahrmeir and Lang (2001), Cressie and Johannesson (2008), Banerjee et al. (2008), and Rue et al. (2009) among others.

Chapter 8

Some other related models

In this chapter, we present four important topics that we have not discussed in the previous chapters, and raise some open issues regarding them that are worth further consideration. Section 8.1 discusses a Gaussian process regression model with multivariate response variables. For a multivariate GPR model the difficulty is in defining a cross-variance function. We discuss a method for constructing a cross-variance function based on convolution (Boyle and Frean, 2005). In Section 8.2, we discuss a GP latent variable model that models the nonlinear relationships between observed functional manifest variables and unobserved functional latent variables by a GPR model. Section 8.3 discusses how to use a GPR model to find the optimal dynamic control in a nonlinear system. Section 8.4 tries to explain the relationship between the GPR model and RKHS (reproducing kernel Hilbert space). The current research on most of these topics up to the time of writing this book is limited and worthy of further development. In this regard, we briefly introduce some basic ideas and the theory of these topics in this chapter. We indeed hope readers, both those who are experienced as well as newcomers to the area, add their insights to the research.

8.1 Multivariate Gaussian process regression model

In this section, we discuss a functional regression problem with multivariate response variables. To simplify notations, we limit our discussion to the bivariate case and consider a two-dimensional response vector of dependent variables $(y_1(\boldsymbol{x}), y_2(\boldsymbol{x}))^T$ for $\boldsymbol{x} \in \mathcal{R}^Q$. Based on the discussion in the previous chapters, we can easily define two marginal Gaussian process regression models for $y_1(\boldsymbol{x})$ and $y_2(\boldsymbol{x})$ separately. Here, the difficulty is how to define cross-variance functions that not only capture interdependence between two response variables, but also ensure that the covariance matrix for the multivariate functional responses is positive definite (or at least non-negative definite). In this section, we introduce a method to construct a suitable cross-variance function by using a kernel convolution method (Higdon et al., 1999; Higdon, 2002; Boyle and Frean, 2005).

Let $\tau(\boldsymbol{x})$ be Gaussian white noise, in other words, $\tau(\boldsymbol{x})$ are independent and identically distributed random variables and each has $N(0,\sigma^2)$ for any $\boldsymbol{x} \in \mathcal{R}^Q$. Let $h(\boldsymbol{x})$ be a smoothing kernel; then a Gaussian process $\xi(\boldsymbol{x})$ can be constructed by convolving $\tau(\boldsymbol{x})$ with $h(\boldsymbol{x})$ as follows (Higdon, 2002):

$$\begin{aligned}\xi(\boldsymbol{x}) &= h(\boldsymbol{x}) \star \tau(\boldsymbol{x}) \\ &= \int h(\boldsymbol{x}-\boldsymbol{\alpha})\tau(\boldsymbol{\alpha})d\boldsymbol{\alpha} = \int h(\boldsymbol{\alpha})\tau(\boldsymbol{x}-\boldsymbol{\alpha})d\boldsymbol{\alpha}, \end{aligned} \tag{8.1}$$

where "$\star$" denotes convolution. Then we have the following result to define a new covariance function with a normal kernel.

Theorem 8.1. *Suppose that a smooth kernel $h(\boldsymbol{x})$ is given by*

$$h(\boldsymbol{x}) = v\exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{\mu})\}, \tag{8.2}$$

then the $\xi(\boldsymbol{x})$ defined in (8.1) *is a Gaussian process with the following covariance function:*

$$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \pi^{Q/2}v^2|\boldsymbol{A}|^{-1/2}\exp\{-\frac{1}{4}(\boldsymbol{x}_i-\boldsymbol{x}_j)^T\boldsymbol{A}(\boldsymbol{x}_i-\boldsymbol{x}_j)\}. \tag{8.3}$$

Proof. From definition (8.1), we know that $\xi(\boldsymbol{x})$ also has zero mean, thus

$$\begin{aligned} k(\boldsymbol{x}_i,\boldsymbol{x}_j) &= \mathrm{Cov}(\xi(\boldsymbol{x}_i),\xi(\boldsymbol{x}_j)) = \mathrm{E}(\xi(\boldsymbol{x}_i)\xi(\boldsymbol{x}_j)) \\ &= \mathrm{E}\left[\int h(\boldsymbol{\alpha})\tau(\boldsymbol{x}_i-\boldsymbol{\alpha})d\boldsymbol{\alpha}\int h(\boldsymbol{\beta})\tau(\boldsymbol{x}_j-\boldsymbol{\beta})d\boldsymbol{\beta}\right] \\ &= \int\int h(\boldsymbol{\alpha})h(\boldsymbol{\beta})\mathrm{E}\left[\tau(\boldsymbol{x}_i-\boldsymbol{\alpha})\tau(\boldsymbol{x}_j-\boldsymbol{\beta})\right]d\boldsymbol{\alpha}d\boldsymbol{\beta}. \end{aligned} \tag{8.4}$$

Since $\tau(\cdot)$ is Gaussian white noise and independent at different points, thus

$$\mathrm{E}\left[\tau(\boldsymbol{x}_i-\boldsymbol{\alpha})\tau(\boldsymbol{x}_j-\boldsymbol{\beta})\right] = \sigma^2\delta((\boldsymbol{x}_j-\boldsymbol{\beta})-(\boldsymbol{x}_i-\boldsymbol{\alpha})),$$

where $\delta(\cdot)$ is the Kronecker delta. Without loss of generality, we assume that $\sigma^2 = 1$. Replacing the related quantity in (8.4) by the above equation, we have

$$\begin{aligned} k(\boldsymbol{x}_i,\boldsymbol{x}_j) &= \int\int h(\boldsymbol{\alpha})h(\boldsymbol{\beta})\delta(\boldsymbol{\alpha}-[\boldsymbol{x}_i-\boldsymbol{x}_j+\boldsymbol{\beta}])d\boldsymbol{\alpha}d\boldsymbol{\beta} \\ &= \int h(\boldsymbol{x}_i-\boldsymbol{x}_j+\boldsymbol{\beta})h(\boldsymbol{\beta})d\boldsymbol{\beta}. \end{aligned}$$

When $h(\cdot)$ takes the form (8.2), it is easy to prove (8.3) from the above equation using the property of normal density function. □

If we take $\boldsymbol{A} = \mathrm{diag}(w_1,\ldots,w_Q)$, a diagonal matrix, then the above defined

$\xi(\boldsymbol{x})$ is a Gaussian process regression model with a squared exponential covariance function (4.6).

The convolution defined in (8.1) can then be used to model a multivariate Gaussian process regression model (Boyle and Frean, 2005). We first define three independent Gaussian white noise processes, namely, $\tau_0(\boldsymbol{x})$, $\tau_1(\boldsymbol{x})$, and $\tau_2(\boldsymbol{x})$. By convolving these Gaussian white noise processes with different smoothing kernel functions, we construct the following four Gaussian processes:

$$\begin{aligned} \xi_a(\boldsymbol{x}) &= h_{a0}(\boldsymbol{x}) \star \tau_0(\boldsymbol{x}); \\ \eta_a(\boldsymbol{x}) &= h_{a1}(\boldsymbol{x}) \star \tau_a(\boldsymbol{x}), \text{ for } a = 1,2. \end{aligned} \tag{8.5}$$

Here, h_{a0} and h_{a1} are suitable smooth kernel functions for $a = 1,2$. It is obvious that $\xi_1(\boldsymbol{x})$ and $\xi_2(\boldsymbol{x})$ are dependent since both are constructed from the same Gaussian white noise $\tau_0(\boldsymbol{x})$, but they are independent to $\eta_1(\boldsymbol{x})$ and $\eta_2(\boldsymbol{x})$. A bivariate Gaussian process regression model can therefore be defined by using these Gaussian processes,

$$y_a(\boldsymbol{x}) = \xi_a(\boldsymbol{x}) + \eta_a(\boldsymbol{x}) + \varepsilon_a(\boldsymbol{x}), \text{ for } a = 1,2, \tag{8.6}$$

where $\varepsilon_a(\boldsymbol{x})$ are error items that are independent and identically distributed with normal distribution $N(0,\sigma_a^2)$. The dependence between the two response variables $y_1(\boldsymbol{x})$ and $y_2(\boldsymbol{x})$ is defined through $\xi_1(\boldsymbol{x})$ and $\xi_2(\boldsymbol{x})$.

Theorem 8.2. *If the smoothing kernel functions in* (8.5) *take the following forms*

$$h_{10}(\boldsymbol{x}) = v_{10} \exp\{-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{A}_{10} \boldsymbol{x}\}, \tag{8.7}$$

$$h_{20}(\boldsymbol{x}) = v_{20} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{A}_{20} (\boldsymbol{x}-\boldsymbol{\mu})\}, \tag{8.8}$$

$$h_{a1}(\boldsymbol{x}) = v_{a1} \exp\{-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{A}_{a1} \boldsymbol{x}\}, \ a = 1,2, \tag{8.9}$$

then $(y_1(\boldsymbol{x}), y_2(\boldsymbol{x}))^T$ *given in* (8.6) *defines a dependent bivariate Gaussian process regression model with zero means and the following covariance function:*

$$\begin{aligned} \boldsymbol{k}(\boldsymbol{x}_i,\boldsymbol{x}_j) &= \begin{pmatrix} Cov(y_1(\boldsymbol{x}_i), y_1(\boldsymbol{x}_j)) & Cov(y_1(\boldsymbol{x}_i), y_2(\boldsymbol{x}_j)) \\ Cov(y_2(\boldsymbol{x}_i), y_1(\boldsymbol{x}_j)) & Cov(y_2(\boldsymbol{x}_i), y_2(\boldsymbol{x}_j)) \end{pmatrix} \\ &\stackrel{d}{=} \begin{pmatrix} k_{11}(\boldsymbol{x}_i,\boldsymbol{x}_j) & k_{12}(\boldsymbol{x}_i,\boldsymbol{x}_j) \\ k_{21}(\boldsymbol{x}_i,\boldsymbol{x}_j) & k_{22}(\boldsymbol{x}_i,\boldsymbol{x}_j) \end{pmatrix}, \end{aligned} \tag{8.10}$$

where, for $a,b = 1,2$, *and* $a \neq b$,

$$k_{aa}(\boldsymbol{x}_i,\boldsymbol{x}_j) = k_{aa}^{\xi}(\boldsymbol{d}) + k_{aa}^{\eta}(\boldsymbol{d}) + \delta_{ij}\sigma_a^2, \tag{8.11}$$

$$k_{ab}(\boldsymbol{x}_i,\boldsymbol{x}_j) = k_{ab}^{\xi}(\boldsymbol{d}), \tag{8.12}$$

with $\boldsymbol{d} = \boldsymbol{x}_i - \boldsymbol{x}_j$ *and*

$$\begin{aligned}
k^{\xi}_{aa}(\boldsymbol{d}) &= \pi^{Q/2} v_{a0}^2 |\boldsymbol{A}_{a0}|^{-1/2} \exp\{-\frac{1}{4}\boldsymbol{d}^T \boldsymbol{A}_{a0} \boldsymbol{d}\}, \\
k^{\xi}_{12}(\boldsymbol{d}) &= (2\pi)^{Q/2} v_{10} v_{20} |\boldsymbol{A}_{10} + \boldsymbol{A}_{20}|^{-1/2} \exp\{-\frac{1}{4}(\boldsymbol{d} - \boldsymbol{\mu})^T \boldsymbol{\Sigma} (\boldsymbol{d} - \boldsymbol{\mu})\}, \\
k^{\xi}_{21}(\boldsymbol{d}) &= k^{\xi}_{12}(-\boldsymbol{d}) \\
k^{\eta}_{aa}(\boldsymbol{d}) &= \pi^{Q/2} v_{a1}^2 |\boldsymbol{A}_{a1}|^{-1/2} \exp\{-\frac{1}{4}\boldsymbol{d}^T \boldsymbol{A}_{a1} \boldsymbol{d}\},
\end{aligned}$$

where $\boldsymbol{\Sigma} = \boldsymbol{A}_{10}(\boldsymbol{A}_{10} + \boldsymbol{A}_{20})^{-1} \boldsymbol{A}_{20} = \boldsymbol{A}_{20}(\boldsymbol{A}_{10} + \boldsymbol{A}_{20})^{-1} \boldsymbol{A}_{10}$.

Proof. For $a = 1, 2$, k^{ξ}_{aa} and k^{η}_{aa} can be obtained from (8.3) in Theorem 8.1. For cross-covariance, we have

$$\begin{aligned}
k^{\xi}_{12}(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \mathrm{Cov}(\xi_1(\boldsymbol{x}_i), \xi_2(\boldsymbol{x}_j)) = \mathrm{E}(\xi_1(\boldsymbol{x}_i)\xi_2(\boldsymbol{x}_j)) \\
&= \mathrm{E}\left[\int h_{10}(\boldsymbol{\alpha}) \tau_0(\boldsymbol{x}_i - \boldsymbol{\alpha}) d\boldsymbol{\alpha} \int h_{20}(\boldsymbol{\beta}) \tau_0(\boldsymbol{x}_j - \boldsymbol{\beta}) d\boldsymbol{\beta}\right] \\
&= \int\int h_{10}(\boldsymbol{\alpha}) h_{20}(\boldsymbol{\beta}) \mathrm{E}[\tau_0(\boldsymbol{x}_i - \boldsymbol{\alpha}) \tau_0(\boldsymbol{x}_j - \boldsymbol{\beta})] d\boldsymbol{\alpha} d\boldsymbol{\beta} \\
&= \int\int h_{10}(\boldsymbol{\alpha}) h_{20}(\boldsymbol{\beta}) \delta(\boldsymbol{\alpha} - [\boldsymbol{x}_i - \boldsymbol{x}_j + \boldsymbol{\beta}]) d\boldsymbol{\alpha} d\boldsymbol{\beta} \\
&= \int h_{10}(\boldsymbol{x}_i - \boldsymbol{x}_j + \boldsymbol{\beta}) h_{20}(\boldsymbol{\beta}) d\boldsymbol{\beta}.
\end{aligned}$$

Replacing h_{a0} by equations (8.7) and (8.8) and using the property of normal density functions, we can obtain the formula for $k^{\xi}_{12}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ given in the theorem. □

Now suppose that we have observed the following data:

$$\mathcal{D} = \left\{ \begin{pmatrix} y_{1i} \\ \boldsymbol{x}_{1i} \end{pmatrix}, i = 1, \ldots, n_1, \begin{pmatrix} y_{2i} \\ \boldsymbol{x}_{2i} \end{pmatrix}, i = 1, \ldots, n_2 \right\}.$$

Let $\boldsymbol{y} = (y_{11}, \ldots, y_{1n_1}, y_{21}, \ldots, y_{2n_2})^T$; we have

$$\boldsymbol{y} \sim N(0, \boldsymbol{\Psi}) \text{ and } \boldsymbol{\Psi} = \begin{pmatrix} \boldsymbol{\Psi}_{11} & \boldsymbol{\Psi}_{12} \\ \boldsymbol{\Psi}_{21} & \boldsymbol{\Psi}_{22} \end{pmatrix}, \tag{8.13}$$

where, for $a, b = 1, 2$,

$$\boldsymbol{\Psi}_{ab} = \begin{pmatrix} k_{ab}(\boldsymbol{x}_{a1}, \boldsymbol{x}_{b1}) & \cdots & k_{ab}(\boldsymbol{x}_{a1}, \boldsymbol{x}_{bn_b}) \\ \vdots & \vdots & \vdots \\ k_{ab}(\boldsymbol{x}_{an_a}, \boldsymbol{x}_{b1}) & \cdots & k_{ab}(\boldsymbol{x}_{an_a}, \boldsymbol{x}_{bn_b}) \end{pmatrix}.$$

Comparing (8.13) with (2.5), we notice that they have similar forms except for the larger dimension of the covariance matrix $\boldsymbol{\Psi}$ defined in (8.13). We can therefore apply the inference methods discussed in the previous chapters to the above bivariate GPR model. For example, we can use an empirical Bayesian approach to estimate the hyper-parameters involved in (8.13). The marginal likelihood can still be expressed by (3.7) although $\boldsymbol{y}$ and $\boldsymbol{\Psi}$ should be replaced by those in (8.13) and the empirical Bayes estimates can be calculated by maximizing this marginal likelihood. Also, based on the discussion in Chapters 2 and 3, we know that the predictive distribution of $\boldsymbol{y}^* = (y_1^*, y_2^*)^T$ at a new point $\boldsymbol{x}^*$ is a bivariate normal distribution with the following mean and covariance matrix:

$$\hat{\boldsymbol{y}}^* = \boldsymbol{\Psi}_{\boldsymbol{y}}^{*T} \boldsymbol{\Psi}^{-1} \boldsymbol{y}, \tag{8.14}$$

$$\hat{\boldsymbol{\Psi}}_{y*} = \boldsymbol{\Psi}^* - \boldsymbol{\Psi}_{\boldsymbol{y}}^{*T} \boldsymbol{\Psi}^{-1} \boldsymbol{\Psi}_{\boldsymbol{y}}^*, \tag{8.15}$$

where $\boldsymbol{\Psi}_{\boldsymbol{y}}^{*T}$ is a $2 \times (n_1 + n_2)$ covariance matrix between $\boldsymbol{y}^*$ and $\boldsymbol{y}$. It is expressed by

$$\begin{pmatrix} k_{11}(\boldsymbol{x}^*, \boldsymbol{x}_{11}) & \cdots & k_{11}(\boldsymbol{x}^*, \boldsymbol{x}_{1n_1}) & k_{12}(\boldsymbol{x}^*, \boldsymbol{x}_{21}) & \cdots & k_{12}(\boldsymbol{x}^*, \boldsymbol{x}_{2n_2}) \\ k_{21}(\boldsymbol{x}_{11}, \boldsymbol{x}^*) & \cdots & k_{21}(\boldsymbol{x}_{1n_1}, \boldsymbol{x}^*) & k_{22}(\boldsymbol{x}^*, \boldsymbol{x}_{21}) & \cdots & k_{22}(\boldsymbol{x}^*, \boldsymbol{x}_{2n_2}) \end{pmatrix}.$$

The $\boldsymbol{\Psi}^*$ is a 2×2 covariance matrix of $\boldsymbol{y}^*$, which is given by

$$\boldsymbol{\Psi}^* = \begin{pmatrix} k_{11}(\boldsymbol{x}^*, \boldsymbol{x}^*) & k_{12}(\boldsymbol{x}^*, \boldsymbol{x}^*) \\ k_{21}(\boldsymbol{x}^*, \boldsymbol{x}^*) & k_{22}(\boldsymbol{x}^*, \boldsymbol{x}^*) \end{pmatrix}.$$

The model defined in (8.6) can be easily extended to build models with multivariate response variables. A simple way is to use the following model:

$$y_a(\boldsymbol{x}) = \xi_a(\boldsymbol{x}) + \eta_a(\boldsymbol{x}) + \varepsilon_a(\boldsymbol{x}), \quad \text{for } a = 1, 2, \ldots, p, \tag{8.16}$$

where $\varepsilon_a(\boldsymbol{x})$ are independent error items and $\eta_a(\boldsymbol{x})$ is constructed by (8.5). To increase flexibility on modeling correlation amongst different response variables, we may define $\xi_a(\boldsymbol{x})$ using more than one Gaussian white noise process, say, $\tau_{10}(\boldsymbol{x}), \ldots, \tau_{T0}(\boldsymbol{x})$. We can then define

$$\xi_a(\boldsymbol{x}) = \sum_{t=1}^{T} h_{t,a0} \star \tau_{t0}(\boldsymbol{x}).$$

The inference based on the above model is quite similar to the one we discussed above for model (8.6); further details can be found in Boyle and Frean (2004).

