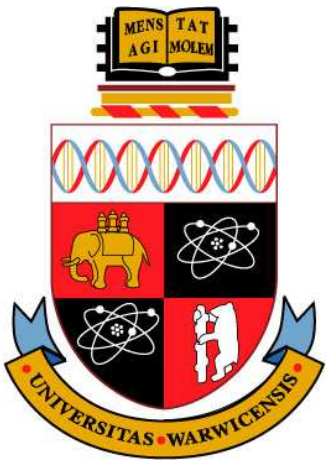

Penalized Spline Models and Applications

by

Maria João Costa

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

July 2008

THE UNIVERSITY OF
WARWICK

Para Os Meus Pais

(To My Parents)

Contents

Acknowledgements

I would like to thank my supervisor Dr Ewart Shaw for providing many ideas along the development of this thesis.

I am grateful to Yiannis Kosmidis for proof reading the current thesis and for providing valuable advice on how to improve the presentation. Many thanks! I would also like to thank Chris Cantwell for proof reading certain parts of the thesis.

This thesis could not have been successfully completed without the financial support from Fundação para a Ciência e a Tecnologia.

Throughout my long journey at the Department of Statistics I received support from many colleagues, friends and family.

I do not have the words to fully express my deep gratitude to my parents and my family. Without their love, support, encouragement and understanding, I would not have made it this far.

À minha Mãe, pelo seu Amor incondicional e por estar sempre do meu lado nos momentos mais difíceis. Mãmã, obrigada por tudo! Ao meu Pai, pelo seu Amor e carinho, por me ter ensinado a tentar ser sempre melhor. Sem a tua ajuda eu sei que não chegaria onde estou agora.

Aos meus irmãos, Pedro e Marcos, por terem suportado as minhas ausências e distância. À minha irmã Zé, por ter constituído, ao longo de todos estes anos, um

exemplo de determinação e coragem. A toda a minha família, por todo o apoio e carinho. Finalmente, a Helena Seca, por ser uma amiga presente, mesmo à distância.

To João, for all his love and support during the last months of this thesis. Thank you for your patient and understanding. Muito obrigada for ser meu fã número um!

I would like to thank Silvia Liverani and Thaís Fonseca for their friendship. Thank you for being there for me.

To Silvia, thank you for your guidance and your support, the many laughs, and for all the endless conversations about everything and nothing. Grazie!

To Thaís, thank you for providing the best advices, for always being in a good mood, and for being the best housemate ever. Valeu!

To Maria Vazquez, it was a pleasure to share this experience with you. Thank you for all your help and support during the rough times. Muchas gracias!

To Mylène Bédard, thank you for the long lunch breaks in the common room, and for being my MCMC guru. Merci!

To Mouna Akacha and Flávio Gonçalves, thank you for being my friends and the best officemates. Your laughs and support made this last few months much easier.

I would like to thank Miguel Ferreira, for his support, help and encouragement during my time at Warwick. I am also grateful to Diogo Pinheiro for his help and guidance when I fist arrived here.

Finally, to everybody at the Department of Statistics, for providing a stimulating and friendly environment during my time at Warwick.

Coventry, UK — Maria João Costa

July 31st, 2008

Declaration

I hereby declare that this thesis is based upon my own research, except when stated otherwise, in accordance with the regulations of the University of Warwick, and has not been submitted elsewhere.

The contents of Chapters three, four, and seven are based upon the research paper Costa & Shaw (2008). This is joint work with Ewart H. Shaw, however the set up and results were performed by myself.

Abstract

Penalized spline regression models are a popular statistical tool for curve fitting problems due to their flexibility and computational efficiency. In particular, penalized cubic spline functions have received a great deal of attention. Cubic splines have good numerical properties and have proven extremely useful in a variety of applications. Typically, splines are represented as linear combinations of basis functions. However, such representations can lack numerical stability or be difficult to manipulate analytically.

The current thesis proposes a different parametrization for cubic spline functions that is intuitive and simple to implement. Moreover, integral based penalty functionals have simple interpretable expressions in terms of the components of the parametrization. Also, the curvature of the function is not constrained to be continuous everywhere on its domain, which adds flexibility to the fitting process.

We consider not only models where smoothness is imposed by means of a single penalty functional, but also a generalization where a combination of different measures of roughness is built in order to specify the adequate limit of shrinkage for the problem at hand.

The proposed methodology is illustrated in two distinct regression settings.

Notation

Unless otherwise stated, the following notational conventions are used throughout the current thesis.