Kaufman and Sain (2010) also discussed a multivariate Gaussian process model. They defined multivariate Gaussian processes with a special structure in a GPFR ANOVA model although their original purpose was to use them

to cope with the constraints of treatment effects. Instead of defining independent GPR models in (5.51) for each treatment effect, they adopted a specific multivariate GP model using the following covariance function:

$$\begin{aligned} k_{ab}(\boldsymbol{x}_i,\boldsymbol{x}_j) &= \mathrm{Cov}(\alpha_a(\boldsymbol{x}_i),\alpha_b(\boldsymbol{x}_j)) \\ &= \begin{cases} (1-\frac{1}{J})k(\boldsymbol{x}_i,\boldsymbol{x}_j), & \text{if } a=b; \\ -\frac{1}{J}k(\boldsymbol{x}_i,\boldsymbol{x}_j), & \text{if } a\neq b, \end{cases} \end{aligned}$$

where J is the total number of factor levels, $k(\cdot,\cdot)$ is a covariance function, and $\alpha_a(\boldsymbol{x}_i)$ is the treatment effect for level a. A general model may be defined using a projection matrix (Kaufman and Sain, 2010), and further investigation in this direction is expected for future research.

8.2 Gaussian process latent variable models

Latent variable models are used to model relationships between observed functional manifest variables and unobserved latent variables. Models commonly used to explain the latent structure include the factor analysis model, the item response theory, the latent class analysis model, and the structural equation model (see, e.g., Bartholomew and Knott, 1999; Lee, 2007). Latent variable models have wide applications in almost all areas of sciences and social sciences. In this section, we consider a latent variable model based on a Gaussian process prior, a so-called Gaussian process latent variable model. This model enables us to model nonlinear relationships between manifest and latent variables nonparametrically and has a great deal of flexibility. In this section, we focus on a specific method proposed by Lawrence (2005).

Let us assume that we are given a set of D-dimensional functional manifest variables $\boldsymbol{Y}(t) = (y_1(t),\ldots,y_D(t))^T$. We denote the Q-dimensional functional latent variables associated with these manifest variables at each data point by $\boldsymbol{Z}(t) = (z_1(t),\ldots,z_Q(t))^T$. These latent variables cannot be observed directly. Some familiar examples include factor scores in a linear factor analysis model and principal components in PCA (principal component analysis) or kernel PCA (Schölkopf and Smola, 2002). Note that the values of those latent variables can only be estimated in a model by using the information from the manifest variables. As a nonparametric model for latent variables, a GP latent variable (GPLV) model is defined by

$$y_d(t) = f_d(\boldsymbol{Z}(t)) + \varepsilon_d(t), \;\; d = 1,\ldots,D, \tag{8.17}$$

where $\varepsilon_d(t)$'s are independent errors and each has a normal distribution $N(0,\sigma_d^2)$. Each $f_d(\cdot)$ is modeled by a GPR model,

$$f_d(\boldsymbol{Z}(t)) \sim GPR(0, k(\boldsymbol{\theta}_d)|\boldsymbol{Z}(t)), \tag{8.18}$$

where the GPR model is defined in (1.11) and $k(\boldsymbol{\theta}_d)$ is a covariance function that depends on the hyper-parameters $\boldsymbol{\theta}_d$. When $\boldsymbol{Z}(t)$ is given, we assume that $f_d(\boldsymbol{Z}(t))$'s are conditionally independent for different d. In model (8.18), we are assuming the same covariance structure for different function variables $y_d(t)$ but with different hyper-parameter $\boldsymbol{\theta}_d$. However, the method discussed below can be applied to models with different covariance kernel structures without much difficulty.

We can assume that latent variables have independent Gaussian priors:

$$\boldsymbol{Z}(t) \sim N(0, \boldsymbol{I}_Q), \tag{8.19}$$

where $\boldsymbol{I}_Q$ is a Q-dimensional identity matrix. This assumption is analogous to the one for factor scores in exploratory factor analysis (see, e.g., Bartholomew and Knott, 1999). Some other models can also be used here, for example, the independent structure among latent variables in (8.19) can be replaced by a general dependent structure.

We now assume that we have observed $\boldsymbol{Y}(t)$ at $t = t_1, \ldots, t_n$ and obtained the following dataset:

$$\mathcal{D} = \{\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n\} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_D\},$$

where $\boldsymbol{Y}_i = (y_{i1}, \ldots, y_{iD})^T$ for $i = 1, \ldots, n$ and $\boldsymbol{y}_d = (y_{1d}, \ldots, y_{nd})^T$ for $d = 1, \ldots, D$. Let z_{ij} be the j-th latent variable associated with data point t_i for $j = 1, \ldots, Q$, and $\boldsymbol{\Theta}$ be all the hyper-parameters involved in the models (8.17) and (8.18) with $p(\boldsymbol{\Theta})$ as their hyper-prior density function.

For the d-th functional variable $\boldsymbol{y}_d$, the parameters include σ_d^2 in (8.17) and $\boldsymbol{\theta}_d$ in (8.18). To simplify notations, let $\boldsymbol{\theta}_d$ comprise both of them. The posterior density function for $(\boldsymbol{Z}, \boldsymbol{\Theta})$ can therefore be expressed by

$$\begin{aligned} p(\boldsymbol{Z}, \boldsymbol{\Theta} | \mathcal{D}) &\propto p(\boldsymbol{Y} | \boldsymbol{Z}, \boldsymbol{\Theta}) p(\boldsymbol{Z}) p(\boldsymbol{\Theta}) \\ &\propto p(\boldsymbol{Z}) \prod_{d=1}^{D} [p(\boldsymbol{y}_d | \boldsymbol{Z}, \boldsymbol{\theta}_d) p(\boldsymbol{\theta}_d)], \end{aligned} \tag{8.20}$$

where $p(\boldsymbol{y}_d | \boldsymbol{Z}, \boldsymbol{\theta}_d)$ is the density function of the following normal distribution:

$$\boldsymbol{y}_d | \boldsymbol{Z} \sim N(0, \boldsymbol{\Psi}_d(\boldsymbol{\theta}_d)), \tag{8.21}$$

where the (i, j)-th element of $\boldsymbol{\Psi}_d$ is given by

$$\Psi_{d,ij} = \mathrm{Cov}(y_{id}, y_{jd} | \boldsymbol{Z}, \boldsymbol{\theta}_d) = k(\boldsymbol{z}_i, \boldsymbol{z}_j | \boldsymbol{\theta}_d) + \sigma_d^2 \delta_{ij}. \tag{8.22}$$

In the GPLV model, we are interested in estimating the values of $(\boldsymbol{Z}, \boldsymbol{\Theta})$. They can be estimated using the posterior distribution given in (8.20). For example,

we can calculate MAP estimates of $(\boldsymbol{Z}, \boldsymbol{\Theta})$ by maximizing the posterior density function, or the following log density function:

$$\sum_{i=1}^{n}\left(-\frac{Q}{2}\log(2\pi)-\frac{1}{2}\boldsymbol{z}_i^T\boldsymbol{z}_i\right)+\sum_{d=1}^{D}p(\boldsymbol{\theta}_d)$$
$$+\sum_{d=1}^{D}\left(-\frac{n}{2}\log(2\pi)-\frac{1}{2}\log|\boldsymbol{\Psi}_d|-\frac{1}{2}\boldsymbol{y}_d^T\boldsymbol{\Psi}_d^{-1}\boldsymbol{y}_d\right). \qquad (8.23)$$

However, calculating the MAP from the above equation may cause numerical problems because the dimension of $\boldsymbol{Z}$, $Q \times n$, is usually very large since n is the sample size. Similar numerical problems exist when we use other Bayesian inference methods based on the posterior distribution (8.20) directly.

Alternatively, we could use an MCMC algorithm. Specifically, we may design a Gibbs sampler to update $\boldsymbol{Z}$ and $\boldsymbol{\Theta}$ in turn, which is a commonly used approach in latent variable models. For GPLV models, we can use the following algorithm.

Algorithm 8.1 (Gibbs sampler for the GPLV model)**.** *One sweep of the Gibbs sampler includes the following two steps:*

1. *sample* $\boldsymbol{\Theta}$ *from* $p(\boldsymbol{\Theta}|\mathcal{D},\boldsymbol{Z})$*;*
2. *sample* $\boldsymbol{Z}$ *from* $p(\boldsymbol{Z}|\mathcal{D},\boldsymbol{\Theta})$.

As we have mentioned before, the $\boldsymbol{y}_d$'s are conditionally independent when $\boldsymbol{Z}$ is given. Thus, from (8.20), we know that the conditional posterior distributions of $\boldsymbol{\theta}_d$ are also independent for different d when $\boldsymbol{Z}$ is given. This conditional independence feature results in an efficient implementation in step 1. That is, we can sample $\boldsymbol{\theta}_d$ separately for $d = 1,\ldots,D$ from the following conditional density:

$$p(\boldsymbol{\theta}_d|\mathcal{D},\boldsymbol{Z}) \propto p(\boldsymbol{y}_d|\boldsymbol{Z},\boldsymbol{\theta}_d)p(\boldsymbol{\theta}_d).$$

This density function is similar to the one in (3.10). We can therefore apply the hybrid MC algorithm discussed in Chapter 3 to sample $\boldsymbol{\theta}_d$ from the above distribution.

In step 2, since the dimension of $\boldsymbol{Z}$ is usually very large, it is rather difficult to sample the whole set of $\boldsymbol{Z}$ from $p(\boldsymbol{Z}|\mathcal{D},\boldsymbol{\Theta})$ directly. Hence, instead of sampling from the full condition of $p(\boldsymbol{Z}|\mathcal{D},\boldsymbol{\Theta})$, we need to resort to a different method. One way is to use the following sub-Gibbs sampler. For $i = 1,\ldots,n$, we consider the full conditional distribution of $\boldsymbol{z}_i$ given $\boldsymbol{\Theta}$, $\boldsymbol{Z}_{-i}$ and the observed data, where $\boldsymbol{Z}_{-i}$ stands for $\{\boldsymbol{z}_j, j \neq i\}$, i.e., all latent variables except for $\boldsymbol{z}_i$. The algorithm is to

$$\text{generate } \boldsymbol{z}_i \text{ from } p(\boldsymbol{z}_i|\mathcal{D},\boldsymbol{\Theta},\boldsymbol{Z}_{-i})$$

in turn. Using the notation $\boldsymbol{Y}_{-i}$ similarly to $\boldsymbol{Z}_{-i}$, this condition density can be

expressed by

$$\begin{aligned} p(\boldsymbol{z}_i|\mathcal{D},\boldsymbol{\Theta},\boldsymbol{Z}_{-i}) &\propto p(\boldsymbol{z}_i)p(\boldsymbol{Y}|\boldsymbol{\Theta},\boldsymbol{Z}_{-i},\boldsymbol{z}_i) \\ &\propto p(\boldsymbol{z}_i)p(\boldsymbol{Y}_i|\boldsymbol{Y}_{-i},\boldsymbol{\Theta},\boldsymbol{Z}) \\ &= p(\boldsymbol{z}_i)\prod_{d=1}^{D} p(\boldsymbol{y}_{id}|\boldsymbol{y}_{-i,d},\boldsymbol{\Theta},\boldsymbol{Z}) \\ &= p(\boldsymbol{z}_i)\prod_{d=1}^{D} \varphi(\boldsymbol{y}_{id}|\mu_{id},\sigma_{id}^2), \end{aligned} \tag{8.24}$$

where φ is the density function of a normal distribution. The last equation is obtained from the fact that the conditional distribution of $\boldsymbol{y}_{id}$ is still a normal distribution when $\boldsymbol{y}_{-i,d}$ and other random quantities are given. The mean and variance are given, respectively, by

$$\begin{aligned} \mu_{id} &= \boldsymbol{\Psi}_{-i,i,d}^T\boldsymbol{\Psi}_{-i,d}^{-1}\boldsymbol{y}_{-i,d}, \\ \sigma_{id}^2 &= \Psi_d(\boldsymbol{z}_i,\boldsymbol{z}_i) - \boldsymbol{\Psi}_{-i,i,d}^T\boldsymbol{\Psi}_{-i,d}^{-1}\boldsymbol{\Psi}_{-i,i,d}, \end{aligned}$$

where $\boldsymbol{\Psi}_{-i,i,d}^T = (\Psi_d(\boldsymbol{z}_1,\boldsymbol{z}_i),\ldots,\Psi_d(\boldsymbol{z}_{i-1},\boldsymbol{z}_i),\Psi_d(\boldsymbol{z}_{i+1},\boldsymbol{z}_i),\ldots,\Psi_d(\boldsymbol{z}_n,\boldsymbol{z}_i))$ is the covariance between y_{id} and $\boldsymbol{y}_{-i,d}$, and $\boldsymbol{\Psi}_{-i,d}$ is the covariance matrix of $\boldsymbol{y}_{-i,d}$. The dimension of $\boldsymbol{z}_i$ is Q, which is usually small, and all the density functions involved in (8.24) are Gaussian densities; the numerical calculation can then be done easily and quickly, and thus the implementation of the sub-Gibbs sampler is usually quite efficient.

After the burn-in period, we can collect a set of samples from the algorithm, and then use a Bayesian sampling-based approach for statistical inference. For example, we can simply estimate the values of $\boldsymbol{\Theta}$ and $\boldsymbol{Z}$ by the sample means.

Lawrence (2005) suggested an alternative practical algorithm based on an active set and sparse greedy approximation as discussed in Section 3.3.2. We describe his idea here briefly. In this method, we first use Algorithm 3.1 to select an active set, denoted by $\mathcal{A}$. The basic idea of this method is to estimate the hyper-parameters and latent variables based on the data associated with the active set. In other words, instead of using the log posterior density based on the whole dataset given in (8.23), we can used the log density based on the active set which is given as follows:

$$\begin{aligned} &\sum_{i\in\mathcal{A}}^{n}\left(-\frac{Q}{2}\log(2\pi)-\frac{1}{2}\boldsymbol{z}_i^T\boldsymbol{z}_i\right)+\sum_{d=1}^{D}p(\boldsymbol{\theta}_d) \\ &+\sum_{d=1}^{D}\left(-\frac{n_a}{2}\log(2\pi)-\frac{1}{2}\log|\boldsymbol{\Psi}_{\mathcal{A},d}|-\frac{1}{2}\boldsymbol{y}_{\mathcal{A},d}^T\boldsymbol{\Psi}_{\mathcal{A},d}^{-1}\boldsymbol{y}_{A,d}\right), \end{aligned}$$

where n_a is the number of observations in active set $\mathcal{A}$, and $\boldsymbol{y}_{\mathcal{A},d} = \{y_{id}, i\in\mathcal{A}\}$

are the observations associated with the active set for the d-th variable. The $\mathbf{\Psi}_{\mathcal{A},d}$ is the $n_a \times n_a$ covariance matrix of $\boldsymbol{y}_{\mathcal{A},d}$. Its element is calculated by (8.22) and depends on $\mathbf{Z}_{\mathcal{A}} = \{\boldsymbol{z}_i, i \in \mathcal{A}\}$. Because $Q \times n_a$, the dimension of the latent variable $\mathbf{Z}_{\mathcal{A}}$ involved in the above posterior density function, is much smaller than $Q \times n$, the dimension of the whole set of latent variables, the calculation of the MAP for $\mathbf{\Theta}$ and $\mathbf{Z}_{\mathcal{A}}$ based on the active set is much faster than the calculation based on the whole dataset.

However, Algorithm 3.1 requires the values of $\mathbf{Z}$ for the whole set to calculate d_j—the criterion used to select the elements of the active sets. Lawrence (2005) suggested using a suboptimization method to calculate the values of $\mathbf{Z}$ in the inactive set. Let j be a data point in the inactive set—the set including all the data points other than those included in the active set. Using the fact that both $\boldsymbol{y}_{\mathcal{A},d}$ and y_{jd} have normal distribution as given in (8.21), we have the following results:

$$y_{jd}|\boldsymbol{y}_{\mathcal{A},d}, \mathbf{Z}_{\mathcal{A}}, \boldsymbol{z}_j, \boldsymbol{\theta}_d \sim N(\mu_{jd}, \sigma^2_{jd}), \tag{8.25}$$

where

$$\begin{aligned} \mu_{jd} &= \mathbf{\Psi}^T_{\mathcal{A},j,d} \mathbf{\Psi}^{-1}_{\mathcal{A},d} \boldsymbol{y}_{\mathcal{A},d}, \\ \sigma^2_{jd} &= \Psi_d(\boldsymbol{z}_j, \boldsymbol{z}_j) - \mathbf{\Psi}^T_{\mathcal{A},j,d} \mathbf{\Psi}^{-1}_{\mathcal{A},d} \mathbf{\Psi}_{\mathcal{A},j,d}, \end{aligned}$$

where $\mathbf{\Psi}_{\mathcal{A},j,d} = \{\Psi_d(\boldsymbol{z}_i, \boldsymbol{z}_j), i \in \mathcal{A}\}$ is the covariance between y_{jd} and $\boldsymbol{y}_{\mathcal{A},d}$, and $\mathbf{\Psi}_{\mathcal{A},d}$ is the covariance matrix of $\boldsymbol{y}_{\mathcal{A},d}$. They can be calculated by (8.22). The estimate of $\boldsymbol{z}_j$ can be calculated by maximizing the following log density function in terms of $\boldsymbol{z}_j$:

$$\sum_{d=1}^{D} \log[\varphi(y_{jd}|\mu_{jd}, \sigma^2_{jd})].$$

The above log density function involves the parameters $\boldsymbol{\theta}_d$ and latent variables $\{\boldsymbol{z}_i, i \in \mathcal{A}\}$, which are unknown. We then need an iterative method. In each iteration, we can evaluate all the quantities by the estimates obtained from the current active set. The optimization of the above log density function in terms of a Q-dimensional latent variable $\boldsymbol{z}_j$ can be achieved very efficiently, since Q is usually quite small. The iterative algorithm is specified as follows.

Algorithm 8.2 (Active set and sparse greedy approximation for the GPLV model)**.** *The iterative algorithm includes the following steps:*

1. *Initialize* $\mathbf{Z}$*;*
2. *Use Algorithm 3.1 to select an active set* $\mathcal{A}$ *and estimate* $\mathbf{\Theta}$ *and* $\mathbf{Z}_{\mathcal{A}}$*;*
3. *Estimate* $\mathbf{Z}$ *for the inactive set;*
4. *Repeat steps 2 and 3 until the algorithm converges.*

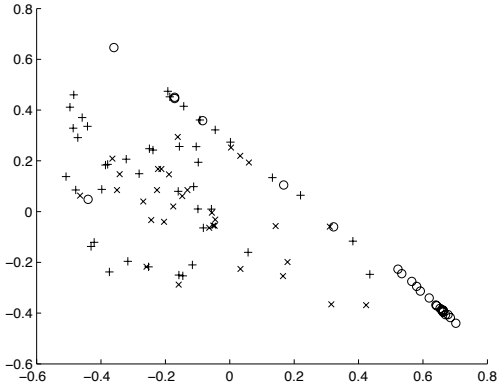
Note that in step 1, we may simply use the values of principle components in PCA as the initial values of $\boldsymbol{Z}$.

Since this method is based on a subset of the whole dataset, the implementation usually proceeds very quickly at the expense of sacrificing some accuracy.

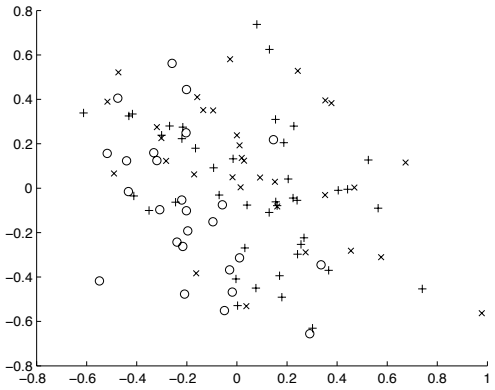
One of the applications of GPLV model is dimension reduction for large-dimensional data. For example, it is usually difficult to classify data $\{\boldsymbol{Y}_i, i = 1, \ldots, n\}$ when D, the dimension of $\boldsymbol{Y}_i$, is large. We can then use a GPLV model to calculate the values of the latent variables $\boldsymbol{z}_i$. The lower dimensional latent variables $\boldsymbol{Z}$ can include most of the information in $\boldsymbol{Y}$ and thus can be used to represent the larger dimensional data $\boldsymbol{Y}$ in statistical inference. We can then use $\boldsymbol{Z}$ to carry out data analysis such as data visualization and classification. To illustrate this idea, we consider the following example.

Example 8.1 (Multiphase oil flow data [Bishop and James, 1993])**.** This dataset has been simulated from an underlying three-dimensional latent variable space. It contains 12 variables (D = 12) representative of three known classes of oil flow within an oil pipeline: stratified, annular, and homogeneous. In this example we are using a random sample with 100 observations (the original set contains 1000 observations). To visualize or classify the data, we need to use the information associated with all 12 variables $\boldsymbol{y}_i = \{y_{id}, d = 1, \ldots, 12\}$. If we just select a subset of those 12 variables, we may lose information and may not be able to classify the oil flow according to the class to which it belongs. As an example, we randomly select two variables, say, y_{ij} and y_{ik}, and plot the pairs of all 100 data points in Figures 8.1(a) and 8.1(b). Note that there is no discernible separation between the classes and it is very difficult to distinguish between them.

We now use a GPLV model with two latent variables. The values of those two latent variables for all 100 data points are estimated. We then use those two-dimensional latent variables to represent the original 12-dimensional variables. The plots of these two latent variables are presented in Figure 8.2(a). Three classes of data points are visualized clearly, and the data points in the same class appear to be grouped together. For comparison, Figure 8.2(b) plots the first two principal components using principal component analysis. Three classes can also be visualized, meaning the two principal components still contain most of information of the original data. However, the separation achieved by PCA in Figure 8.2(b) is less sharp compared with the classification obtained with the GPLV model in Figure 8.2(a) and therefore the two principal components are summarizing the original 12 variables less efficiently (i.e., they are discarding more information). This was also confirmed by Lawrence (2005) from a simulation study. They reported that the classification error rate was only about 0.1% when a GPLV model was used, the error rate increased to about 2.4% when GPLV model with an active set and sparse greedy approximation was used, but the error rate was about 16% when PCA was used.



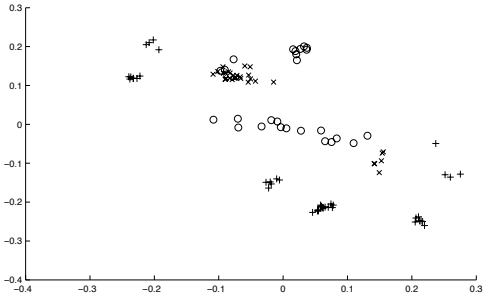
(a)



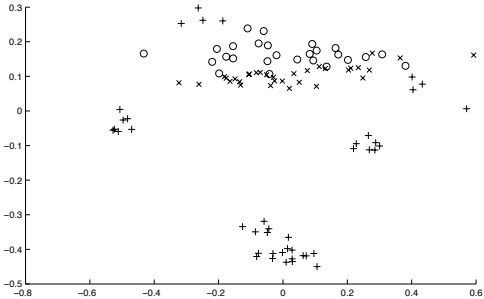
(b)

Figure 8.1 *Oil flow data: plots of two variables randomly selected from the original 12 variables. Three classes are represented by "+", "×", and "○", respectively.*

Further discussion and other examples can be found in Lawrence (2004, 2005), Urtasun and Darrell (2007), and Ko and Fox (2009). The GP latent variable models in (8.17) and (8.18) can be interpreted as an extension of PCA and are known to capture the nonlinearity automatically (Lawrence, 2005). The lower dimensional latent variables can be used to summarize the original variables that are represented in a much larger dimensional space; they can, therefore, be used in classification, clustering, stochastic monitoring, dynamic malfunction detection, and others. However, each latent variable is associated



(a)



(b)

Figure 8.2 *Oil flow data: plots of two latent variables estimated from (a) GPLV models and (b) PCA. Three classes are represented by "+", "×", and "○", respectively.*

with all the observed manifest variables and thus, in turn, makes very difficult its physical interpretation. Such a lack of physical meaning often limits the application of the model. In linear models, this problem, however, is usually addressed by introducing a special model structure. For example, we can define a special loading matrix in a confirmatory factor analysis model or in a structural equation model such that each latent variable is associated with certain variables. Thus, the latent variable can have a clear physical interpretation (see, e.g., Jöreskog and Sörbom, 1999; Brown, 2006; Lee, 2007). The idea can be introduced into the GPLV model by defining a special structure. Specifically, we can define a model such that each f_d depends on a subset of $\{z_1(t), \ldots, z_Q(t)\}$; in other words, each latent variable depends on a subset of $\{y_1(t), \ldots, y_D(t)\}$. This gives some physical interpretation to each latent variable if the model structure is carefully designed. Further details can be found in Serradilla and Shi (2010). This flexible model structure enables us to extend

the model applicability to areas such as those of process monitoring and fault detection in chemical and process engineering.

8.3 Optimal dynamic control using GPR model

The Gaussian process regression model has been successfully applied to nonlinear control systems (see, e.g., Murray-Smith et al., 2003; Sbarbaro et al., 2004; Sbarbaro and Murray-Smith, 2005). Since the GPR model is a nonparametric model, we can use it to capture the nonlinearity of the control system automatically. There is no need to assume a given nonlinear model in advance.

We now revisit the example of renal data discussed in Section 1.3.2 to illustrate the control problem. Let $y(t_i)$ be the level of Hb observed at time t_i for a particular patient, and let $d(t_i)$ be the dose level of epoetin which should be prescribed for the patient at time t_i. Now the problem is how to control the level $d(t_{i+1})$ such that the Hb level $y(t_{i+1})$ comes close to a desired target level (which is 11.8 in this example) at time t_{i+1}; or equivalently, we need to control the dosage change $x(t_{i+1}) = d(t_{i+1}) - d(t_i)$. Note that we used a slightly different notation to the one usually used in the community of system control engineering, in which $d(t_{i+1}) - d(t_i)$ is usually denoted by $x(t_i)$ to emphasize that $x(t_i)$ is the action we need to take at current time t_i, and the action will result in the output $y(t_{i+1})$ at time t_{i+1}.

Technically, a dual adaptive control problem is to find a control input $x(t)$ to minimize the following N-stage objective function or cost function (see, e.g., Fabri and Kadirkamanathan, 1998):

$$J_{dual} = \sum_{j=1}^{N} E\left\{[y(t_{n+j}) - y_r(t_{n+j})]^2 | \mathcal{D}_{t_n}\right\}, \tag{8.26}$$

where $y_r(t_{n+j})$ is the desired target level at time t_{n+j}, and $\mathcal{D}_{t_n}$ is the data observed up to time t_n. Here, $y(t)$ is modeled by

$$y(t_i) = h(\boldsymbol{z}(t_i), x(t_i)) + \varepsilon(t_i),$$

where $h(\cdot,\cdot)$ is a nonlinear function, $\boldsymbol{z}(t_i)$ is a state vector of input covariates, and $x(t_i)$ is a control variable. In renal data, $\boldsymbol{z}(t_i)$ may include the observations of Hb level and the dose level up to time t_i and $x(t_i)$ is the change of dosage. In practice, we often use a one-stage objective function, i.e., $N = 1$. However, it is not feasible to optimize the objective function in (8.26) directly (see, e.g., Fabri and Kadirkamanathan, 1998). We usually consider a so called suboptimal objective function (see also Sbarbaro and Murray-Smith, 2005).

$$\begin{aligned} J &= \mathrm{E}\left\{[y(t_{n+1}) - y_r(t_{n+1})]^2 | \mathcal{D}_{t_n}\right\} + \lambda x(t_{n+1})^2 \\ &= [\mathrm{E}(y(t_{n+1}) | \mathcal{D}_{t_n}) - y_r(t_{n+1})]^2 \\ &\quad + \mathrm{Var}(y(t_{n+1}) | \mathcal{D}_{t_n}) + \lambda x(t_{n+1})^2. \end{aligned} \tag{8.27}$$

Here, parameter λ induces a penalty on high dosage change.

A Gaussian process regression model can be used to model $y(t)$ (see, e.g., Murray-Smith et al., 2003; Sbarbaro and Murray-Smith, 2005). If we have collected data from many different subjects, we could use a Gaussian process functional regression (GPFR) model (Shi and Li, 2010). If we use a GPFR model with the linear functional mean model defined in (5.11), we can express the model as

$$y(t) = \boldsymbol{u}^T \boldsymbol{\beta}(t) + \tau(\boldsymbol{z}(t), x(t)) + \varepsilon(t), \tag{8.28}$$

where $\boldsymbol{u}$ is a vector of scalar covariates such as a patient's weight and gender, $\boldsymbol{z}(t)$ is the state vector of input covariates, and $x(t)$ is a control variable. Both $\boldsymbol{z}(t)$ and $x(t)$ are functional covariates and, although $x(t)$ could be multidimensional, we only consider a single control input in this section, i.e., assuming $x(t)$ is one-dimensional.

Before discussing the general model, we first consider a simple model with the following structure (Sbarbaro and Murray-Smith, 2005):

$$y(t) = \boldsymbol{u}^T \boldsymbol{\beta}(t) + f(\boldsymbol{z}(t)) + x(t) g(\boldsymbol{z}(t)) + \varepsilon(t), \tag{8.29}$$

where both $f(\boldsymbol{z}(t))$ and $g(\boldsymbol{z}(t))$ are nonlinear and depend on the current state vector $\boldsymbol{z}(t)$. We assume GPR models for both nonlinear functions, and that the two GPR models are independent. For convenience, we also assume that the covariance functions, $k(\cdot,\cdot)$, of both Gaussian processes are of the same but have different unknown parameters. There are no difficulties in extending the method to use different covariance structures. Thus, the two GPR models are given by

$$\begin{aligned} f(\boldsymbol{z}(t)) &\sim GPR[0, k(\boldsymbol{\theta}_f)|\boldsymbol{z}(t)], \\ g(\boldsymbol{z}(t)) &\sim GPR[0, k(\boldsymbol{\theta}_g)|\boldsymbol{z}(t)]. \end{aligned} \tag{8.30}$$

The covariance function of $y(t)$ in (8.29) is therefore given by

$$\begin{aligned} & \Psi(t_i, t_j) = \mathrm{Cov}(y(t_i), y(t_j)) \\ = \;& k(\boldsymbol{z}(t_i), \boldsymbol{z}(t_j); \boldsymbol{\theta}_f) + x(t_i) k(\boldsymbol{z}(t_i), \boldsymbol{z}(t_j); \boldsymbol{\theta}_g) x(t_j) + \sigma^2 \delta_{ij}, \end{aligned} \tag{8.31}$$

after a covariance kernel has been chosen properly (see the discussion in Chapter 4). If we have obtained data from M subjects, we then have M realizations of the same process or, in other words, M batches. We can use the method discussed in Chapter 5 to learn about models (8.29) and (8.30) by making inference about the values of $\boldsymbol{\theta}_f$, $\boldsymbol{\theta}_g$ and the parameters involved in the error terms.

We can now consider the control problem for a particular subject assuming that all the unknown parameters in (8.29) and (8.30) are given. Let us assume that we have obtained data at n data points $t_1, \ldots, t_n$ for the subject. The data include the response variable $y(t_i)$, the state vector $\boldsymbol{z}(t_i)$, and $x(t_i)$ for $i = 1, \ldots, n$.

Supposing that the current time is t_n, the objective of the optimal control is to find the optimal value of $x(t_{n+1})$ to minimize the objective function (8.27). To simplify notation, we use t to replace t_{n+1}, thus we need to find $x(t)$ by minimizing the following objective function:

$$J = [\hat{\mu}_y(t) - y_r(t)]^2 + \text{Var}(y(t)|\mathcal{D}) + \lambda x(t)^2, \tag{8.32}$$

where $\hat{\mu}_y(t)$ is the prediction of $y(t)$ when the control input is $x(t)$, and $\mathcal{D}$ denotes the data observed up to time t_n. Using the results given in Section 5.3.1, we can calculate the prediction by the following equation:

$$\hat{\mu}_y(t) = \hat{\mu}(t) + \boldsymbol{H}^T(\boldsymbol{y}_n - \hat{\boldsymbol{\mu}}_n), \tag{8.33}$$

where $\boldsymbol{y}_n = (y(t_1), \ldots, y(t_n))^T$, $\hat{\boldsymbol{\mu}}_n = (\hat{\mu}(t_1), \ldots, \hat{\mu}(t_n))^T$, and $\hat{\mu}(t)$ is given by (5.20). The $\boldsymbol{H}$ is given by

$$\boldsymbol{H}^T = [\boldsymbol{\Psi}(\boldsymbol{t}_n, t)]^T \boldsymbol{\Psi}_n^{-1},$$

where $\boldsymbol{\Psi}_n$ is the covariance matrix of $\boldsymbol{y}_n$ whose element $\boldsymbol{\Psi}(t_i, t_j)$ is given in (8.31), and $\boldsymbol{\Psi}(\boldsymbol{t}_n, t)$ is a $n \times 1$ vector with element $\boldsymbol{\Psi}(t_i, t) = \text{Cov}(y(t_i), y(t))$. From (8.31), we have

$$[\boldsymbol{\Psi}(\boldsymbol{t}_n, t)]^T = \boldsymbol{C}_{n,f}^T + x(t)\boldsymbol{C}_{n,g}^T \boldsymbol{D}_n,$$

where $\boldsymbol{C}_{n,f}$ is an $n \times 1$ vector with element $k(z(t_i), z(t); \boldsymbol{\theta}_f)$, $\boldsymbol{C}_{n,g}$ is defined similarly but with the element $k(z(t_i), z(t); \boldsymbol{\theta}_g)$, and $\boldsymbol{D}_n$ is a diagonal matrix with element $x(t_i)$ for $i = 1, \ldots, n$. Thus, (8.33) can be expressed by

$$\hat{\mu}_y(t) = a + bx(t) \;\text{ with }\; a = \hat{\mu}(t) + \boldsymbol{C}_{n,f}^T \boldsymbol{y}_n^*, \;\; b = \boldsymbol{C}_{n,g}^T \boldsymbol{D}_n \boldsymbol{y}_n^*, \tag{8.34}$$

and $\boldsymbol{y}_n^* = \boldsymbol{\Psi}_n^{-1}(\boldsymbol{y}_n - \hat{\boldsymbol{\mu}}_n)$. The predictive variance of $y(t)$ is given by

$$\text{Var}(y(t)|\mathcal{D}) = \gamma(\psi(t,t) - \boldsymbol{H}^T \boldsymbol{\Psi}_n \boldsymbol{H}), \tag{8.35}$$

where $\gamma = (1 + \boldsymbol{u}^T(\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{u})$, $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M)^T$, $\boldsymbol{u}_i$ are the vector of scalar covariates for the i-th subject, while $\boldsymbol{u}$ is the scalar covariates for the current subject, and $\boldsymbol{\psi}(t,t) = Cov(y(t), y(t))$ is given by (8.31). We can prove the above result by directly using Theorem 5.1 to model (8.28). In addition, based on the fact that

$$\begin{aligned} \Psi(t,t) &= k(\boldsymbol{z}(t), \boldsymbol{z}(t); \boldsymbol{\theta}_f) + x(t)^2 k(\boldsymbol{z}(t), \boldsymbol{z}(t); \boldsymbol{\theta}_g) + \sigma^2 \\ &\overset{\text{d}}{=} C_{t,f} + x(t)^2 C_{t,g} + \sigma^2, \end{aligned}$$

and

$$\boldsymbol{H}^T \boldsymbol{\Psi}_n \boldsymbol{H} = (\boldsymbol{C}_{n,f}^T + x(t)\boldsymbol{C}_{n,g}^T \boldsymbol{D}_n) \boldsymbol{\Psi}_n^{-1} (\boldsymbol{C}_{n,f} + x(t)\boldsymbol{D}_n \boldsymbol{C}_{n,g}),$$

we immediately have the following results:

$$\frac{\partial \hat{\mu}_y(t)}{\partial x(t)} = b = \boldsymbol{C}_{n,g}^T \boldsymbol{D}_n \boldsymbol{y}_n^*, \tag{8.36}$$

and

$$\frac{\partial \mathrm{Var}(y(t)|\mathcal{D})}{\partial x(t)} = 2\gamma[C_{t,g}x(t) - \boldsymbol{C}_{n,g}^T \boldsymbol{D}_n \boldsymbol{\Psi}_n^{-1}(\boldsymbol{C}_{n,f} + x(t)\boldsymbol{D}_n \boldsymbol{C}_{n,g})]. \tag{8.37}$$

The following control law is the consequence of the above results.