$\|\boldsymbol{x}\| = \sqrt{\sum_i x_i^2}$	the norm of $\boldsymbol{x}$
$\boldsymbol{A}^\intercal$	the transpose of $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	the inverse of $\boldsymbol{A}$
$[\boldsymbol{A}]^+$	the generalized inverse of $\boldsymbol{A}$
$\mathrm{rk}(\boldsymbol{A})$	the rank of $\boldsymbol{A}$
$\mathrm{tr}(\boldsymbol{A})$	the trace of $\boldsymbol{A}$
$\mathrm{diag}(a_1, \ldots, a_n)$	the diagonal matrix with elements $a_1, \ldots, a_n$
$\boldsymbol{A}_{i\cdot}$	the ith row of $\boldsymbol{A}$
$\Pr(E)$	the probability of the event E
X, x	random variable, observed value
$f_X(x)$	the probability density function of X
$\mathcal{F}(x) = \Pr(X \leq x)$	the cumulative probability distribution function of X
$\mathbb{E}[X]$, $V(X)$	the expected value of X, the variance of X

D	the available data
$(\ell)\, L$	the (log) likelihood function
$(\ell_p)\, L_p$	the (log) partial likelihood function
$p\,(\boldsymbol{\theta} \mid \lambda)$	the joint prior distribution of $\boldsymbol{\theta}$ with hyperparameter λ
$p\,(\boldsymbol{\theta} \mid D)$	the joint posterior distribution of $\boldsymbol{\theta}$ given D
$p\,(\boldsymbol{\theta} \mid \cdot)$	the full conditional distribution of $\boldsymbol{\theta}$
$h(t)$	the hazard function
$h_0(t)$	the baseline hazard function
$H(t)$	the cumulative hazard function
$H_0(t)$	the baseline cumulative hazard function
$S(t)$	the survival function
$g^{(d)}$	the dth derivative of g
g', g''	the first derivative of g, the second derivative of g
$\mathbb{R}$	the set of real numbers
$\mathbb{N} = \{1, 2, 3, \dots\}$	the set of natural numbers
$[l, r] = \{x \in \mathbb{R} : l \leq x \leq r\}$	a closed interval
$(l, r) = \{x \in \mathbb{R} : l < x < r\}$	an open interval
$\forall$	for all
$\triangle\gamma_m = \gamma_{m+1} - \gamma_m$	the forward difference
$\nabla\gamma_m = \gamma_m - \gamma_{m-1}$	the backward difference

Matrices are denoted by capital bold letters, vectors by lower case bold letters.

Abbreviations

The following abbreviations are used in the main text. In addition to their statement here, for the readers convenience, they are re-introduced in their first occurrence in the current thesis.

AIC	**A**kaike's **i**nformation **c**riterion
CPS	**c**urrent **p**opulation **s**urvey
DP	**d**ouble **p**enalty
GAMs	**g**eneralized **a**dditive **m**odels
GLM	**g**eneralized **l**inear **m**odels
MAP	**m**aximum **a** **p**osteriori
MCMC	**M**arkov **c**hain **M**onte **C**arlo
PBC	**p**rimary **b**iliary **c**irrhosis
PH	**p**roportional **h**azards
RSS	**r**esidual **s**um of **s**quares
SP	**s**ingle **p**enalty
TPB	**t**runcated **p**ower **b**asis
VFDP	**v**alue-**f**irst **d**erivative **p**arametrization

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Penalized Spline Regression: A Brief Review

Modern statistical theory began with the fitting of parametric models to data. The following is a typical scenario. A distribution is assumed for a response variable Y, and the mean or some other parameter is modeled as a linear function of a covariate X. The parameters of the linear function are then estimated by maximum likelihood. Examples of this are the normal linear regression model, the logistic regression model for binary data, or Cox's proportional hazards model for survival data (Cox, 1972). These models all assume a linear (or some parametric) form for the covariate effects.

The aforementioned parametric assumption turns out to be too restrictive for many practical applications. A more convenient framework is to move away from linear functions and model the dependency of Y on X in a nonparametric fashion, by replacing the linear or parametric function of the predictor by a smooth function $g(X)$. The smooth term g is typically estimated using spline functions though other techniques, such as local polynomials or kernel smoothers, are also available;

see, for example, Loader (2004). Spline functions, however, enjoy great popularity, being intrinsically connected with the important statistical problem of curve fitting (Schoenberg, 1964a,b), and having good numerical properties (De Boor, 1978; Schumaker, 1981; Green & Silverman, 1994).

Until recently, the techniques for the estimation of the smooth terms in a regression model using spline functions followed two main approaches: regression splines and smoothing splines. Regression splines are defined using a small, carefully chosen number of knots to guarantee smoothness. Hence, their position on the domain of the curve to be estimated plays a crucial role, as more knots should be placed in regions of greater flexibility of the underlying true function. Data-driven methods for knot placement have been developed in the literature. Friedman (1991) proposed an adaptive knot selection algorithm, calling it "MARS" for Multivariate Adaptive Regression Splines. A Bayesian approach using reversible jump Markov chain Monte Carlo (RJMCMC, see Green, 1995) was proposed by Denison et al. (1998b).

Smoothing splines arise as the solution to an optimization problem. Schoenberg (1964a,b) defined smoothness of a curve through the integral of the squared dth order derivative. The solution to the resulting penalized residual sum of squares criterion is a spline of order $2d - 1$ with a knot at every design point (Schoenberg, 1964a,b). The idea of penalizing a measure of goodness of fit by one of roughness goes back at least to Whittaker (1923). The work by Wahba (1978, 1983, 1990) and co-authors, and that of Silverman (1985) opened up the theory of smoothing spline functions to the statistical literature. Green & Silverman (1994), or Eubank (1999), provide an excellent overview of smoothing spline techniques and applications in statistics.

The discussion above presents smoothing splines as the best solution for estimat-

ing g in the sense that smoothing splines solve a well defined optimization problem. However, the number of parameters to be estimated is as large as the number of observations (or design points), making them a computationally intensive choice for modeling the unknown functions g, especially if the number of smooth terms p in the model is large.

Almost simultaneously, Eilers & Marx (1996) and Ruppert & Carroll (1997) proposed the use of penalized splines (or P-splines), a different approach which can be seen as a compromise between smoothing and regression splines. The procedure dates back from Wahba (1980), or O'Sullivan (1986), but it was with the papers by Eilers & Marx (1996) and Ruppert & Carroll (1997) that it achieved general recognition. The key idea is to represent the curve g by an overfitted spline function, and to control the smoothness by subtracting a penalty term from the model's likelihood function, in similar fashion to smoothing splines. Nevertheless, in penalized splines the number of knots is typically far less than the number of observations. Hence, penalized splines are more efficient than smoothing splines from a computational point of view. Eilers & Marx (1996) represented splines as linear combinations of B-spline functions (De Boor, 1978) on an equidistant grid of knots. For this particular choice of knots, Eilers & Marx (1996) derived an approximation to the penalty defined by Schoenberg (1964a,b) and called it difference penalty. This setup is easy to use, and allows great flexibility, in that any order of penalty can be combined with any order of the B-spline basis. However, equidistant knots are not always suitable. Take, for example, the case of data arising in survival analysis. The presence of censoring may create regions in the domain of the curve one wishes to estimate with little or no information regarding its shape, and so extra care is needed when choosing the locations of the knots (Gray, 1992).

Ruppert & Carroll (1997) use the truncated power basis (TPB) to represent the components g as penalized splines. Smoothness is controlled by a ridge type penalty over the coefficients of the parametrization. Rather than placing the knots on an equidistant grid, Ruppert & Carroll (1997) chose to use equidistant quantiles of the observations for the variable X as knot locations.

Most of the research involving penalized spline regression models considers a single penalty functional as a measure of roughness. However, single penalty models may have limitations concerning the right specification of the limit of smoothness (Gray, 1992; Dannegger et al., 1995). Moreover, Marx & Eilers (1998), Eilers & Marx (2003) and Eilers & Goeman (2004) have shown that a combination of penalty functionals may produce estimates with better tail behaviour. Aldrin (2006) focused on the problem of prediction and found that combined penalty functionals tend to yield models with better prediction ability.

Penalized spline models have enjoyed increased popularity since the papers by Eilers & Marx (1996) and Ruppert & Carroll (1997). The range of applications is vast. Generalized additive models (GAMs, see Hastie & Tibshirani, 1986, 1990b) and nonproportional hazards models are two such examples. GAMs are an extension of additive models to response variables that belong to the exponential family of distributions. Many useful distributions fall into this category, for example, the Bernoulli or the Poisson. Penalized splines have been extensively used to model the smooth terms in a GAM. Lang & Brezger (2004) provided an extensive simulation study within a Gaussian regression setting, comparing Bayesian P-splines with several competing alternatives. Crainiceanu et al. (2007) modeled heteroscedastic errors and spatially adaptive smoothing through penalized spline functions using Bayesian mixed model inference tools. Fahrmeir et al. (2004) focused upon the

analysis of space-time data. They proposed several methodologies for Bayesian inference and concluded that a hybrid approach combining both empirical and full Bayes posterior analysis might yield better spline estimates. Stemming from an idea first introduced by Gamerman (1997), Brezger & Lang (2006) provided efficient Markov chain Monte Carlo (MCMC) algorithms to sample from the posterior of interest in a GAM. The book by Ruppert et al. (2003) is an excellent reference for nonparametric and semiparametric regression models based upon P-splines.

Cox's proportional hazards (PH) model and partial likelihood function (Cox, 1972) allow the impact of covariates on survival to be estimated in a flexible manner since the distribution of the survival times needs not be specified. However, inferences rely upon the proportional hazards assumption, which is not always appropriate. Many techniques have been developed to overcome the PH constraint. One such approach is to model the covariate effects as smooth functions of the follow-up time that can be well approximated by spline functions. Penalized splines have also been a valuable tool here. Basing inferences upon the partial likelihood function, Gray (1992) modeled covariate effects considering both quadratic splines and piecewise constant functions. Gray (1992) argues that such models provide estimated effects with better right tail behaviour. Moreover, quadratic splines and piecewise constant functions shrink the estimates towards the PH assumption. Kauermann (2005) also allowed time-varying effects of the predictors but modeled the baseline hazard as well by deriving a Poisson approximation to the model's full likelihood function. A similar route was followed by Lambert & Eilers (2005) within a life-table approach. Hennerfeind et al. (2006) and Kneib & Fahrmeir (2007) studied the general class of geoadditive survival models that include time-varying covariate effects as a particular case.

1.2 Outline of Thesis

The current thesis is organized in the following way. In Chapter 2 we provide a brief description of spline functions and their properties. Here, splines are defined as local polynomial functions. We discuss the two most common parametrizations for polynomial splines, the TPB and the B-spline basis, pointing out advantages and disadvantages of both representations. This chapter concludes with a short overview of additive models and their implementation using spline functions.