Theorem 8.3 (Control law)**.** *The optimal control input based on the objective function* (8.27) *is given by*

$$x^*(t) = \frac{(y_r(t)-a)b + \gamma \boldsymbol{C}_{n,g}^T \boldsymbol{D}_n \boldsymbol{\Psi}_n^{-1} \boldsymbol{C}_{n,f}}{b^2 + \lambda + \gamma[C_{t,g} - \boldsymbol{C}_{n,g}^T \boldsymbol{D}_n \boldsymbol{\Psi}_n^{-1} \boldsymbol{D}_n \boldsymbol{C}_{n,g}]} \tag{8.38}$$

if λ *is given, where a and b are defined in* (8.34)*.*

Proof. From (8.32), (8.36), and (8.37), we have

$$\begin{aligned} \frac{\partial J}{\partial x(t)} &= 2(\hat{\mu}_y - y_r(t))\frac{\partial \hat{\mu}_y}{\partial x(t)} + \frac{\partial \mathrm{Var}(y(t)|\mathcal{D})}{\partial x(t)} + 2\lambda x(t) \\ &= 2[a + bx(t) - y_r(t)]b + 2\lambda x(t) \\ &\quad + 2\gamma[C_{t,g}x(t) - \boldsymbol{C}_{n,g}^T \boldsymbol{D}_n \boldsymbol{\Psi}_n^{-1}(\boldsymbol{C}_{n,f} + x(t)\boldsymbol{D}_n \boldsymbol{C}_{n,g})]. \end{aligned}$$

Letting the above derivative be zero we have the result (8.38). □

The objective function in (8.32) can be modified by

$$J = [\hat{\mu}_y(t) - y_r(t)]^2 + rVar(y(t)|\mathcal{D}) + \lambda x(t)^2.$$

Thus, (8.32) corresponds to $r = 1$, indicating a cautious controller (Fabri and Kadirkamanathan, 1998). If we don't consider the uncertainty caused by model learning, i.e., assuming $r = 0$, the optimal control input is simplified as

$$x^*(t) = \frac{(y_r(t)-a)b}{b^2 + \lambda}.$$

If we further ignore the penalty term in (8.32), i.e., assuming $\lambda = 0$ as well, the optimal control input is $x^*(t) = (y_r(t) - a)/b$; i.e., $x^*(t)$ is simply the solution of the following equation:

$$\hat{\mu}_y(t) = a + bx(t) = y_r(t).$$

This is the method we used in Section 5.4.2 for the example of renal data. It is

fine to use this solution as a guideline in practice. However, an optimal decision needs to consider other factors. For example, in dose-response analysis, we should always avoid an abrupt change of dosage, and then we need to penalize the change in the objective function (8.27) (see more discussion in Murphy, 2003; Rosthoj et al., 2006). The selection of the values of r and λ depends on the circumstances of each individual control system (see, for example, Fabri and Kadirkamanathan, 1998).

In addition to model (8.29) where $y(t)$ depends on the control variable $x(t)$ linearly, we now consider the general model (8.28) but $\tau(\cdot)$ is modeled by a GPR model. Specifically,

$$\tau(\boldsymbol{z}(t), x(t)) \sim GPR[0, k(\boldsymbol{\theta})|\tilde{\boldsymbol{z}}(t)],$$

where $\tilde{\boldsymbol{z}}(t) = (\boldsymbol{z}(t), x(t))^T$. The covariance function of $y(t)$ can therefore be expressed by

$$\Psi(t_i, t_j) = \mathrm{Cov}(y(t_i), y(t_j)) = k(\tilde{\boldsymbol{z}}(t_i), \tilde{\boldsymbol{z}}(t_j); \boldsymbol{\theta}) + \sigma^2 \delta_{ij}.$$

Under this general model, the prediction of the posterior mean and variance of $y(t)$ are given by

$$\begin{aligned} \hat{\mu}_y(t) &= \boldsymbol{u}^T \hat{\boldsymbol{\beta}}(t) + [\boldsymbol{\Psi}(\boldsymbol{t}_n, t)]^T \boldsymbol{\Psi}_n^{-1} (\boldsymbol{y}_n - \hat{\boldsymbol{\mu}}_n), \\ \mathrm{Var}(y(t)|\mathcal{D}) &= \gamma[\Psi(t,t) - [\boldsymbol{\Psi}(\boldsymbol{t}_n, t)]^T \boldsymbol{\Psi}_n^{-1} \boldsymbol{\Psi}(\boldsymbol{t}_n, t), \end{aligned}$$

where

$$[\boldsymbol{\Psi}(\boldsymbol{t}_n, t)]^T = \mathrm{Cov}(y(t), \boldsymbol{y}_n) = [k(\tilde{\boldsymbol{z}}(t), \tilde{\boldsymbol{z}}(t_1)), \ldots, k(\tilde{\boldsymbol{z}}(t), \tilde{\boldsymbol{z}}(t_n))]^T.$$

We further denote that

$$G_{t_i} = \frac{\partial k(\tilde{\boldsymbol{z}}(t), \tilde{\boldsymbol{z}}(t_i))}{\partial x(t)}, \quad G_t = \frac{\partial k(\tilde{\boldsymbol{z}}(t), \tilde{\boldsymbol{z}}(t))}{\partial x(t)},$$

and $\boldsymbol{G}_n = (G_{t_1}, \cdots, G_{t_n})^T$. Thus, we have

$$\frac{\partial \hat{\mu}_y(t)}{\partial x(t)} = \boldsymbol{G}_n^T \boldsymbol{\Psi}_n^{-1} (\boldsymbol{y}_n - \hat{\boldsymbol{\mu}}_n)$$

and

$$\frac{\partial \mathrm{Var}(y(t)|\mathcal{D})}{\partial x(t)} = 2\gamma[G_t - 2\boldsymbol{G}_n^T \boldsymbol{\Psi}_n^{-1} \boldsymbol{\Psi}(\boldsymbol{t}_n, t)].$$

Then differentiating the objective function J in (8.32) gives

$$\begin{aligned} &2[\hat{\mu}_y(t) - y_r(t)] \boldsymbol{G}_n^T \boldsymbol{\Psi}_n^{-1} (\boldsymbol{y}_n - \hat{\boldsymbol{\mu}}_n) \\ &+\gamma[G_t - 2\boldsymbol{G}_n^T \boldsymbol{\Psi}_n^{-1} \boldsymbol{\Psi}(\boldsymbol{t}_n, t)] + 2\lambda x(t) = 0. \end{aligned} \tag{8.39}$$

Since both $\boldsymbol{G}_n$ and $\boldsymbol{\Psi}(\boldsymbol{t}_n, t)$ involve $x(t)$, we cannot find the solution of $x(t)$ in an analytical form. We therefore have to use a numerical method to solve the equation. Further discussion can be found in Shi and Li (2010).

Numerical examples and applications about using GPR dynamic control can be found in, e.g., Sbarbaro et al. (2004), Sbarbaro and Murray-Smith (2005), and Kocijan and Murray-Smith (2005). Some recent developments and applications can be found in, e.g., Pronzato (2008) and Grancharova et al. (2008).

An alternative but commonly used approach in nonlinear system control is based on artificial neural networks. The readers are referred to the details in Narendra and Parthasarathy (1990), Chen and Khalil (1995), and Fabri and Kadirkamanathan (1998) among others.

8.4 RKHS and Gaussian process regression

In this section, we describe briefly an important connection between Reproducing Kernel Hilbert Space (RKHS) and Gaussian process regression. The theory of RKHS and its relationship with smoothing splines have been well reviewed by Wahba (1990), while van der Vaart and van Zanten (2008b) have focused applications in nonparametric Bayesian statistics using Gaussian priors. Instead of giving mathematical details of RKHS, we provide a basic idea of RKHS associated with Gaussian process regression, and discuss some research problems related to this connection. In this regard, we base our discussion on the materials given in Wahba (1990), Seeger (2004), Kakade et al. (2006), and Seeger et al. (2008); further theoretical details and mathematical exposition are referred to Wahba (1990) and van der Vaart and van Zanten (2008b).

Simply speaking, the connection between Gaussian process regression and RKHS is mainly described by the covariance function $k(\cdot,\cdot)$ of the Gaussian process, by the Karhunen-Loéve expansion, and by the Mercer's theorem as briefly discussed in Chapter 1. That is, every mean-zero Gaussian process is defined by an RKHS, and by the Mercer's theorem it is ensured that there exist orthonormal eigenfunctions $\{\phi_j\}$ with corresponding eigenvalues λ_j's given by (Wahba, 1990; Seeger, 2004)

$$\int k(\boldsymbol{x},\boldsymbol{x}')\phi_j(\boldsymbol{x}')d\boldsymbol{x}' = \lambda_j\phi_j(\boldsymbol{x}), \quad j = 1,2,\ldots,$$

if we assume a uniform distribution for $\boldsymbol{x}$ in (1.14). Based on this idea, we can write the covariance kernel as

$$k(\boldsymbol{x},\boldsymbol{x}') = \sum_j \lambda_j\phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}').$$

In other words, the covariance function $k(\boldsymbol{x},\boldsymbol{x}')$ of a Gaussian process can

be identified with the Reproducing Kernel (RK) from an Reproducing Kernel Hilbert Space $\mathcal{H}$. Specifically, as pointed out in Seeger et al. (2008) and Kakade et al. (2006), every positive semidefinite kernel $k(\cdot,\cdot)$ is associated with a unique RKHS, $\mathcal{H}$. It is defined as follows: consider the linear space of all finite kernel expansions over any $x_1,\ldots,x_n$ of the form $f(x)=\sum_{i=1}^{n}\alpha_i k(x,x_i)$ with the inner product

$$\left\langle \sum_i \alpha_i k(\cdot,x_i), \sum_j \beta_j k(\cdot,y_j) \right\rangle_k = \sum_{i,j} \alpha_i \beta_j k(x_i,y_j),$$

and defines the RKHS $\mathcal{H}$ as the completion of this space. Here $\langle\cdot,\cdot\rangle_k$ denotes an inner product associated with a covariance kernel $k(\cdot,\cdot)$. By construction, $\mathcal{H}$ contains all finite kernel expansions $f(x)=\sum_{i=1}^{n}\alpha_i k(x,x_i)$ with

$$\|f\|_k^2 = \alpha^T \boldsymbol{K}\alpha,\ \boldsymbol{K}(i,j)=k(x_i,x_j).$$

The characteristic property of $\mathcal{H}$ is that all (Dirac) evaluation functionals are represented in $\mathcal{H}$ itself by the functions $k(\cdot,x_i)$, meaning that

$$\langle f, k(\cdot,x_i)\rangle_k = f(x_i).$$

This reproducing property indicates that convergence in norm in $\mathcal{H}$ implies pointwise convergence, so all $f\in\mathcal{H}$ are pointwise defined. Based on this framework, Seeger et al. (2008) investigated information consistency of Gaussian process regression methods with commonly used covariance kernels. Using the same approach, Kakade et al. (2006) studied bounds on the regret for nonparametric Bayesian algorithms of Gaussian process regression models.

In addition, the prediction can be explained in the similar framework of:

$$\mathrm{E}(f^*|\boldsymbol{x},\boldsymbol{x}^*) = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}_i,\boldsymbol{x}^*),$$

where the α_i's are from the vector $\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_n)^T=(\boldsymbol{K}(\boldsymbol{x},\boldsymbol{x})+\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}$. Indeed, this is the form given by the Representer theorem (Kimeldorf and Wahba, 1971). The Representer theorem enables us to solve an optimization problem in an infinite dimensional space $\mathcal{H}$ by ensuring the solutions are represented by the linear combination of kernels, i.e., belonging to the span of finite particular kernels (see, e.g., Schölkopf and Smola, 2002).

Furthermore, the RKHS theory provides a useful framework to obtain the explicit functional form that minimizes the regularized risk functional. This connection leads to smoothing splines by incorporating a penalty term in the optimization procedure with Gaussian process prior as a special case for roughness penalties. Further details can be found in Kimeldorf and Wahba (1971), Wahba (1990), and Seeger (2004).

Other kernel methods related to the RKHS framework have been developed, such as kernel PCA (Schölkopf et al., 1998), relevance vector machine (RVM) (Tipping, 2001), and other Bayesian kernel methods. Theoretical studies on general frameworks for RKHS and Bayesian kernel models have also been studied. In particular, the RVM proposed by Tipping (2001) concerns the response y expressed as an expansion of the kernels evaluated at the predictors, based on the idea of the Representer theorem as mentioned before,

$$y(x) = \sum_{j=1}^{n} \alpha_j k(x, x_j) + \varepsilon.$$

By considering suitable priors on α_j and hyper-priors, the RVM provides a sparse solution for regression and classification from a Bayesian perspective (Tipping, 2001; Schölkopf and Smola, 2002). Other Bayesian kernel models such as the Laplacian process model and the Student's t process can be found in Schölkopf and Smola (2002) and Rasmussen and Williams (2006).

In order to understand the connection between RKHS and kernel methods (which include the Gaussian process regression model) with a generalized framework, Pillai et al. (2007) investigated the theoretical foundations that characterize the function space for Bayesian kernel models in terms of the integral operator and signed measure. They discussed the construction of priors on space of signed measures using Gaussian and Lévy processes as well as its computational issues. For more details, we refer to Pillai et al. (2007) and references therein.

This page intentionally left blank

Appendix

A.1 An extension of Schwartz's theorem for posterior consistency

Theorem A.1. *(Choi and Schervish, 2007) Let $\{Z_i\}_{i=1}^{\infty}$ be independently distributed with densities $\{f_i(\cdot;\theta)\}_{i=1}^{\infty}$, with respect to a common σ-finite measure, where the parameter θ belongs to an abstract measurable space Ω. The densities $f_i(\cdot;\theta)$ are assumed to be jointly measurable. Let $\theta_0 \in \Omega$ and let P_{θ_0} stand for the joint distribution of $\{Z_i\}_{i=1}^{\infty}$ when θ_0 is the true value of θ. Let $\{U_n\}_{n=1}^{\infty}$ be a sequence of subsets of Ω. Let θ have prior Π on Ω. Define*

$$\begin{aligned} \Lambda_i(\theta_0,\theta) &= \log\frac{f_i(Z_i;\theta_0)}{f_i(Z_i;\theta)}, \\ KL_i(\theta_0,\theta) &= \mathrm{E}_{\theta_0}(\Lambda_i(\theta_0,\theta)), \\ V_i(\theta_0,\theta) &= \mathrm{V}ar_{\theta_0}(\Lambda_i(\theta_0,\theta)). \end{aligned}$$

(A1) Prior positivity of neighborhoods.
Suppose that there exists a set B with $\Pi(B) > 0$ such that

(i) $\sum_{i=1}^{\infty} \frac{V_i(\theta_0,\theta)}{i^2} < \infty$, $\forall\ \theta \in B$,

(ii) *For all $\varepsilon > 0$, $\Pi(B \cap \{\theta : KL_i(\theta_0,\theta) < \varepsilon \text{ for all } i\}) > 0$.*

(A2) Existence of tests.
Suppose that there exist test functions $\{\Phi_n\}_{n=1}^{\infty}$, sets $\{\Omega_n\}_{n=1}^{\infty}$, and constants $C_1, C_2, c_1, c_2 > 0$ such that

(i) $\sum_{n=1}^{\infty} \mathrm{E}_{\theta_0}\Phi_n < \infty$,

(ii) $\sup_{\theta \in U_n^C \cap \Omega_n} \mathrm{E}_{\theta}(1-\Phi_n) \le C_1 e^{-c_1 n}$,

(iii) $\Pi(\Omega_n^C) \le C_2 e^{-c_2 n}$.

Then

$$\Pi(\theta \in U_n^C | Z_1, \ldots, Z_n) \to 0 \quad a.s.[P_{\theta_0}]. \tag{A.1}$$

Some concepts and notations used above are defined as follows.

Definition A.1. *Let $\theta_0 \in \Omega$ and let $P_{\theta_0}^{\infty}$ stand for the joint distribution of $\{Z_i\}_{i=1}^{\infty}$*

when θ_0 is the true value of θ. Then $\Pi(U^C|Z_1,Z_2,\ldots,Z_n)$ is said to go to 0 exponentially with $P^{\infty}_{\theta_0}$ probability 1, if there exists a $w>0$ such that

$$P^{\infty}_{\theta_0}\left\{\Pi(U^C|Z_1,Z_2,\ldots,Z_n) > e^{-nw} \text{ i.o. }\right\} = 0,$$

where i.o. stands for "infinitely often".

To establish exponential consistency, the Kullback-Leibler support condition, related to the prior positivity condition $(A1)$ of Theorem A.1, is the first one to be verified, and is stated in terms of the Kullback-Leibler neighborhood.

Definition A.2. *The Kullback-Leibler (KL) divergence is denoted by $KL(\theta_0,\theta) = E_{\theta_0}\ \log[f_\theta/f_{\theta_0}]$. A KL neighborhood $KL_\varepsilon(\theta_0)$ of θ_0 is denoted by $\{\theta : KL(\theta_0,\theta) < \varepsilon\}$. A point θ_0 is said to be in the KL support of Π if for all $\varepsilon>0, \Pi(KL_\varepsilon(\theta_0))>0$.*

The following proposition (Choi and Ramamoorthi, 2008) shows that the Kullback-Leibler support condition takes care of the denominator in (2.11). Note that this proposition holds for i.i.d. observations. In the case of independent but nonidentically distributed observations as given in (2.3), additional conditions are needed as described in $(A1)$.

Proposition A.1. *(Choi and Ramamoorthi, 2008) If θ_0 is in the KL support of Π then for all $w>0$,*

$$\lim_{n\to\infty} e^{nw} J(Z_1,Z_2,\ldots,Z_n) = \infty \ a.s.\ P^{\infty}_{\theta_0}.$$

The exponential consistency, i.e., the exponential convergence of the posterior probability to 0, would follow if it can be established that there exists $w_0>0$ such that $e^{nw_0}J_{U^C}(Z_1,Z_2,\ldots,Z_n)\to 0\ a.s.\ P^{\infty}_{\theta_0}$. Sufficient conditions to achieve this are stated in terms of strong separation (Schwartz, 1965), equivalent to the existence of an unbiased test for testing $H_0:\theta=\theta_0$ versus $H_1:\theta\in U^C$ (Le Cam, 1986). Such a test is also known as a uniformly (exponentially) consistent test (Ghosh and Ramamoorthi, 2003; Choi and Ramamoorthi, 2008) and is related to the condition $(A2)$ of Theorem A.1. The existence of uniformly consistent tests is alternatively represented by the entropy condition, and the general posterior consistency theorems have been stated in various ways in terms of both the entropy condition and the KL support condition (see, e.g., Barron et al., 1999; Ghosal et al., 1999; Walker, 2004). A comparative discussion on them also can be found in Choi and Ramamoorthi (2008). The formal proof of Theorem A.1 and the technical details can be found in Choi and Schervish (2007). Further remarks on the posterior consistency can be found in Choi and Ramamoorthi (2008).

A.2 Assumption P

Here, we state the general assumption, Assumption P, about the Gaussian process prior distribution that guarantees $\Pi(B) > 0$, the key condition of $(A1)$ in Theorem A.1.

Assumption P.

P1. For all $n \geq 1$, all $w > 0$, and all $x_1, \ldots, x_n \in [0,1]$, the n-variate covariance matrix, $\boldsymbol{K}$ with $\boldsymbol{K}(i,j) = k(x_i, x_j; w)$, is nonsingular.

P2. The covariance function, $k(x, x'; w)$, has the form $g(w|x - x'|)$, where $g(x)$ is a positive multiple of a nowhere zero density function on $\mathcal{R}$ and four times continuously differentiable on $\mathcal{R}$.

P3. The mean function $\mu(x)$ of the Gaussian process is continuously differentiable in $[0,1]$.

P4. There exists $0 < \delta < 1/2$ and $b_1,\ b_2 > 0$ such that

$$\kappa\left\{w > n^{\delta}\right\} = \Pr\left\{w > n^{\delta}\right\} < b_1 \exp(-b_2 n),\ \ \forall n \geq 1.$$

A direct consequence of **P1** is that the prior variance of $f(x_i)$, $\boldsymbol{K}(i,i)$, is always positive. Otherwise, the covariance matrix generated by the covariance function cannot always be nonsingular. **P2** and **P3** are sufficient to make sure that the support of the GP prior contains every continuously differentiable function.

Note that the prior positivity of $(A1)$, i.e., $\Pi(B > 0)$, depends on the support of the Gaussian process prior, and without a smoothing parameter w in the covariance function of the Gaussian process, only functions in a reproducing kernel Hilbert space of the fixed covariance function would be in the support of the Gaussian process prior, which could not suffice $\Pi(B) > 0$ (see details in Ghosal and Roy, 2006; Tokdar and Ghosh, 2007; van der Vaart and van Zanten, 2008b).

Another important consequence of **P2** and **P3** is that there exists a constant $K_2 > 0$ such that

$$g_0(h) = 1 - \frac{K_2}{2} h^2 + O(h^4), \tag{A.2}$$

where $g_0(h) = g(|h|)/g(0)$, which can be seen easily by Taylor's expansion of $g_0(h)$. In addition, these two conditions guarantee the existence of a continuous sample derivative $f'(x)$ with probability one, and that $f'(x)$ is also a Gaussian process. Many covariance functions such as those in Table 4.1 satisfy Assumptions **P1**-**P3**. This condition is also useful for the construction of uniformly consistent tests, which will be explained further in the verification of $(A2)$ in Section A.3. Specifically, we show as follows that (A.2) is held for each covariance function given in Table 4.1.

(a) The squared-exponential covariance function: see equation (2.13).

(b) Cauchy (rational quadratic) covariance function:

$$g_0(h) = \frac{1}{1+h^2} = 1 - h^2 + O(h^4), \text{ as } h \to 0$$

and $\dfrac{1}{1+h^2}$ is a positive multiple of the Cauchy density.

(c) Matérn covariance function with $v > 2$:

$$g_0(h) = \frac{1}{\Gamma(v)2^{v-1}}(wh)^v \mathcal{K}_v(wh),$$

where $w > 0$ and $\mathcal{K}_v(x)$ is a modified Bessel function of order ν. It is known (Abrahamsen 1997, p. 43) that for $m < v$,

$$\left.\frac{d^{2m-1}\boldsymbol{K}_v(h)}{dh^{2m-1}}\right|_{h=0} = 0,$$

and

$$-\infty < \left.\frac{d^{2m}\boldsymbol{K}_v(h)}{dh^{2m}}\right|_{h=0} < 0.$$

Consequently, if $v > 2$, then there exists a constant $\xi > 0$ such that

$$\boldsymbol{K}_v(h) = \boldsymbol{K}_v(0) - \xi h^2 + O(h^4), \text{ as } h \to 0.$$

In addition, the Matérn covariance function can be shown to be integrable. From the definition of the Matérn covariance function, it is known that when the power, v, is $m + \frac{1}{2}$ with m, a nonnegative integer, the Matérn covariance function is of the form $e^{-\alpha|t|}$ times a polynomial in $|t|$ of degree m (see Stein, 1999, p. 31). Therefore, if v is in the form of $m + \frac{1}{2}$, the Matérn covariance function is integrable.

A.3 Construction of uniformly consistent tests

To verify condition (A2) in Theorem A.1 under a Gaussian process regression model, we construct uniformly consistent tests that have exponentially small type I and II errors in the following way. Let $M_n = O(n^\alpha)$ for some $1/2 < \alpha < 1$ and define Ω_n,

$$\Omega_n = \left\{ f(\cdot) \ : \ \sup_{x\in[0,1]} |f(x)| < M_n \ , \quad \sup_{x\in[0,1]} |f'(x)| < M_n \right\}. \tag{A.3}$$

Now, the n-th test is constructed by combining a collection of tests, one for each of the finitely many elements of Ω_n. Those finitely many elements come

from the covering of Ω_n by small balls. In addition, it is straightforward from Theorem 2.7.1 of van der Vaart and Wellner (1996) that there exists a constant C' such that the ε-covering number $N(\varepsilon, \Omega_n, \|\cdot\|_\infty)$ of Ω_n satisfies

$$\log N(\varepsilon, \Omega_{1n}, \|\cdot\|_\infty) \leq \frac{C' M_n}{\varepsilon^d},$$

where $\|\cdot\|_\infty$ is the supremum norm. For each n and each ball in the covering of Ω_{1n}, we find a test with small type I and type II error probabilities. Then we combine the tests and show that they satisfy subconditions (i) and (ii) of $(A2)$. Theorem A.2 states the existence of tests for the fixed design case based on $W_{\varepsilon,n}$. For more technical details about the proof, readers may refer to Choi and Schervish (2007).

Theorem A.2. *Suppose that the values of the covariate in* $[0,1]$ *arise according to a fixed design. Let P be the joint distribution of* $\{Y_n, n = 1^\infty\}$. *Then there exist test functions* $\{\Phi_n\}$ *and a constant* $C_5 > 0$ *that satisfy :*

(i) $\sum_{n=1}^{\infty} \mathrm{E}_{P_{f_0}} \Phi_n < \infty,$

(ii) $\sup_{f \in W^C_{\varepsilon,n} \bigcap \Omega_n} \mathrm{E}_P(1 - \Phi_n) \leq \exp(-C_5 n).$

As the construction of sieves described above, if the Gaussian process prior distribution for the regression function, Π, assigns exponentially small probability to the two sets $\Omega^C_{n,0} = \{f(\cdot) : \|f\|_\infty > M_n\}$ and $\Omega^C_{n,1} = \{f(\cdot) : \|f'\|_\infty > M_n\}$, then the condition (iii) of $(A2)$ holds. Note that $\|f\|_\infty = \sup_{x \in [0,1]} |f(x)|$.

The following lemma is useful for the probability bound for Ω^C_n and the existence of sample path derivatives of a Gaussian process prior.

Lemma A.1. *Given the smoothing parameter, w, let* $f(x)$ *be a mean zero Gaussian process on* $[0,1]$ *with a squared exponential covariance function. Then* $f(x)$ *has continuously differentiable sample paths and the first derivative process,* $f'(x)$, *is also a Gaussian process with a suitable covariance function. Further, there exist constants* d_1 *and* d_2, *such that*

$$\Pr\left\{ \sup_{0 \leq x \leq 1} |f(x)| > M \right\} \leq 2\exp\left(-d_1 M^2\right) \tag{A.4}$$

and

$$\Pr\left\{ \sup_{0 \leq x \leq 1} |f'(x)| > M \right\} \leq 2\exp\left(-d_2 M^2\right). \tag{A.5}$$

The formal proof of Lemma A.1 and the result in (2.14) can be found in Proposition A.2.7 of van der Vaart and Wellner (1996), Theorem 5 of Ghosal and Roy (2006) and Choi (2005).

A.4 Hybrid Monte Carlo algorithm

A hybrid Monte Carlo (HMC) algorithm was needed in Chapter 3 and other chapters for generating samples from an appropriate posterior distribution. The HMC, as first introduced by Duane et al. (1987), was originally applied in a molecular simulation. Based on the Metropolis acceptance-rejection rule, it produces Monte Carlo samples for a given target distribution. Here, we provide further details about the sampling from $p(\boldsymbol{\theta}|\mathcal{D})$ in (3.14) based on a HMC algorithm.

To illustrate the method, we write $p(\boldsymbol{\theta}|\mathcal{D}) \propto \exp(-\mathcal{E})$, where $\mathcal{E}$ is called potential energy. The idea of HMC is to create a fictitious dynamical system where the parameter vector of interest, $\boldsymbol{\theta}$, is augmented by a set of latent variables, $\boldsymbol{\phi}$, with the same dimension as that of $\boldsymbol{\theta}$. The kinetic energy is defined as a function of the associated momenta: $\mathcal{K}(\boldsymbol{\phi}) = \frac{1}{2}\sum \phi_i^2/\lambda$. Here, the latent variables have normal distribution independently with zero mean and variance λ. The total energy of the system, $\mathcal{H}$, is defined as the sum of the kinetic energy, $\mathcal{K}$, and the potential energy, $\mathcal{E}$. Therefore, HMC samples are drawn from the joint distribution $p(\boldsymbol{\theta},\boldsymbol{\phi}|\mathcal{D}) \propto \exp(-\mathcal{H}) = \exp(-\mathcal{E}-\mathcal{K})$.

One sweep of a variation of the HMC algorithm (see, e.g., Horowitz, 1991; Rasmussen, 1996; Neal, 1997) is given as follows:

(i) Starting from the current state $(\boldsymbol{\theta},\boldsymbol{\phi})$, calculate the new state $(\boldsymbol{\theta}(\varepsilon),\boldsymbol{\phi}(\varepsilon))$ by the following "leapfrog" steps with step size ε:

$$\begin{aligned}
\phi_i(\frac{\varepsilon}{2}) &= \phi_i - \frac{\varepsilon}{2}\frac{\partial \mathcal{E}}{\partial \theta_i}(\boldsymbol{\theta}),\\
\theta_i(\varepsilon) &= \theta_i + \varepsilon\phi_i(\frac{\varepsilon}{2})/\lambda,\\
\phi_i(\varepsilon) &= \phi_i(\frac{\varepsilon}{2}) - \frac{\varepsilon}{2}\frac{\partial \mathcal{E}}{\partial \theta_i}(\boldsymbol{\theta}(\varepsilon)),
\end{aligned}$$

where $\partial\mathcal{E}(\boldsymbol{\theta})/\partial\theta_i$ is the first derivative of $\mathcal{E}$ evaluated at $\boldsymbol{\theta}$.

(ii) The new state $(\boldsymbol{\theta}^*,\boldsymbol{\phi}^*)$ is such that

$$(\boldsymbol{\theta}^*,\boldsymbol{\phi}^*) = \begin{cases} (\boldsymbol{\theta}(\varepsilon),\boldsymbol{\phi}(\varepsilon)), & \text{with probability } \min(1, \frac{p(\boldsymbol{\theta},\boldsymbol{\phi})}{p(\boldsymbol{\theta}(\varepsilon),\boldsymbol{\phi}(\varepsilon))});\\ (\boldsymbol{\theta},-\boldsymbol{\phi}), & \text{otherwise,}\end{cases}$$

where $p(\boldsymbol{\theta},\boldsymbol{\phi})/p(\boldsymbol{\theta}(\varepsilon),\boldsymbol{\phi}(\varepsilon)) = \exp[\mathcal{H}(\boldsymbol{\theta}(\varepsilon),\boldsymbol{\phi}(\varepsilon)) - \mathcal{H}(\boldsymbol{\theta},\boldsymbol{\phi})]$.

(iii) Generate v_i from the standard Gaussian distribution, and update the parameter ϕ_i to $\alpha\phi_i^* + \sqrt{1-\alpha^2}v_i$.

Here, one example of the setting is $\varepsilon = 0.5N_m^{-1/2}$, $\lambda = 1$, and $\alpha = 0.95$ (see Rasmussen, 1996).

A systematic discussion on HMC methods can be found in Liu (2001).

A.5 Differentiability theorems

The following theorem presents sufficient conditions that the sample paths of covariance functions are continuously differentiable; see discussions given in Section 4.1.2.

Theorem A.3. *(Cramer and Leadbetter, 1967, p. 171) Let $\sigma(x,u)$ be an isotropic correlation function and $\sigma(x,u) = r(|x-u|)$. Suppose that, for some $a > 3$ and $\lambda > 0$, $r(h)$ has the expansion*

$$r(h) = 1 - \frac{\lambda h^2}{2} + O\left\{ \frac{h^2}{|\log|h||^a} \right\}.$$

Then, there exists an equivalent process, possessing continuously differentiable sample paths with probability one.

The generalization of the differentiability theorem above can be obtained for the nonstationary or anisotropic covariance function case. For the one-dimensional case, $Q = 1$, we have the following theorem.

Theorem A.4. *(Cramer and Leadbetter, 1967, p. 185) Suppose the mean function, $\mu(x)$, has a continuous derivative, $\mu'(x)$, in a bounded interval, $0 \le x \le T$. Let $r(x,u)$ have a continuous mixed second derivative $r_{11}(x,u) = \partial^2 r/\partial x \partial u$ satisfying, for some constants $C > 0$, $a > 3$, $0 \le x \le T$, and all sufficiently small h,*

$$\Delta_h \Delta_h r_{11}(x,x) \le \frac{C}{|\log|h||^a},$$

where $\Delta_h \Delta_k r(x,u) = r(x+h,u+k) - r(x+h,u) - r(x,u+k) + r(x,u)$. Then, there exists an equivalent process, possessing a sample derivative that is continuous on $0 \le x \le T$, with probability one.

A.6 Asymptotic properties of the empirical Bayes estimates

The empirical Bayes estimate of $\boldsymbol{\theta}$, as discussed in Section 3.1, is calculated by maximizing the marginal log-likelihood defined in (3.7). Since the observations are correlated, we need to discuss the regularity conditions and the consistency theory based on dependent observations. Hereafter, we build on the results given in Basawa and Prakasa Rao (1980).

Let $\boldsymbol{Y}^n = (Y_1, \ldots, Y_n)$, $n \ge 1$, be a sequence of random samples with density $p(\boldsymbol{y}^n; \boldsymbol{\theta}) = p(y_1, \ldots, y_n; \boldsymbol{\theta})$. Let $\boldsymbol{\theta}_0$ be the true value of $\boldsymbol{\theta}$. Let

$$p_k(\boldsymbol{\theta}) = p(\boldsymbol{y}^k; \boldsymbol{\theta}) / p(\boldsymbol{y}^{k-1}; \boldsymbol{\theta})$$

for every $k \ge 1$. Assume that the function $p_k(\boldsymbol{\theta})$ is twice differentiable with respect to $\boldsymbol{\theta}$ for all $\boldsymbol{\theta}$ in a neighborhood I of $\boldsymbol{\theta}_0$ and all $\boldsymbol{y}^k$. Further, assume that the support of $p(\boldsymbol{y}^n; \boldsymbol{\theta})$ is independent of $\boldsymbol{\theta} \in I$. Define $\phi_k(\boldsymbol{\theta}) = \log p_k(\boldsymbol{\theta})$ and let $\dot{\phi}_k(\boldsymbol{\theta})$ be the $p \times 1$ vector whose i-th component is $\dot{\phi}_{k,i} = \frac{\partial}{\partial \boldsymbol{\theta}_i} \phi_k(\boldsymbol{\theta})$ and

$\ddot{\phi}_k(\boldsymbol{\theta})$ be the $p \times p$ matrix whose (ij)-th component is $\ddot{\phi}_{k,i,j} = \frac{\partial^2}{\partial\boldsymbol{\theta}_i \partial\boldsymbol{\theta}_j}\phi_k(\boldsymbol{\theta})$. For simplicity, we formulate the regularity conditions for the one-dimensional case. Denote

$$U_k(\theta) = \dot{\phi}_k(\theta),\ V_k(\theta) = \ddot{\phi}_k,\ U_k = U_k(\theta_0),\ V_k = V_k(\theta_0).$$

Let $L_n(\theta) = \log p(\boldsymbol{y}^n;\theta)$. Let $\mathcal{F}_n$ be the σ-field generated by Y_j, $1 \le j \le n$ and $\mathcal{F}_0$ be the trivial σ-field. Assume that the following conditions are satisfied:

(C1) $\phi_k(\theta)$ is thrice differentiable with respect to θ for all $\theta \in I$. Let $W_k(\theta) = \dddot{\phi}_k(\theta)$ be the third derivative of $\phi_k(\theta)$ with respect to θ.

(C2) Differentiation twice with respect to θ of $p(\boldsymbol{y}^n;\theta)$ under the integral sign is permitted for $\theta \in I$ in $\int p(\boldsymbol{y}^n;\theta)d\mu^n(\boldsymbol{y}^n)$.

(C3) $\mathrm{E}|V_k| < \infty$, $\mathrm{E}|Z_k| < \infty$ where $Z_k = V_k + U_k^2$.

Define the random variables

$$i_k(\theta_0) = \mathrm{Var}[U_k|\mathcal{F}_{k-1}] = \mathrm{E}[U_k^2|\mathcal{F}_{k-1}]$$

and

$$I_n(\theta_0) = \sum_{k=1}^{n} i_k(\theta_0).$$

Let $S_n^* = \sum_{i=1}^n V_k + I_n(\theta_0)$.

In addition to (C1)–(C3), assume that the following condition holds.

(C4) There exists a sequence of constants $K(n) \to \infty$ as $n \to \infty$ such that

(i) $\{K(n)\}^{-1} S_n \xrightarrow{p} 0$.

(ii) $\{K(n)\}^{-1} S_n^* \xrightarrow{p} 0$.

(iii) there exists $a(\theta_0) > 0$ such that, for every $\varepsilon > 0$,

$$P[\{K(n)\}^{-1} I_n(\theta_0) \ge 2a(\theta_0)] \ge 1 - \varepsilon$$

for all $n \ge N(\varepsilon)$, and

(iv) $\{K(n)\}^{-1} \sum_{k=1}^n \mathrm{E}|W_k(\theta)| < M < \infty$ for all $\theta \in I$ and for all n.

We recall Theorem 2.1 in Basawa and Prakasa Rao (1980) as follows.

Theorem A.5. *Under the above regularity conditions, (C1)–(C4), the likelihood equation has a root $\hat{\boldsymbol{\theta}}_n$ with $P_{\boldsymbol{\theta}_0}$-probability approaching one that is consistent for $\boldsymbol{\theta}_0$ as $n \to \infty$.*

We now apply this to the empirical Bayes estimates $\hat{\boldsymbol{\theta}}_n$.