Chapter 3 introduces the ideas behind the penalized likelihood methodology. We start by motivating the use of penalized splines, following with a discussion on the bias-variability trade-off and on several forms for the penalty term. The standard single penalty functional and its role in the smoothing spline approach are described. We then propose a generalization of the penalized likelihood criterion based upon double penalty functionals. This generalization allows us to correctly specify the limit of smoothness inherent in the criterion to be optimized. Moreover, whilst a single penalty model is appropriate in a variety of settings, for some applications a double penalty functional may provide better spline estimates. Finally, the Bayesian interpretation of the penalized likelihood criterion is discussed.

In Chapter 4 we propose a parametrization for cubic spline functions called value-first derivative parametrization (VFDP). The parametrization is defined locally and is easy to set up and implement. It adds flexibility to the curve fitting problem by allowing the second derivative of the cubic spline to be discontinuous across the knots. We show that standard single and double penalty functionals have simple expressions in terms of the components of the parametrization. Additionally, we see that the penalization of the different levels of complexity of the spline curve can be

made explicit through the VFDP.

Chapter 5 describes the Bayesian inference process for additive models in detail. Double penalty estimates are obtained with a hybrid approach that combines empirical Bayes methods with an MCMC algorithm. The results from a simulation study designed to compare the VFDP with the standard approach based upon B-spline basis functions are presented in Chapter 5. The two parametrizations are compared in terms of the performance of the resulting splines regarding estimation with both single and double penalty functionals. We conclude that splines represented in terms of the VFDP can outperform those in the span of a B-splines basis, particularly when double penalty functionals are implemented.

Chapter 6 concerns the application of the VFDP to the broad class of GAMs. We outline the theory for the closely related class of generalized linear models and describe a stochastic version of the local scoring algorithm to obtain posterior estimates of the smooth terms in a GAM through MCMC techniques. The proposed methodology, based upon the VFDP, is then illustrated using a data set involving a binary response variable. Single and double penalty spline estimates are computed, with the latter providing a better fit to the data.

A more complex setting is that of nonproportional hazards models described in Chapter 7. This chapter starts with a review of important concepts in the survival analysis framework. The Cox PH model and the partial likelihood function are presented. We then focus upon a more general model that overcomes the proportional hazards assumption by allowing covariate effects to vary smoothly with time. The latter are estimated through the VFDP. Posterior estimates are obtained with an MCMC scheme similar to that described in Chapter 6. However, the risk sets in the partial likelihood increase the complexity of the algorithm. In the real data

application we consider, covariate effects obtained with the double penalty model tend to have better right tail behavior than those obtained with a single penalty functional.

Chapter 8 indicates some related topics for further research. A summary of the main results is given in Chapter 9.

Appendix A includes the proof to the equivalences stated in Chapter 4. Appendix B describes the general form for both the Gibbs sampler and the Metropolis-Hastings algorithm. Some selected chain paths from the MCMC outputs resulting from the data analysis carried out in Chapter 6 and Chapter 7 are presented in Appendix C. Finally, Appendix D contains the MATLAB 7.0.1 (The MathWorks, 2008) codes and functions developed to carry out the analysis presented in Chapter 7.

Very general terms and concepts commonly used in Bayesian statistics are not defined. A good reference for both basic and in depth concepts and methods is Carlin & Louis (2000). The data sets used in Chapter 6 and Chapter 7 can be found on the StatLib website (`lib.stat.cmu.edu`). The thesis was typeset using LaTeX under the WinEdt distribution (`www.winedt.com`). When terms are introduced for the first time, they will be highlighted by the use of italics.

Chapter 2

Spline Models

2.1 Introduction

Linear regression is one of the oldest and most widely used statistical techniques. A typical setting comprises a univariate response variable whose expected value is modeled in terms of a linear predictor, a parametric function of a set of predictor variables thought to influence the value of the response variable. Inference is based upon the key assumption that the linear predictor depends linearly upon the parameters that define it.

There are very many data sets where a linear model provides an inappropriate fit. The classical approach is to extend the linear predictor by adding polynomial functions of the predictor variables. However, ordinary polynomial models are inadequate in many situations. This is particularly the case when one approximates functions that arise from the physical world. Functions that express physical relationships are frequently of a disjoint or dissociate nature. This is to say that their behavior in one region may be completely unrelated to their behavior in another region. Polynomials, along with most other mathematical functions, have just

the opposite property. Namely, their behavior in a small region determines their behavior everywhere.

Splines do not suffer this handicap. Spline functions are piecewise polynomials, with the polynomial pieces joining in the so called *knots* and fulfilling continuity conditions for the function itself and some of its derivatives. Splines have been introduced in the statistical literature as interpolators. However, most statistical problems deal with data subject to measurement error. Hence, in this case, it is desirable to create a type of spline that could pass near, in some sense, to the data but not be constrained to interpolate exactly. This is known as spline smoothing and is related to the problem of curve fitting, upon which the current thesis focus.