Proposition A.2. *Also, under regularity conditions (C1)–(C4), the marginal likelihood equation in (3.7) has a root $\hat{\boldsymbol{\theta}}_n$ with $P_{\boldsymbol{\theta}_0}$-probability approaching one which is consistent for $\boldsymbol{\theta}_0$ as $n \to \infty$. In addition, there exists a sequence r_n such that $r_n \to \infty$ as $n \to \infty$ and*

$$r_n^{-1} l_n'(\boldsymbol{\theta}) = O_p(1) \text{ and } \|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0\| = O_p(r_n^{-1}). \tag{A.6}$$

Proof. Notice that the marginal distribution of $\boldsymbol{Y}^n = (Y_1, \ldots, Y_n)^T$, $n \geq 1$, has a multivariate normal distribution with mean $\mathbf{0}$ and covariance $\boldsymbol{\Psi}^{\boldsymbol{\theta}}_{N\times N}$. In this case, also note that $\boldsymbol{y}^k$ has a nonsingular $N_k(\mathbf{0}_k, \boldsymbol{\Psi}^{\boldsymbol{\theta}}_{k\times k})$ distribution. Thus, from standard theory of the multivariate normal distribution, $p_k(\boldsymbol{\theta})$, the conditional probability density of Y_k given $\boldsymbol{Y}^{k-1}$ is also a normal density with mean $m_k(\boldsymbol{\theta})$ and variance $v_k(\boldsymbol{\theta})$, where $m_k(\boldsymbol{\theta})$ and $v_k(\boldsymbol{\theta})$ are some functions of $\boldsymbol{\theta}$, determined by a linear combination of the matrices $\boldsymbol{\Psi}^{\boldsymbol{\theta}}_{k\times k}$ and its inverse.

Thus, without loss of generality, assuming that θ is a scalar, $\phi_k(\theta)$ and its derivatives are given by

$$\begin{aligned}
\phi_k(\theta) &= -\log(\sqrt{2\pi m_k(\theta)}) - \left\{ \frac{1}{2v_k(\theta)} (y_k - m_k(\theta))^2 \right\} \\
\dot{\phi}_k(\theta) &= -\frac{m_k'(\theta)}{m_k(\theta)} + \frac{v_k'(\theta)}{2v_k(\theta)^2}(y_k - m_k(\theta))^2 - \frac{(y_k - m_k(\theta))}{v_k(\theta)} m_k'(\theta) \\
\ddot{\phi}_k(\theta) &= A_k(\theta)(y_k - m_k(\theta))^2 + B_k(\theta)(y_k - m_k(\theta)) + C_k(\theta),
\end{aligned}$$

where $A_k(\theta), B_k(\theta), C_k(\theta)$ are some functions of θ, based on the first and second derivatives of $m_k(\theta)$ and $v_k(\theta)$.

Notice that $z_k = (y_k - m_k(\theta))/\sqrt{v_k(\theta)}$ has a standard normal distribution and its square has a χ^2 distribution, given y^{k-1}. Therefore, it follows that $(C1)$–$(C3)$ hold under the nonsingular normal distribution with suitable mean and variance, thrice differentiable with respect to θ. In addition, since the conditional distribution of z_k is a nondegenerate normal distribution, there exists the constants $M_1 > 0$ and $m_1 > 0$ such that

$$i_k(\theta_0) = m_1 \leq \mathrm{Var}[U_k | \mathcal{F}_{k-1}] < M_1.$$

Since the distribution of z_k is determined independently of k, the constants M_1 and m_1 are achieved uniformly on k.

Define $K(n) = I_n(\theta_0)$. Then, it is easy to see $K(n) = O(n)$, and thus satisfies $(i) - (iii)$ in (C4). In addition, the third derivative of ϕ_k, $\dddot{\phi}_k(\theta)$, is also given based on the first, the second, and the third derivatives of $m_k(\theta)$ and $v_k(\theta)$. Note that as $m_k(\beta)$ and $v_k(\beta)$ are thrice differentiable with respect to θ for all $\theta \in I$, it is clear that the condition (iv) of $(C4)$ also holds. Therefore, the solution of the likelihood equation $\boldsymbol{\theta}_n$ is consistent for $\boldsymbol{\theta_0}$ by Theorem A.5.

In order to check the asymptotic normality, we need to verify additional conditions for asymptotic normality in Basawa and Prakasa Rao (1980). However, since convergence in probability implies convergence in distribution, it is certain that there exists a sequence r_n such that

$$r_n^{-1} l_n'(\boldsymbol{\theta}) = O_p(1) \text{ and } \|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta_0}\| = O_p(r_n^{-1}).$$

□

A.7 Asymptotic properties for the penalized GPR models

Proofs of Theorems 4.2 and 4.3 are quite similar to those of Theorem 1 and Lemma 1 in Fan and Li (2001) but different, in that we deal with the likelihood based on dependent observations, for which regularity conditions were described previously in Appendix A.6. Therefore, instead of $\sqrt{n}$-consistent estimator for MLE based on an independent and identically distributed sample, we base the proofs of Theorems 4.2 and 4.3 on an r_n-consistent estimator and regularity conditions. Hence, we simplify the proof of each theorem by stating necessary steps only. For technical details, the interested reader may refer to the original proof based on independent and identically distributed observations in Fan and Li (2001).

Proof of Theorem 4.2. Let $\alpha_n = r_n^{-1} + a_n$. What we need to show is that for any given $\varepsilon > 0$, there exists a large constant C such that

$$Pr\left(\sup_{\|\boldsymbol{u}\|=C} l_p(\boldsymbol{\theta}_0 + \alpha_n \boldsymbol{u}) > l_p(\boldsymbol{\theta}_0) \right) \geq 1 - \varepsilon. \tag{A.7}$$

(A.7) shows that there exists a local minimum in the ball $\{\boldsymbol{\theta}_0 + \alpha_n \boldsymbol{u} : \|\boldsymbol{u}\| \leq C\}$ with probability no less than $1 - \varepsilon$. Consequently, there is a local minimizer $\hat{\boldsymbol{\theta}}$ such that $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\| = O_p(\alpha_n)$.

As conditions (4.25) and (4.26) specified in Section 4.3.4 hold for P_{λ_n}, then

$$\begin{aligned} D_n(u) &= nl_p(\boldsymbol{\theta}_0 + \alpha_n \boldsymbol{u}) - nl_p(\boldsymbol{\theta}_0) \\ &= -\left(l_n(\boldsymbol{\theta}_0 + \alpha_n \boldsymbol{u}) - l_n(\boldsymbol{\theta}_0)\right) \\ &\quad + n \sum_{q=1}^{Q} \left(P_{\lambda_n}(w_q^0 + \alpha_n u_q) - P_{\lambda_n}(w_q^0) \right) \\ &\geq A_n - B_n. \end{aligned}$$

For A_n and B_n, we use the standard argument on the Taylor expansion of the likelihood and penalty functions, respectively.

Let $l_n'(\boldsymbol{\theta}_0)$ be the gradient vector of l_n, evaluated at $\boldsymbol{\theta_0}$. Then, we have

$$A_n = \alpha_n l_n'(\boldsymbol{\theta}_0)\boldsymbol{u} - \frac{1}{2}\boldsymbol{u'} I_n(\boldsymbol{\theta_0})\boldsymbol{u} n \alpha_n^2 \{1 + o_p(1)\}, \tag{A.8}$$

where $I_n(\boldsymbol{\theta_0})$ is defined as in the statement of regularity conditions.

Note that $r_n^{-1} l_n'(\boldsymbol{\theta_0}) = O_p(1)$, as in (A.6), and the term with the penalty function B_n is the same as in (A.2) of Fan and Li (2001). Thus, the remaining steps are identical to the proof of Theorem 1 in Fan and Li (2001). Hence, this completes the proof. □

Proof of Theorem 4.3. The proof is based on Lemma 1 and Theorem 2 in Fan and Li (2001). For the sparsity, it is sufficient to prove

$$\lim_{n\to\infty} P(\hat{\boldsymbol{w}}_{\mathcal{B}} = \mathbf{0}) = 1 \text{ if } \boldsymbol{w}^0_{\mathcal{B}} = \mathbf{0}. \tag{A.9}$$

From the first-order Taylor expansion of the partial derivatives of the log-likelihood, the partial derivatives of the penalized minus log-likelihood function, $\partial l_p(\hat{\boldsymbol{\theta}}_n)/\partial w_q$, is given as

$$\begin{aligned} n\frac{\partial l_p(\boldsymbol{\theta})}{\partial w_q} &= -\frac{\partial l_n(\boldsymbol{\theta}_0)}{\partial w_q} - \frac{1}{2}\sum_{q'\neq q}\frac{\partial^2 l_n(\boldsymbol{\theta}^*)}{\partial w_{q'}\partial w_q}(\hat{w}_{q'} - w_q^{(0)}) \\ &\quad + n\frac{\partial P_{\lambda_n}(\hat{\boldsymbol{\theta}}_n)}{\partial w_q}, \end{aligned} \tag{A.10}$$

where $\boldsymbol{\theta}^*$ lies in-between $\hat{\boldsymbol{\theta}}_n$ and $\boldsymbol{\theta}_0$. As $\hat{\boldsymbol{\theta}}_n$ is an r_n consistent estimator, and both

$$\frac{\partial l_n(\boldsymbol{\theta}_0)}{\partial w_q} \text{ and } \frac{1}{2}\sum_{q'\neq q}\frac{\partial^2 l_n(\boldsymbol{\theta}^*)}{\partial w_{q'}\partial w_q}(\hat{w}_{q'} - w_q^{(0)})$$

are on the order of $O_p(r_n)$, thus, we have

$$n\frac{\partial l_p(\boldsymbol{\theta})}{\partial w_q} = n\lambda_n\left(-O_p\left(\frac{r_n}{n\lambda_n}\right) + \frac{1}{\lambda_n}\frac{\partial P_{\lambda_n}(\hat{\boldsymbol{\theta}}_n)}{\partial w_q}\right). \tag{A.11}$$

We assume $\lambda_n \to 0$, $\frac{n\lambda_n}{r_n} \to \infty$, and that

$$\liminf_{n\to\infty}\liminf_{\theta\to 0^+}\frac{1}{\lambda_n}\frac{\partial P_{\lambda_n}(\hat{\boldsymbol{\theta}}_n)}{\partial w_q} > 0, \tag{A.12}$$

then for any $\varepsilon > 0$ and $d+1 \le q \le Q$, when $n \to \infty$, we have

$$\frac{\partial l_p(\boldsymbol{\theta})}{\partial w_q} > 0 \text{ for } 0 < \hat{w}_q < \varepsilon. \tag{A.13}$$

Therefore, $P(\hat{w}_q = 0) \to 1$ when $n \to \infty$. Then we have

$$\lim_{n\to\infty} P(\hat{\boldsymbol{w}}_{\mathcal{B}} = \mathbf{0}) = 1 \text{ if } \boldsymbol{w}^{(0)}_{\mathcal{B}} = \mathbf{0}. \tag{A.14}$$

And this concludes the model sparsity. □

A.8 Matrix algebra

If $\boldsymbol{M}$ and $\boldsymbol{N}$ are $n \times p$ and $p \times n$ matrices, respectively, and $\boldsymbol{A}$ is an $n \times n$ matrix, then

$$(\boldsymbol{A}+\boldsymbol{MN})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{M}(\boldsymbol{I}_p + \boldsymbol{NA}^{-1}\boldsymbol{M})^{-1}\boldsymbol{NA}^{-1}. \tag{A.15}$$

If we take $p = n$ and $\boldsymbol{N} = \boldsymbol{I}_n$, an $n \times n$ identity matrix, we have

$$\begin{aligned}(\boldsymbol{A}+\boldsymbol{M})^{-1} &= \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}(\boldsymbol{A}^{-1}+\boldsymbol{M}^{-1})^{-1}\boldsymbol{A}^{-1} &\text{(A.16)}\\ &= \boldsymbol{M}^{-1} - \boldsymbol{M}^{-1}(\boldsymbol{A}^{-1}+\boldsymbol{M}^{-1})^{-1}\boldsymbol{M}^{-1}. &\text{(A.17)}\end{aligned}$$

Results in (A.15) and (A.17) are useful on deriving the posterior distribution of $p(\boldsymbol{f}|\mathcal{D})$, the marginal distribution of $\boldsymbol{y}$, and other formulae throughout the book. More general results are given as follows. Let

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{pmatrix},$$

then the determinant of the matrix $\boldsymbol{A}$ is given by

$$\begin{aligned}\begin{vmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{vmatrix} &= |\boldsymbol{A}_{11}||\boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}| \\ &= |\boldsymbol{A}_{22}||\boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}|.\end{aligned}$$

Let $\boldsymbol{A}_{11} = \boldsymbol{B}$, $\boldsymbol{A}_{12} = \boldsymbol{M}$, $\boldsymbol{A}_{22} = \boldsymbol{I}$ and $\boldsymbol{A}_{21} = \boldsymbol{N}$, where $\boldsymbol{I}$ is an identity matrix, then we have

$$|\boldsymbol{B} - \boldsymbol{MN}| = |\boldsymbol{B}||\boldsymbol{I} - \boldsymbol{NB}^{-1}\boldsymbol{M}|.$$

The inverse matrix of $\boldsymbol{A}$ can be expressed as

$$\boldsymbol{A}^{-1} = \begin{pmatrix} \boldsymbol{A}_{11}^{-1} + \boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}\boldsymbol{A}_{22.1}^{-1}\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1} & -\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}\boldsymbol{A}_{22.1}^{-1} \\ -\boldsymbol{A}_{22.1}^{-1}\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1} & \boldsymbol{A}_{22.1}^{-1} \end{pmatrix},$$

where $\boldsymbol{A}_{22.1} = \boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}$, or

$$\boldsymbol{A}^{-1} = \begin{pmatrix} \boldsymbol{A}_{11.2}^{-1} & -\boldsymbol{A}_{11.2}^{-1}\boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1} \\ -\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}\boldsymbol{A}_{11.2}^{-1} & \boldsymbol{A}_{22}^{-1} + \boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}\boldsymbol{A}_{11.2}^{-1}\boldsymbol{A}_{12}\boldsymbol{A}_{12}^{-1} \end{pmatrix},$$

where $\boldsymbol{A}_{11.2} = \boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}$. From the above two equations, we can easily get the following formula:

$$\begin{aligned}\boldsymbol{A}_{11.2}^{-1} &= (\boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21})^{-1} \\ &= \boldsymbol{A}_{11}^{-1} + \boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}\boldsymbol{A}_{22.1}^{-1}\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}.\end{aligned}$$

We can use this equation to prove (A.15).

A.9 Laplace approximation

The Laplace method is a technique for approximating integrals when the integrand has a unique global maximum. Let $h(\boldsymbol{\theta})$ be a smooth function of the p-dimensional vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)^T$, and $q(\boldsymbol{\theta}, n)$ be a smooth function of n and $\boldsymbol{\theta}$. The Laplace method provides an analytical approximation to the integral of the form

$$I = \int h(\boldsymbol{\theta}) \exp\{q(\boldsymbol{\theta}, n)\} d\boldsymbol{\theta}, \tag{A.18}$$

based on the following regularity conditions:

(L1) The function $q(\boldsymbol{\theta}, n)$ is maximized at $\hat{\boldsymbol{\theta}}_n$ in the interior of the parameter space Θ.

(L2) The function $q(\boldsymbol{\theta}, n)$ is thrice differentiable.

(L3) The function $h(\boldsymbol{\theta})$ is continuously differentiable, bounded, and positive on Θ. The first-order partial derivatives of $h(\boldsymbol{\theta})$ are also bounded on Θ.

(L4) The negative of the Hessian matrix of $n^{-1} q(\boldsymbol{\theta}, n)$,

$$\boldsymbol{Q}\left(\hat{\boldsymbol{\theta}}_n, n\right) = -\frac{1}{n} \frac{\partial^2 q(\boldsymbol{\theta}, n)}{\partial \boldsymbol{\theta} \boldsymbol{\theta}^T}\bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_n},$$

is positive definite.

In addition to the regularity conditions in (L1)-(L4), there are supplementary assumptions, for example, those given in the basic description by Barndorff-Nielsen and Cox (1989). More detailed conditions and a theoretical discussion of Laplace approximations to posterior distributions can be found in Section 7.4.3 in Schervish (1995).

The Laplace method involves a Taylor series expansion of both $h(\boldsymbol{\theta})$ and $q(\boldsymbol{\theta}, n)$ about $\hat{\boldsymbol{\theta}}_n$, which gives

$$\begin{aligned} I \quad \approx \quad & \int \left[\left\{ h(\hat{\boldsymbol{\theta}}_n) + (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_n)^T \frac{\partial h(\hat{\boldsymbol{\theta}}_n)}{\partial \boldsymbol{\theta}} + \text{smaller terms} \right\} \right. \\ & \left. \exp\left\{ q\left(\hat{\boldsymbol{\theta}}_n, n\right) - \frac{n}{2} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_n)^T \boldsymbol{Q}\left(\hat{\boldsymbol{\theta}}_n, n\right) (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_n)^T + \text{smaller terms} \right\} \right] d\boldsymbol{\theta}. \end{aligned}$$

Note that $\exp\left\{ -\frac{n}{2} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_n)^T \boldsymbol{Q}\left(\hat{\boldsymbol{\theta}}_n, n\right) (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_n)^T \right\}$ is the multivariate normal density with mean $\hat{\boldsymbol{\theta}}_n$ and variance matrix $n^{-1} \boldsymbol{Q}(\hat{\boldsymbol{\theta}}_n, n)^{-1}$. Therefore under the regularity conditions, the Laplace approximation to the integral in (A.18) is obtained as follows:

$$I \approx \exp\left\{ q(\hat{\boldsymbol{\theta}}_n, n) \right\} h(\hat{\boldsymbol{\theta}}_n) \frac{(2\pi)^{\frac{p}{2}}}{n^{\frac{p}{2}} \left| \boldsymbol{Q}\left(\hat{\boldsymbol{\theta}}_n, n\right) \right|^{1/2}} \times (1 + o(1)). \tag{A.19}$$

In applications, $q(\boldsymbol{\theta}, n)$ may be the log-likelihood function or the logarithm of the unnormalized posterior density, whereas $\hat{\boldsymbol{\theta}}_n$ may be the MLE or the posterior mode.

The interested reader should refer to the monograph by Small (2010) for further details about the general exposition on asymptotic expansions and their statistical applications.

Bibliography

Abrahamsen, P. (1997). A review of Gaussian random fields and correlation functions. Technical Report 917. Norwegian Computing Center, Osla.

Adams, R. P. and Stegle, O. (2008). Gaussian process product models for nonparametric nonstationarity. In *25th International Conference on Machine Learning*.

Adler, R. J. (1990). *An Introduction to Continuity, Extrema, and Related Topics for Gaussian Processes*, volume 12 of *IMS Lecture notes-monograph series*. Hayward, CA.

Adler, R. J. (1981). *The geometry of random fields*. Wiley, New York.