In this chapter we introduce spline functions and their application to a general class of regression models termed *additive models* (Hastie & Tibshirani, 1990b). Our main aims are i) to review some standard definitions and concepts concerning spline functions and ii) to provide the link between spline functions and additive models.

Spline functions and their properties have been extensively studied in the past; De Boor (1978) is the standard technical textbook on splines. Also, Schumaker (1981) and Wahba (1990) provide excellent background on spline functions. The book by Ruppert et al. (2003) provides a good account of the use of splines in regression models.

2.2 Spline Functions

We shall treat only univariate polynomial splines. Extensions to higher dimensions can be achieved through the tensor product of a number of univariate spline functions. Surface smoothing is beyond the scope of the current thesis.

2.2.1 Definition and Properties

A spline function $g(x)$ of degree s is a piecewise polynomial. The polynomial pieces (all of degree s) join together at the *knots* k_m, $m = 1, \ldots, \mathcal{K}$. For our purposes, the set of knots $\{k_m\}_{m=1}^{\mathcal{K}}$ will always represent a strictly increasing sequence. Hence, for $m = 1, \ldots, \mathcal{K}-1$, we can write

$$g(x) = G_m(x) = c_{0m} + c_{1m}x + c_{2m}x^2 + \ldots + c_{sm}x^s, \quad k_m \leq x < k_{m+1}\,. \tag{2.1}$$

We shall suppress the dependency of $g(x)$ upon x whenever this is clear from the context. The plot in Figure 2.1 represents the polynomial pieces defining a cubic spline (s=3) with five knots $k_1, \ldots, k_5$.

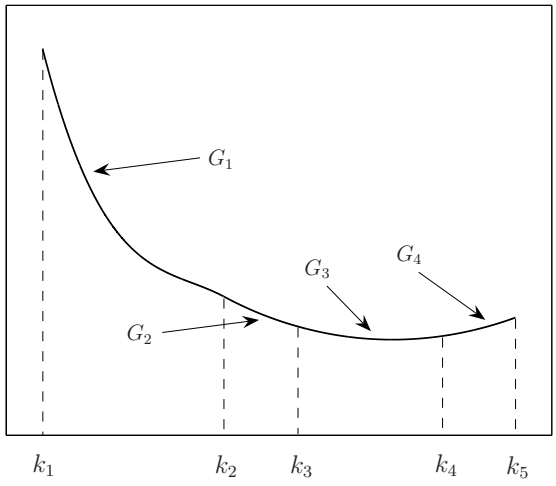

Figure 2.1: The four polynomial pieces making up a cubic spline with five knots.

The polynomial pieces $G_m(x)$ join smoothly at the knots, obeying continuity

conditions on the function and typically also on its first $s-1$ derivatives, i.e.,

$$G_m^{(d)}(k_{m+1}) = G_{m+1}^{(d)}(k_{m+1}), \quad d = 0, \ldots, s-1, \ m = 1, \ldots, \mathcal{K}-2\,. \tag{2.2}$$

The parameters defining the spline function g are:

i) The degree of the spline function, s.

ii) The number of knots, $\mathcal{K}$.

iii) The position of the knots, $\{k_m\}_{m=1}^{\mathcal{K}}$.

iv) The number of free coefficients of the spline function, $\mathcal{K}+s-1$ if the equalities in (2.2) hold; to see this note that each of the $\mathcal{K}-1$ polynomial pieces has $s+1$ coefficients, and that the s continuity conditions at each interior knot introduce $(\mathcal{K}-2)s$ constraints, leaving $(\mathcal{K}-1)(s+1)-(\mathcal{K}-2)s = \mathcal{K}+s-1$ free coefficients. Thus, the space of spline functions of degree s with $\mathcal{K}$ knots, continuous and with $s-1$ continuous derivatives is an $(\mathcal{K}+s-1)$-dimensional space.

The sth derivative of the spline will be a step function with (possible) jumps at the interior knots $k_2, \ldots, k_{\mathcal{K}-1}$. A popular subset of spline functions, called *natural splines*, is defined by one further restriction. If $s+1$, the order of the spline, is even, then g is a natural spline if it fulfills the following condition: g is a polynomial of order $(s+1)/2$ outside $[k_1, k_{\mathcal{K}}]$. If (2.2) holds, these further $(s+1)/2$ constraints decrease the number of free parameters to $\mathcal{K}-2$. Natural spline functions are popular because they arise as the solution to an optimization problem as we shall see later on in Chapter 3. Other boundary conditions are usually better. De Boor (1978) provides a thorough analysis of several boundary (or end-knot) conditions.

The knots $\{k_m\}_{m=1}^{\mathcal{K}}$ cover the domain of the variable x of interest. Thus, if $x \in [l, r]$, then $k_1 = l$ and $k_{\mathcal{K}} = r$. Every function on the interval $[l, r]$ can be approximated arbitrarily well by polynomial splines with the degree s fixed, provided a sufficient number of knots are allowed. In principle, the number and positions of the knots are free parameters that need to be estimated. In practice they are often taken to be fixed. If one has particular knowledge of the shape of the function to be approximated, then more knots should be placed in regions of greater variability (e.g., positions of maxima and minima). There exist strategies for the optimal selection of the number and position of the knots (Friedman, 1991). These are typically iterative algorithms which seek to minimize some form of goodness-of-fit criterion through the addition or deletion of knots. On the other extreme we find approaches that avoid the selection of the number and positions of knots altogether. Smoothing splines (Wahba, 1990; Hastie & Tibshirani, 1986) have a knot at each distinct observed value of x. Hence, in this case, the locations of the knots are determined by the observed values of x. Penalized splines (Eilers & Marx, 1996) take $\mathcal{K}$ to be large but less than n, the number of observations. Overfitting in both smoothing and penalized splines is controlled by means of a *penalty functional* of the spline g. Penalized spline models are the topic of Chapter 3.

2.2.2 Spline Parametrizations

The definition of a spline function in terms of polynomials as in (2.1) is convenient once the polynomial coefficients are known. However, for both computational of for mathematical discussion, it is usually simpler to define the spline of degree s with knots $\{k_m\}_{m=1}^{\mathcal{K}}$ as a linear combination of *spline basis functions*. The latter are a

set of linearly independent spline functions of degree s with knots $\{k_m\}_{m=1}^{\mathcal{K}}$ which span the desired space of spline functions.

2.2.2.1 Truncated Power Basis

We shall start by defining the *truncated power basis* (TPB) for splines of degree s. The building block is the so called *truncated power function* which takes the form

$$(x)_+^s = \max\{x^s, 0\}.$$

The function $\varphi(x) = (x-k)_+^s$ is a piecewise polynomial, of degree s, with one breakpoint at k. For $s > 0$ the function $\varphi(x)$ is continuous across k and has $s-1$ continuous derivatives with a jump in the sth derivative at k of size $s!$. If $s = 0$ then $\varphi(x)$ has a jump of size 1 at k. Figure 2.2 illustrates a set of truncated power functions for $s = 3$. For a spline function g of degree s with knots $\{k_m\}_{m=1}^{\mathcal{K}}$ the truncated power basis is defined as

$$g(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_s x^s + \sum_{m=2}^{\mathcal{K}-1} \gamma_m \varphi_m(x) \quad x \in [l,\, r]\,, \tag{2.3}$$

where $\varphi_m(x) = (x-k_m)_+^s$. We call k_m the *defining knot* of $\varphi_m(x)$. Any continuous spline function of degree s with knots $\{k_m\}_{m=1}^{\mathcal{K}}$ and $s-1$ continuous derivatives can be expressed in the form (2.3). Again we require $\mathcal{K}+s-1$ parameters to represent the spline g.

It turns out that the TPB is not the most convenient representation for spline functions. For a very nonuniform grid of knots $\{k_m\}_{m=1}^{\mathcal{K}}$, some of the basis functions $\varphi_m(x)$ become nearly linearly dependent on the others. Thus, small changes in the

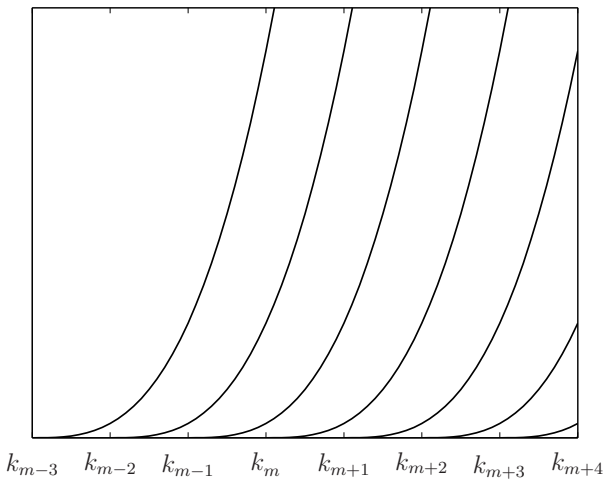

Figure 2.2: Truncated power functions of degree 3.

coefficients γ_m might produce much smaller or much larger changes in the function $g(x)$ that we wish to approximate. Further, even though a particular truncated power function is designed to accommodate one knot interval, it is evaluated at all points to the right of its defining knot and the numbers usually become large. This will result in badly conditioned systems when estimating the coefficients of the representation.

2.2.2.2 B-splines Basis

B-splines (De Boor, 1978) can be seen as a generalization of the truncated power functions that tries to overcome the aforementioned numerical problem. Essentially, B-splines are obtained by forming appropriately scaled $(s+1)$th divided differences of the truncated power functions to obtain a new basis whose elements vanish outside

a relatively small interval. Let

$$\psi_l(x) = \frac{(k_l - x)_+^s}{\prod\limits_{\substack{v=m \\ v \neq l}}^{m+s+1} (k_l - k_v)} .$$