Adler, R. J. and Taylor, J. E. (2007). *Random Fields and Geometry*. Springer, New York.

Amewou-Atisso, M., Ghosal, S., Ghosh, J. K., and Ramamoorthi, R. V. (2003). Posterior consistency for semi-parametric regression problems. *Bernoulli*, 9(2):291–312.

Attias, H. (1999). Inferring parameters and structure of latent variable models by variational Bayes. In Prade, H. and Laskey, K., editors, *Proc. 15th Conference on Uncertainty in Artificial Intelligence*, pages 21–30, Morgan Kaufmann Publishers, Stockholm, Sweden.

Attias, H. (2000). A variational Bayesian framework for graphical models. In Solla, S., Leen, T., and Muller, K.-R., editors, *Advances in neural information processing systems*, volume 12, pages 209–215, MIT Press, Cambridge, MA.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). Gaussian predictive process models for large spatial data sets. *J. R. Statist. Soc. B*, 70:825–848.

Banks, D. L., Olszewski, R. T., and Maxion, R. A. (1999). Comparing methods for multivariate nonparametric regression. Technical report CMU-CS-99-102. School of Computer Science, Carnegie Mellon University.

Barndorff-Nielsen, O. E. and Cox, D. R. (1989). Asymptotic techniques for use in statistics. In *Monographs on statistics and applied probability*. Chapman & Hall, London.

Barron, A., Schervish, J. M., and Wasserman, L. (1999). The consistency of posterior distributions in nonparametric problems. *Ann. Statist.*, 27:536–561.

Barron, A. R. (1999). Information-theoretic characterization of Bayes performance and the choice of priors in parametric and nonparametric problems. In *Bayesian Statistics, 6*, pages 27–52, Oxford University Press, New York.

Bartholomew, D. J. and Knott, M. (1999). *Latent Variable Models and Factor Analysis*, volume 7 of *Kendall's Library of Statistics*, second edition. Edward Arnold, London.

Basawa, I. V. and Prakasa Rao, B. L. S. (1980). *Statistical Inference for Stochastic Processes*. Academic Press, London.

Billingsley, P. (1995). *Probability and Measure*, third edition, Wiley, New York.

Bishop, C. M. and James, G. D. (1993). Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. *Nuclear Instrum. Methods Phys. Res.*, A327:580–593.

Boyle, P. and Frean, M. (2004). Multiple-output Gaussian process regression. Technical report, Victoria University of Wellington.

Boyle, P. and Frean, M. (2005). Dependent Gaussian processes. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in neural information processing systems*, volume 17, pages 217–224, MIT Press, Cambridge, MA.

Breiman, L. (1968). *Probability*. Addison-Wesley, MA.

Breiman, L. and Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation. *J. Am. Statist. Assoc.*, 80:580–598.

Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *J. Am. Statist. Assoc.*, 88:9–25.

Brown, T. A. (2006). *Confirmatory Factor Analysis for Applied Research*. Guilford, New York.

Burden, F. R. (2000). Use of automatic relevance determination in qsar studies using Bayesian neural networks. *J. Chem. Inf. Comput. Sci.*, 40:1423–1430.

Cai, T. and Hall, P. (2006). Prediction in functional linear regression. *Ann. Statist.*, 34:2159–2179.

Candes, E. J. and Plan, Y. (2007). Near-ideal model selection by l_1 minimization. *Ann. Statist.*, 37:2145–2177.

Candes, E. J. and Tao, T. (2007). The Dantzig selector: statistical estimation

when p is much larger than n (with discussion). *Ann. Statist.*, 35:2313–2404.

Cappé, O., Robert, C. P., and Rydén, T. (2003). Reversible jump, birth-and-death and more general continuous time Markov chain Monte Carlo samplers. *J. R. Statist. Soc. B*, 65:679–700.

Cardot, H., Crambes, C., Kneip, A., and Sarda, P. (2006). Smoothing splines estimators in functional linear regression with errors-in-variables. *Comput. Statist. Data Anal.*, 51:4832–4848.

Cardot, H., Ferraty, F., and Sarda, P. (1999). Functional linear model. *Statist. Probab. Lett.*, 45:11–22.

Carlin, B. P. and Louis, T. A. (1996). *Bayes and empirical Bayes methods for data analysis*. Chapman & Hall/CRC, London.

Castillo, I. (2008). Lower bounds for posterior rates with Gaussian process priors. *Electron. J. Stat.*, 2:1281–1299.

Chen, F. and Khalil, H. (1995). Adaptive control of a class of nonlinear discrete-time systems. *IEEE. Trans. Automatic Control*, 40:791–801.

Chen, J. and Li, P. (2009). Hypothesis test for normal mixture models: the EM approach. *Ann. Statist.*, 37:2523–2542.

Chen, M.-H., Shao, Q.-M., and Ibrahim, J. G. (2000). *Monte Carlo Methods in Bayesian Computation*. Springer Series in Statistics. Springer-Verlag, New York.

Chen, T. and Martin, E. (2009). Bayesian linear regression and variable selection for spectroscopic calibration. *Anal. Chim. Acta*, 631:13–21.

Cheng, B. and Titterington, D. M. (1994). Neural networks: a review from a statistical perspective (with discussion). *Statist. Sci.*, 9:2–54.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *J. Am. Statist. Assoc.*, 90(432):1313–1321.

Choi, T. (2005). *Posterior consistency in nonparametric regression problems under Gaussian process priors*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.

Choi, T. (2007). Alternative posterior consistency results in nonparametric binary regression using Gaussian process priors. *J. Statist. Plann. Inference*, 137:2975–2983.

Choi, T. and Ramamoorthi, R. V. (2008). Remarks on consistency of posterior distributions. In *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*, volume 3 of *Inst. Math. Stat. Collect.*, pages 170–186, Institute of Mathematics Statistics, Beachwood, OH.

Choi, T. and Schervish, M. J. (2007). On posterior consistency in nonparamet-

ric regression problems. *J. Multivariate Anal.*, 98:1969–1987.

Choi, T., Shi, J. Q., and Wang, B. (2011). A Gaussian process regression approach to a Bayesian single-index model. *J. Nonparametric Statist.*, 23:21–36.

Choi, T., Shi, J. Q., and Wang, B. (2010). Gaussian process partially linear regression model. Technical report, Newcastle University.

Choudhuri, N., Ghosal, S., and Roy, A. (2004). Bayesian estimation of the spectral density of time series. *J. Am. Statist. Assoc.*, 99:1050–1059.

Choudhuri, N., Ghosal, S., and Roy, A. (2007). Nonparametric binary regression using a Gaussian process prior. *Statist. Methodol.*, 4:227–243.

Chu, W. and Ghahramani, Z. (2005). Gaussian processes for ordinal regression. *J. Machine Learning Res.*, 6:1019–1041.

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory*, second edition. Wiley-Interscience (John Wiley & Sons), Hoboken, NJ.

Cramer, H. and Leadbetter, M. R. (1967). *Stationary and Related Stochastic Processes*. Wiley, New York.

Cressie, N. and Johannesson, G. (2008). Fixed rank kriging for very large spatial data sets. *J. R. Statist. Soc. B*, 70:209–226.

Cucker, F. and Smale, S. (2001). On the mathematical functions of learning. *Bull. Am. Math. Soc.*, 39:1–49.

Daley, R. (1991). *Atmospheric data analysis*. Cambridge University Press, Cambridge, UK.

Delaigle, A., Hall, P., and Apanasovich, T. V. (2009). Weighted least squares methods for prediction in the functional data linear model. *Electron. J. Statist.*, 3:865–885.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:1–38.

Denison, D. G. T., Mallick, B. K., and Smith, A. F. M. (1998). Automatic Bayesian curve fitting. *J. R. Statist. Soc. B*, 60:333–350.

Diaconis, P. and Freedman, D. A. (1986). On the consistency of Bayes estimates. *Ann. Statist.*, 14:1–26.

Diggle, P. J., Ribeiro, P. J., and Christensen, O. F. (2003). An introduction to model based geostatistics. In Møller, J., editor, *Lecture Notes in Statistics*, volume 173, Springer-Verlag, New York.

Diggle, P. J., Tawn, J. A., and Moyeed, R. A. (1998). Model based geostatistics (with discussion). *Appl. Statist.*, 47:299–350.

Donoho, D. L. and Johnstone, I. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81:425–455.

Doob, J. L. (1949). Application of the theory of martingales. *Coll. Int. du C. N. R. S. Paris*, pages 23–27.

Duan, N. and Li, K. C. (1991). Slicing regression: a link-free regression method. *Ann. Statist.*, 19:505–530.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Phys. Lett. B*, 195:216–222.

Eggermont, P. and LaRiccia, V. N. (2009). *Maximum penalized likelihood estimation volume II:regression*. Springer Series in Statistics. Springer, New York.

Fabri, S. and Kadirkamanathan, V. (1998). Dual adaptive control of stochastic systems using neural networks. *Automatica*, 14:245–253.

Fahrmeir, L. and Lang, S. (2001). Bayesian inference for generalized additive mixed models based on Markov random field priors. *Appl. Statist.*, 50:201–220.

Fahrmeir, L. and Tutz, G. (2001). *Multivariate statistical modelling based on generalized linear models*, second edition. Springer, Berlin.

Fan, J. and Gijbels, I. (1996). *Local polynomial modelling and its applications*. Chapman & Hall, London.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Statist. Assoc.*, 96:1348–1360.

Fan, J. and Lv, J. (2003). Sure independence screening for ultra high dimensional feature space (with discussion). *J. R. Statist. Soc. B*, 70:849–911.

Fan, J., Yao, Q., and Cai, Z. (2003). Adaptive varying-coefficient linear models. *J. R. Statist. Soc. B*, 65:57–80.

Fan, J. and Zhang, J.-T. (2000). Two-step estimation of functional linear models with applications to longitudinal data. *J. R. Statist. Soc. B*, 62:303–322.

Fan, J. and Zhang, W. (1999). Statistical estimation in varying coefficient models. *Ann. Statist.*, 27:1491–1518.

Faraway, J. (1997). Regression analysis for a functional response. *Technometrics*, 39:254–261.

Faraway, J. (2001). Modelling hand trajectories during reaching motions. Technical Report #383. Department of Statistics, University of Michigan.

Fernandez, C. and Green, P. (2002). Modelling spatially correlated data via mixtures: a Bayesian approach. *J. R. Statist. Soc. B*, 64:805–826.

Ferraty, F. and Vieu, P. (2006). *Nonparametric functional data analysis*. Springer, New York.

Fraley, C. and Raftery, A. E. (2006). Mclust version 3 for R: normal mixture

modelling and model-based clustering. Technical Report #504. Department of Statistics, University of Washington.

Frank, I. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools (with discussion). *Technometrics*, 35:109–148.

Freedman, D. A. (1963). On the asymptotic behavior of Bayes' estimates in the discrete case. *Ann. Math. Statist.*, 34:1386–1403.

Freedman, D. A. (1965). On the asymptotic behavior of Bayes' estimates in the discrete case ii. *Ann. Math. Statist.*, 36:454–456.

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. *J. Am. Statist. Assoc.*, 76:817–823.

Fu, W. J. (1998). Penalized regressions: the bridge versus the lasso. *J. Comput. Graph. Statist.*, 7:397–416.

Gamerman, D. and Lopes, H. F. (2006). *Markov chain Monte Carlo-stochastic simulation for Bayesian inference, second edition.* Chapman & Hall/CRC, New York.

Ge, Y. and Jiang, W. (2006). On consistency of Bayesian inference with mixtures of logistic regression. *Neural Comput.*, 18(1):224–243.

Geisser, S. (1975). The predictive sample reuse method with applications. *J. Am. Statist. Assoc.*, 70:320–328.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:721–741.

Ghosal, S., Ghosh, J. K., and Ramamoorthi, R. V. (1999). Posterior consistency of Dirichlet mixtures in density estimation. *Ann. Statist.*, 27:143–158.

Ghosal, S., Ghosh, J. K., and van der Vaart, A. W. (2000). Convergence rates of posterior distributions. *Ann. Statist.*, 28(2):500–531.

Ghosal, S. and Roy, A. (2006). Posterior consistency of Gaussian process prior for nonparametric binary regression. *Ann. Statist.*, 34:2413–2429.

Ghosal, S. and van der Vaart (2007). Convergence rates of posterior distributions for noniid observations. *Ann. Statist.*, 35: 192–223.

Ghosh, J. K., Delampady, M., and Samanta, T. (2006). *An Introduction to Bayesian Analysis theory and methods.* Springer Texts in Statistics. Springer, New York.

Ghosh, J. K. and Ramamoorthi, R. V. (2003). *Bayesian nonparametrics*. Sprinber-Verlag, New York.

Gibbs, M. N. and MacKay, D. J. C. (1997). Efficient implementation of Gaussian processes. Technical report, Department of Physics, Cavendish Laboratory, Cambridge University.

Gilks, W. R. and Wild, P. (1997). Adaptive rejection sampling for Gibbs sam-

pling. *Appl. Statist.*, 41:337–348.

Girard, A. and Murray-Smith, R. (2005). Gaussian processes: prediction at a noisy input and application to iterative multiple-step ahead forecasting of time-series. In Murray-Smith, R. and Shorten, R., editors, *Switching and learning in feedback systems, Lecture Notes in Computing Science*, volume 3355, pages 158–184, Springer-Verlag, New York.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537.

Görür, D. and Rasmussen, C. E. (2010). Dirichlet process Gaussian mixture models: choice of the base distribution. *J. Comput. Sci. Tech.*, 25:653–664.

Grancharova, A., Kocijan, J., and Johansen, T. A. (2008). Explicit stochastic predictive control of combustion plants based on Gaussian process models. *Automatica*, 44:1621–1631.

Green, P. and Silverman, B. (1994). *Nonparametric regression and generalized linear models: a roughness penalty approach*. Chapman & Hall, New York.

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation and some robust and resistant alternatives. *J. R. Statist. Soc. B*, 46:149–192.

Green, P. J. (1995). Reversible jump MCMC computation and Bayesian model determination. *Biometrika*, 82:711–732.

Green, P. J. and Richardson, S. (2000). Spatially correlated allocation models for count data. Technical report. University of Bristol.

Grünwald, P. D. (2007). *The minimum description length principle*. MIT Press, Cambridge, MA.

Gu, C. (2002). *Smoothing splines ANOVA models*. Springer-Verlag, New York.

Hall, P., Müller, H.-G., and Yao, F. (2008). Modelling sparse generalized longitudinal observations with latent Gaussian processes. *J. R. Statist. Soc. B*, 70:703–723.

Härdle, W. and Stoker, T. M. (1989). Investigating smooth multiple regression by the method of average derivatives. *J. Am. Statist. Assoc.*, 84:986–995.

Hastie, T. and Tibshirani, R. J. (1990). *Generalized additive model*. Chapman & Hall, London.

Hastie, T. and Tibshirani, R. J. (1993). Varying coefficient models. *J. R. Statist. Soc. B*, 55:757–796.

Higdon, D. (2002). Space and space-time modelling using process convolutions. In Anderson, C., Barnett, V., Chatwin, P., and El-Shaarawi, A., editors, *Quantitative Methods for Current Environmental Issues*, pages 37–56. Springer Verlag, New York.

Higdon, D., Swall, J., and Kern, J. (1999). Non-stationary spatial modeling. In Bernardo, J. M., Berger, J., Dawid, A., and Smith, A., editors, *Bayesian Statistics*, volume 6, Oxford University Press, Oxford, U.K.

Hinton, G. E. and van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *Proc. 6th Annu. Conf. Computational Learning Theory*, pages 5–13. ACM Press: New York.

Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G. (2010). *Bayesian nonparametrics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge.

Horowitz, A. (1991). A generalized guided Monte Carlo algorithm. *Phys. Lett. B*, 268:247–252.

Huber, P. J. (1985). Projection pursuit (with discussion). *Ann. Statist.*, 13:435–525.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *J. Classif.*, 2:193–218.

Jain, S. and Neal, R. M. (2004). A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *J. Comput. Graphical Statist.*, 13:158–182.

James, G. and Sugar, C. (2003). Clustering for sparsely sampled functional data. *J. Am. Statist. Assoc.*, 98:397–408.

Jiang, W. (2006). On the consistency of Bayesian variable selection for high dimensional binary regression and classification. *Neural Comput.*, 18(11):2762–2776.

Jordan, M. I. (2004). Graphical models. *Statist. Sci.*, 19:140–155.

Jöreskog, K. G. and Sörbom, D. (1999). *LISREL 8 users reference guide*. Scientific Software International, Lincolnwood, IL.

Journel, A. G. and Huijbregts, C. J. (1978). *Mining geostatistics*. Academic Press, London.

Kakade, S., Seeger, M., and Foster, P. (2006). Worst-case bounds for Gaussian process models. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in neural information processing systems*, volume 18, MIT Press, Cambridge, MA.

Kamnik, R., Bajd, T., and Kralj, A. (1999). Functional electrical stimulation and arm supported sit-to-stand transfer after paraplegia: a study of kinetic parameters. *Artificial Organs*, 23:413–417.

Kamnik, R., Shi, J. Q., Murray-Smith, R., and Bajd, T. (2005). Nonlinear modelling of FES-supported standing up in paraplegia for selection of feedback sensors. *IEEE Transact. Neural Systems Rehab. Eng.*, 13:40–52.

Kass, R. and Raftery, A. (1995). Bayes factors. *J. Am. Statist. Assoc.*, 90:773–795.

Kaufman, C. G. and Sain, S. R. (2010). Bayesian functional ANOVA modelling using Gaussian process prior distributions. *Bayesian Analysis*, 5:123–150.

Kimeldorf, G. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. *J. Math. Analysis and Appl.*, 33:82–95.

Ko, J. and Fox, D. (2009). Learning GP-Bayes filters via Gaussian process latent variable models. *Robotics: Science and Systems*, Seattle, WA.

Kocijan, J. and Murray-Smith, R. (2005). Non-linear predictive control with a Gaussian process model. In Shorten, R. and Murray-Smith, R., editors, *Switching and learning in feedback systems, Lecture Notes in Computer Science*, volume 3355, pages 609–616, Springer-Verlag, Heidelberg, Germany.

Kuss, M. and Rasmussen, C. E. (2005). Assessing approximate inference for binary Gaussian process classification. *J. Machine Learning Res.*, 6:1679–1704.

Larson, S. (1931). The shrinkage of the coefficient of multiple correlation. *J. Educat. Psychol.*, 22:45–55.

Laslett, G. M. (1994). Kriging and splines: an empirical comparison of their predictive performance in some applications. *J. Am. Statist. Assoc.*, 89:391–409.

Lawrence, N. D. (2004). Gaussian process latent variable models for visualization of high dimensional data. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in neural information processing systems*, volume 16, pages 329–336, MIT Press, Cambridge, MA.

Lawrence, N. D. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *J. Machine Learning Res.*, 6:1783–1816.