A B-spline function of degree s is defined as

$$B_m(x) = (k_{m+s+1} - k_m) \sum_{l=m}^{m+s+1} \psi_l(x) . \tag{2.4}$$

B-spline functions have a number of useful properties:

i) From the definition in (2.4) we see that the function $B_m(x)$ has small support, i.e.,

$$B_m(x) = 0 , \quad \text{for } x \notin [k_m, k_{m+s+1}] .$$

One consequence of the above property is that only $s + 1$ B-splines have any particular interval $[k_u, k_{u+1}]$ in their support. These will be $B_{u-s}, \ldots, B_u$. Note that we need an additional set of s knots at each end of the domain of g in order to generate a complete B-spline basis. So, given $\{k_m\}_{m=1}^{\mathcal{K}}$, the additional $2s$ knots are such that

$$\underbrace{k_{-(s-1)} < \ldots < k_0}_{s \text{ additional knots}} < k_1 < \ldots < k_{\mathcal{K}} < \underbrace{k_{\mathcal{K}+1} < \ldots < k_{\mathcal{K}+s}}_{s \text{ additional knots}} .$$

These extra knots can be placed in an arbitrary way as they will not affect the quality of the fitted curve in the domain of interest, $[l, r]$.

ii) $\sum\limits_m B_m(x) = 1 , \quad x \in [l, r]$.

iii) The functions $B_m(x)$ are positive on their support, i.e.,

$$B_m(x) > 0\,, \quad x \in (k_m, k_{m+s+1})\,.$$

The plot in Figure 2.3 represents a set of adjacent B-spline functions of degree $s = 3$.

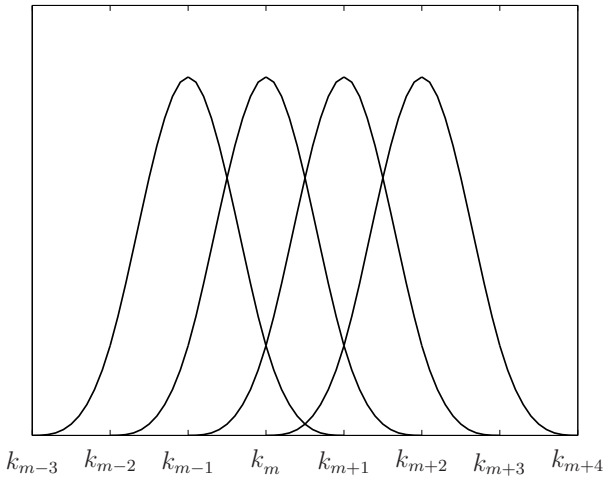

Figure 2.3: Illustration of the set of B-spline functions of degree 3 having the knot interval $[k_m,\, k_{m+1})$ *on their support.*

The local nature of B-spline functions results in a more numerically stable fitting process when compared to that obtained using truncated power basis functions. The B-spline representation of the function g of interest takes the form:

$$g(x) = \sum_{m=1}^{\mathcal{K}_B} \gamma_m B_m(x)\,, \quad x \in [l,\, r] \tag{2.5}$$

where $\mathcal{K}_B = \mathcal{K} + s - 1$ is the dimension of the B-spline representation. A complete treatment of B-spline functions and their properties is given in De Boor (1978).

2.3 Extending the Linear Regression Model

This section introduces the general class of additive models and motivates the representation of its smooth terms through spline functions. The additive model is a generalization of the usual linear regression model. Hence, we start this section with a review of the linear model and its limitations.

Consider the standard multiple linear regression problem. Given a response variable Y and p selected explanatory variables (or covariates) $X_1, \ldots, X_p$, our goal is to model the dependency of Y on $X_1, \ldots, X_p$. The standard tool is the multiple linear regression model:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon\,, \tag{2.6}$$

where $\epsilon \sim N(0, \sigma^2)$, independently of the X_j's. This model makes the strong assumption that the dependency of $\mathbb{E}[Y]$ on $X_1, \ldots, X_p$ is linear in each of the predictors. If this assumption holds, even approximately, then the linear regression model in (2.6) is an extremely useful tool because not only does it provide a simple description of the data, it also summarizes the contribution of each predictor with a single coefficient, the β_j's.

Linear models such as that in (2.6) are often too simple for most practical applications. There are many ways in which the linear model (2.6) can be generalized. A straightforward extension consists in adding polynomial terms like X_j^2, X_j^3, etc., to the model in (2.6). However, it is usually difficult to determine which such terms should be added in order to obtain a good fit to the data. Furthermore, polynomial models suffer from a series of drawbacks. As mentioned in Section 2.1, the data fits

are not local, which means that perturbations in the data may affect the polynomial fit in remote regions. Also, the fitting process for polynomials can be numerically ill-conditioned due to the presence of collinearity, a result of the spurious correlation among the power transformations of the predictors X_j. Surface smoothers are another possible generalization of the model (2.6). Essentially these model the dependency of Y on $X_1, \ldots, X_p$ through a p-dimensional surface as follows:

$$Y = g(X_1, \ldots, X_p) + \epsilon \,. \tag{2.7}$$

However, it is not clear how one should define a local fit in p dimensions, though tensor product of splines attempt to deal with such issue (see, e.g., Wahba, 1990; Wood, 2006b). Moreover, the effect of any individual predictor on the response Y is difficult to interpret if the surface smoother in (2.7) is fitted to the data.