Lawrence, N. D. and Jordan, M. I. (2005). Semi-supervised learning via Gaussian processes. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in neural information processing systems*, volume 17, pages 753–760, MIT Press, Cambridge, MA.

Lawrence, N. D., Seeger, M., and Herbrich, R. (2003). Fast sparse Gaussian process methods: the informative vector machine. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in neural information process-*

ing systems, volume 15, pages 609–616, MIT Press, Cambridge, MA.

Le Cam, L. M. (1986). *Asymptotic methods in statistical decision theory*. Springer, New York.

Lee, S. Y. (2007). *Structural equation modelling: a Bayesian approach*. John Wiley & Sons, London.

Leng, C., Zhang, W., and Pan, J. (2009). Semiparametric mean-covariance regression analysis for longitudinal data. *J. Am. Statist. Assoc.*, 105:181–193.

Li, K. (1991). Sliced inverse regression for dimension reduction. *J. Am. Statist. Assoc.*, 86:316–342.

Lindley, D. V. and Smith, A. F. M. (1972). Bayes estimates for the linear model. *J. Roy. Statist. Soc. Ser. B*, 34:1–41.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer-Verlag, New York.

MacKay, D. J. C. (1995). Ensemble learning and evidence maximization. *Proc. NIPS Conf.*

MacKay, D. J. C. (1998). Introduction to Gaussian processes. In Bishop, C. M., editor, *Neural networks and machine learning*, volume 168 of NATO ASI Series, pages 133–165, Springer, Berlin.

MacKay, D. J. C. (1998b). Introduction to Gaussian processes. In Bishop, C. M., editor, *Neural networks and machine learning*, volume 168 of NATO ASI Series, pages 133–165, Springer, Berlin.

Martin, A. D. and Quinn, K. M. (2007). MCMCpack: Markov chain Monte Carlo (MCMC) package. Technical report, Washington University in St. Louis. R package version 0.8-1, URL http://mcmcpack.wustl.edu.

Martin, A. D., Quinn, K. M., and Park, J. H. (2010). MCMCpack: Markov chain Monte Carlo in R. *J. Statist. Software*. http://www.law.berkeley.edu/files/jstatsoftMCMCpack.pdf.

Matheron, G. (1973). The intrinsic random functions and their applications. *Adv. Appl. Probability*, 5:439–468.

McCullagh, P. and Nelder, J. (2000). *Generalized linear models*, second edition. Chapman & Hall/CRC, Boca Raton, FL.

McLachlan, G. J., Do, K. A., and Ambroise, C. (2004). *Analyzing micro array gene expression data*. Wiley, New York.

McLachlan, G. J. and Peel, D. (2000). *Finite mixture models*. Wiley, New York.

Meng, X.-L. and Wong, W. H. (1996). Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statist. Sinica*, 6(4):831–860.

Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. *Uncertnty Artif. Intell.*, 17:362–369.

Mosteller, F. and Turkey, J. (1968). Data analysis, including statistics. In *Handbook of social psychology*, Addison-Wesley, Reading, MA.

Mosteller, F. and Wallace, D. L. (1963). Inference in an authorship problem. *J. Am. Statist. Assoc.*, 58:275–309.

Müller, H. (2005). Functional modelling and classification of longitudinal data. *Scand. J. Statist.*, 32:223–240.

Murphy, S. (2003). Optimal dynamic treatment regimes (with discussion). *J. R. Statist. Soc. B*, 65:331–366.

Murray-Smith, R., Sbarbaro, D., Rasmussen, C. E., and Girard, A. (2003). Adaptive, cautious, predictive control with Gaussian process priors. In *Proceedings of 13-th IFAC Symposium on System Identification*. Rotterdam.

Narendra, K. and Parthasarathy, P. (1990). Identification and control of dynamical systems using neural networks. *IEEE. Trans. Neural Networks*, 1:4–27.

Neal, R. (1999). Regression and classification using Gaussian process priors (with discussion). In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 6*, pages 475–501. Oxford University Press, Oxford.

Neal, R. M. (1996). *Bayesian learning for neural networks*. Springer-Verlag, New York.

Neal, R. M. (1997). Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical report, Department of Statistics and Department of Computer Science, University of Toronto. http://www.cs.utoronto.ca/∼radford/.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *J. Comput. Graph. Statist.*, 9:249–265.

Nickisch, H. and Rasmussen, C. E. (2008). Approximations for binary Gaussian process classification. *J. Machine Learning Res.*, 9:2035–2078.

O'Hagan, A. (1978). Curve fitting and optimal design for prediction (with discussion). *J. R. Statist. Soc. B*, 40:1–42.

Opper, C. M. and Vivarelli, F. (1999). General bounds on Bayes errors for regression with Gaussian processes. In Kearns, M., Solla, S. A., and Cohn, D., editors, *Advances in neural information processing systems*, volume 11, page 302–308, MIT Press, Cambridge, MA.

Paciorek, C. J. and Schervish, M. J. (2004). Nonstationary covariance functions for Gaussian process regression. In Touretzky, D. S., Mozer, C. M.,

and Hasselmo, E. M., editors, *Advances in neural information processing systems*, volume 16, MIT Press, Cambridge, MA.

Park, T. and Casella, G. (2008). The Bayesian lasso. *J. Am. Statist. Assoc.*, 103:681–686.

Petrone, S., Guindani, M., and Gelfand, A. E. (2009). Hybrid Dirichlet mixture models for functional data. *J. Roy. Statist. Soc. Ser. B*, 71:755–782.

Pillai, N. S., Wu, Q., Liang, F., Mukherjee, S., and Wolpert, R. L. (2007). Characterizing the function space for Bayesian kernel models. *J. Mach. Learn. Res.*, 8:1769–1797 (electronic).

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

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical recipes*. Cambridge University Press, Cambridge.

Pronzato, L. (2008). Optimal experimental design and some related control problems. *Automatica*, 44:303–325.

Ramsay, J. and Dalzell, C. J. (1991). Some tools for functional data analysis (with discussion). *J. R. Statist. Soc. B*, 53:539–572.

Ramsay, J. O. (1998). Principal differential analysis: Data reduction by differential operators. *J. R. Statist. Soc. B*, 58:495–508.

Ramsay, J. O., Hooker, G., and Graves, S. (2009). *Functional data analysis with R and MATLAB*. Springer, New York.

Ramsay, J. O. and Li, X. (1998). Curve registration. *J. R. Statist. Soc. B*, 60:351–363.

Ramsay, J. O. and Silverman, B. W. (2002). *Applied functional data analysis: methods and case studies*. Springer, New York.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional data analysis*. second edition. Springer, New York.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *J. Am. Statist. Assoc.*, 66:846–850.

Rasmussen, C. E. (1996). Evaluation of Gaussian processes and other methods for non-linear regression. Ph.D. thesis, University of Toronto, Toronto, Canada.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press, Cambridge, MA.

Ravishanker, N. and Dey, D. K. (2002). *A first course in linear model theory*. Chapman & Hall/CRC, New York.

Rice, J. A. and Silverman, B. W. (1991). Estimating the mean and covariance nonparametrically when the data are curves. *J. R. Statist. Soc. B*, 53:233–

243.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *J. R. Statist. Soc. B*, 59:731–792.

Robert, C. P. and Casella, G. (2004). *Monte Carlo statistical methods*, second edition. Springer Texts in Statistics. Springer-Verlag, New York.

Rosthoj, S., Fullwood, C., Henderson, R., and Stewart, S. (2006). Estimation of optimal dynamic anticoagulation regimes from observational data: a regret-based approach. *Statistics in Medicine*, 25:4197–4215.

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications*. Chapman & Hall/CRC, London.

Rue, H. and Martino, S. (2007). Approximate Bayesian inference for hierarchical Gaussian Markov random field models. *J. Statist. Plann. Inference*, 137(10):3177–3192.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models using integrated nested Laplace approximations (with discussion). *J. R. Statist. Soc. B*, 71:319–392.

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003). *Semiparametric regression*. Cambridge University Press, London.

Sampson, P. D. and Guttorp, P. (1992). Nonparametric estimation of nonstationary spatial covariance structure. *J. Am. Statist. Assoc.*, 87:108–119.

Sbarbaro, D. and Murray-Smith, R. (2005). Self-tuning control of nonlinear systems using Gaussian process prior model. In Murray-Smith, R. and Shorten, R., editors, *Switching and learning in feedback systems, Lecture Notes in Computer Science*, volume 3355, pages 140–157, Springer, Heidelberg.

Sbarbaro, D., Murray-Smith, R., and Valdes, A. (2004). Multivariable generalized minimum variance control based on artificial neural networks and Gaussian process models. In *International Symposium on Neural Networks*. Springer Verlag, New York.

Schervish, M. J. (1995). *Theory of statistics*. Springer Series in Statistics. Springer-Verlag, New York.

Schmidt, A. M. and O'Hagan, A. (2003). Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *J. R. Statist. Soc. B*, 65:745–758.

Schölkopf, B. and Smola, A. J. (2002). *Learning with kernels*. MIT Press, Cambridge, MA.

Schölkopf, B., Smola, A. J., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.,* 10:1299–

1319.

Schwartz, L. (1965). On Bayes procedures. *Z. Wahr. Verw. Gebiete*, 4:10–26.

Schwarz, G. E. (1978). Estimating the dimension of a model. *Ann. Statist.*, 6:461–464.

Seeger, M. (2004). Bayesian Gaussian processes for machine learning. *Int. J. Neural Sys.*, 14.

Seeger, M., Williams, C. K. I., and Lawrence, N. (2003). Fast forward selection to speed up sparse Gaussian process regression. In Bishop, C. and Frey, B. J., editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics Society for Artificial Intelligence and Statistics*. Key West, FL.

Seeger, M. W., Kakade, S. M., and Foster, D. P. (2008). Information consistency of nonparametric Gaussian process methods. *IEEE Trans. Inform. Theory*, 54(5):2376–2382.

Serradilla, J. and Shi, J. Q. (2010). Gaussian process factor analysis analysis model with application in stochastic monitoring. Technical report, School of Maths. & Stats., Newcastle University, UK.

Shen, X. and Wasserman, L. (2001). Rates of convergence of posterior distributions. *Ann. Statist.*, 29:666–686.

Shi, J. Q. and Li, J. L. (2010). Optimal dynamic control in dose-response curve prediction. Technical report, School of Maths. & Stats., Newcastle University, UK.

Shi, J. Q., Murray-Smith, R., and Titterington, D. M. (2003). Bayesian regression and classification using mixtures of Gaussian process. *Int. J. Adaptive Control Signal Process.*, 17:149–161.

Shi, J. Q., Murray-Smith, R., and Titterington, D. M. (2005a). Hierarchical Gaussian process mixtures for regression. *Stat. Comput.*, 15:31–41.

Shi, J. Q., Murray-Smith, R., Titterington, D. M., and Pearlmutter, B. A. (2005b). Learning with large data-sets using a filtering approach. In Murray-Smith, R. and Shorten, R., editors, *Switching and learning in feedback systems*, volume 23, pages 128–139. Springer-Verlag, New York.

Shi, J. Q. and Wang, B. (2008). Curve prediction and clustering with mixtures of Gaussian process functional regression models. *Statist. Comput.*, 18:267–283.

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

Shi, J. Q., Wang, B., Will, E. J., and West, R. M. (2010). Dose-response curve

prediction using GPFR models with applications to the control of renal anaemia. Technical report, School of Maths. & Stats., Newcastle University, UK.

Silverman, B. W. (1984). Spline smoothing: the equivalent variable kernel method. *Ann. Statist.*, 12: 898–916.

Skilling, J. (2006). Nested sampling for general Bayesian computation. *Bayesian Anal.*, 1:833–860.

Small, C. G. (2010). *Expansions and asymptotics for statistics*. Monographs on Statistics and Applied Probability. Chapman & Hall, New York.

Sollich, P. and Williams, K. I. (2005). Understanding Gaussian process regression using the equivalent kernel. In *Deterministic and statistical methods in machine learning*, volume 3635 of *Lecture Notes in Computer Science*, pages 211–228. Springer-Verlag, Berline/Heidelberg.

Stein, M. L. (1999). *Interpolation of spatial data: some theory for kriging*. Springer, New York.

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components—an alternative to reversible jump methods. *Ann. Statist.*, pages 40–74.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *J. R. Statist. Soc. B*, 36:111–147.

Sundararajan, S. and Keerthi, S. S. (2001). Predictive approaches for choosing hyperparameters in Gaussian processes. *Neural Comput.*, 13:1103–1118.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, 58:267–288.

Tierney, L. and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *J. Am. Statist. Assoc.*, 81(393):82–86.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1(3):211–244.

Titterington, D. M. (2004). Bayesian methods for neural networks and related models. *Statist. Sci.*, 19:128–139.

Titterington, D. M., Smith, A., and Makov, U. (1985). *Statistical analysis of finite mixture distributions*. Wiley, New York.

Tokdar, S. T. and Ghosh, J. K. (2007). Posterior consistency of Gaussian process priors in density estimation. *J. Statist. Plann. Inference*, 137:34–42.

Urtasun, R. and Darrell, T. (2007). Discriminative Gaussian process latent variable model for classification. In *24th International Conference on Machine Learning*, pages 927–934, ACM Press: New York.

van de Geer, S. (2008). High-dimensional generalized linear models and the lasso. *Ann. Statist.*, 36:614–645.

van der Vaart, A. and van Zanten, H. (2010). Information rates of nonparametric Gaussian process methods. Preprint.

van der Vaart, A. W. and van Zanten, J. H. (2008a). Rates of contraction of posterior distributions based on Gaussian process priors. *Ann. Statist.*, 36(3):1435–1463.

van der Vaart, A. W. and van Zanten, J. H. (2008b). Reproducing kernel Hilbert spaces of Gaussian priors. In *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*, volume 3 of *Inst. Math. Stat. Collect.*, pages 200–222. Inst. Math. Statist., Beachwood, OH.

van der Vaart, A. W. and Wellner, J. A. (1996). *Weak convergence and empirical processes*. Springer Series in Statistics. Springer-Verlag, New York.

Volkova, N. and Arab, L. (2006). Evidence-based systematic literature review of hemoglobin/ hematocrit and all-cause mortality in dialysis patients. *Am. J. Kidney Disease*, 47:24–36.

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

Walker, S. (2004). New approaches to Bayesian consistency. *Ann. Statist.*, 32(5):2028–2043.

Wand, M. P. and Jones, M. C. (1995). *Kernel smoothing*. Chapman & Hall, New York.

Wang, B. and Shi, J. Q. (2011). Generalised Gaussian process regression model for non-Gaussian functional data. Technical report, Newcastle University, UK.

Wang, B. and Titterington, D. M. (2006). Convergence properties of a general algorithm for calculating variational Bayesian estimates for a normal mixture model. *Bayes. Anal.*, 1:625–650.

Wang, S., Nan, B., Zhou, N., and Zhu, J. (2009). Hierarchically penalized Cox regression with grouped variables. *Biometrika*, 96:307–322.

West, R. M., Harris, K., Gilthorpe, M. S., Tolman, C., and Will, E. J. (2007). A description of patient sensitivity to epoetins and control of renal anaemia—functional data analysis applied to a randomized controlled clinical trial in hemodialysis patients. *J. Am. Soc. Nephrol.*, 18:237–2376.

Widom, H. (1963). Asymptotic behavior of the eigenvalues of certain integral equations. *Trans. Am. Math. Soc.*, 109:278–295.

Will, E. J., Richardson, D., Tolman, C., and Bartlett, C. (2007). Development and exploitation of a clinical decision support system for the management of renal anaemia. *Nephrol. Dialysis Transplant.*, 22, (Suppl. 4).

Williams, C. K. I. (1998). Prediction with Gaussian processes: from linear re-

gression to linear prediction and beyond. In Jordan, M. I., editor, *Learning and inference in graphical models*, pages 599–621, MIT Press, Cambridge, MA.

Williams, C. K. I. and Rasmussen, C. E. (1996). Gaussian processes for regression. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in neural information processing systems*, volume 8, pages 514–520, MIT Press, Cambridge, MA.

Williams, C. K. I., Rasmussen, C. E., Schwaighofer, A., and Tresp., V. (2002). Observations on the Nyström method for Gaussian process prediction. Technical report, University of Edinburgh.

Williams, C. K. I. and Seeger, M. (2001). Using the Nyström method to speed up kernel machines. In Leen, T. K., Diettrich, T. G., and Tresp, V., editors, *Advances in neural information processing systems, 13*, pages 682–688, MIT Press, Cambridge, MA.

Wu, Y. and Ghosal, S. (2008). Posterior consistency for some semi-parametric problems. *Sankhyā*, 70(2, Ser. A):267–313.

Yao, F., Müller, H. G., and Wang, J. L. (2005a). Functional data analysis for sparse longitudinal data. *J. Am. Statist. Assoc.*, 100:577–590.

Yao, F., Müller, H. G., and Wang, J. L. (2005b). Functional linear regression analysis for longitudinal data. *Ann. Statist.*, 33:2873–2903.

Yi, G. (2009). Variable selection with penalized Gaussian process regression models. Ph.D. thesis, Newcastle University.

Yi, G., Shi, J. Q., and Choi, T. (2011). Penalized Gaussian process regression and classification for high-dimensional nonlinear data. *Biometrics*, in press.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Statist. Soc. B*, 68:49–68.

Zou, H. (2006). The adaptive lasso and its oracle properties. *J. Am. Statist. Soc.*, 101:1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Statist. Soc. B*, 67:301–320.

This page intentionally left blank

Statistics

Gaussian Process Regression Analysis for Functional Data presents nonparametric statistical methods for functional regression analysis, specifically the methods based on a Gaussian process prior in a functional space. The authors focus on problems involving functional response variables and mixed covariates of functional and scalar variables.

Covering the basics of Gaussian process regression, the first several chapters discuss functional data analysis, theoretical aspects based on the asymptotic properties of Gaussian process regression models, and new methodological developments for high dimensional data and variable selection. The remainder of the text explores advanced topics of functional regression analysis, including novel nonparametric statistical methods for curve prediction, curve clustering, functional ANOVA, and functional regression analysis of batch data, repeated curves, and non-Gaussian data.

Features

- Presents new nonparametric statistical methods for functional regression analysis, including Gaussian process functional regression (GPFR) models, mixture GPFR models, and generalized GPFR models
- Covers various topics in functional data analysis, including curve prediction, curve clustering, and functional ANOVA
- Describes the asymptotic theory for Gaussian process regression
- Discusses new developments in Gaussian process regression, such as variable selection using the penalized technique
- Implements the methods via MATLAB® and C, with the codes available on the author's website

Many flexible models based on Gaussian processes provide efficient ways of model learning, interpreting model structure, and carrying out inference, particularly when dealing with large dimensional functional data. This book shows how to use these Gaussian process regression models in the analysis of functional data.

CRC Press
Taylor & Francis Group
an **informa** business
www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
711 Third Avenue
New York, NY 10017
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

K11716