The aforementioned interpretation problem highlights an important feature of the linear model that has made it so popular for statistical inference: the linear model is additive in the predictor effects. This additivity property implies that one can examine the predictor effects separately, in the absence of interactions. *Additive models* (Hastie & Tibshirani, 1986, 1990b) retain this important feature as they are additive in the predictor effects. An additive model is defined by

$$Y = \beta_0 + \sum_{j=1}^{p} g_j(X_j) + \epsilon \,, \tag{2.8}$$

where as before the error ϵ is independent of the X_j's with $\epsilon \sim N(0, \sigma^2)$. The components $g_j(X_j)$ are arbitrary smooth univariate functions. They define 'transformed predictors' $g_j(X_j)$ which act additively on the response variable. Thus we can ex-

amine the effect of each predictor in the model separately. For estimation purposes the assumption that $\mathbb{E}[g_j(X_j)] = 0$ is required, since otherwise the intercept β_0 in (2.8) is unidentifiable, i.e., any of the fitted g_j could be shifted by some arbitrary constant, accompanied by an offsetting shift in β_0, and the resulting set of estimates would fit the data equally well.

The model in (2.8) assumes that all predictors take values over a continuous domain and estimates all its components in a nonparametric fashion. In some situations, however, one or more predictors may be categorical variables. In this case the additive model in (2.8) can be modified to include both linear and smooth terms. If there exist q such categorical predictors, $V_1, \ldots, V_q$ say, a *semiparametric model* is defined as

$$Y = \beta_0 + \beta_1 V_1 + \cdots + \beta_q V_q + \sum_{j=1}^{p} g_j(X_j) + \epsilon \,. \tag{2.9}$$

More complex models, involving interaction terms, like $g(X_u, X_v)$ for example, can also be included in the model. However, in this thesis we shall focus on additive models such as those in (2.8) and (2.9).

Having decided on the regression model, one needs to choose how to represent the unknown smooth terms g_j. The discussion in Section 2.1 and Section 2.2 regarding the local nature of spline functions presents them as a flexible approximation tool. Additionally, De Boor (1978), and also Green & Silverman (1994), derive important results regarding the numerical optimality of spline interpolants, and in particular of cubic spline interpolants, since these minimize an intuitive measure of roughness based upon the curvature of the function to be estimated. We shall see in the next chapter that cubic spline functions are also optimal in the sense that they are the solution to an optimization problem within the framework of curve fitting.

The discussion above suggests that spline functions ought to be a good choice for representing smooth curves in any statistical framework, and hence are the building block of all the smooths presented in this thesis.

2.4 Summary

The definition of a spline function and its properties have been presented. We have seen that spline functions are a flexible modeling tool with good numerical properties. We described and discussed two common parametrizations for spline functions. Finally, we presented the very general class of additive models and briefly discussed the representation of the smooth terms in an additive model using spline functions.

Chapter 3

A Review of Penalized Likelihood Methods

3.1 Introduction

Statisticians are continually faced with the problem of recovering a smooth function when only noisy measurements of it are available. The additive model (2.8) described in the previous chapter, where, in the presence of error ϵ, interest lay in estimating the smooth terms g_j, is one such example.

Smoothing methods have become a popular modeling tool in a wide class of statistical contexts. The main idea behind any smoothing technique is to allow the data to dictate the shape of the curve of interest rather than imposing a rigid parametric structure. For this reason, smoothing methods are often referred to as *nonparametric*. Running averages, kernel smoothers, etc. are, together with spline smoothing, a few examples of smoothing methods.

Spline smoothing techniques are good solutions to the estimation of the true function (known only to be smooth) for three reasons mainly. First, they are flex-

ible enough to respond to local variation without allowing pathological behaviour. Second, the actual degree of smoothing is controllable, even when unknown, and third, the good theoretical properties of splines suggest that they are a good model for any smooth function.

The spline smoothing approach to curve fitting gained significant visibility with the work of Schoenberg (1964a,b), and then later with Wahba (1978, 1983) and co-authors. The different methodologies for curve fitting problems based upon spline functions can be categorized as: smoothing splines, regression splines, and penalized splines. Smoothing splines have a knot at each unique observed value of the variable of interest. Overfitting is controlled by means of a *roughness penalty functional* of the curve. Smoothing splines require that many parameters be estimated, typically at least as many parameters as there are observations, and therefore special algorithms are needed to attain computational efficiency. See Wahba (1990) or Green & Silverman (1994) for a detailed technical treatment of smoothing splines and their properties. On the other extreme lie regression splines, which avoid the use of a penalty functional by controlling for smoothness through a careful selection of the number and positions of the knots. Friedman (1991) proposed an adaptive knot selection algorithm, calling it "MARS" for Multivariate Adaptive Regression Splines. A Bayesian version of the latter was proposed by Denison et al. (1998b).

More recently, Eilers & Marx (1996) re-introduced the use of penalized splines (e.g. Wahba, 1980), a low rank smoother that can be seen as a compromise between smoothing and regression splines. In penalized splines, the number of knots defining the spline function is larger than that justified by the data, but smaller than the number of observations. The level of overfitting is controlled by placing a roughness penalty over the curve, in similar fashion to smoothing splines.

Later in this thesis we shall see that smoothing splines arise naturally as the solution to a well defined optimization problem and are therefore, in some sense, the best approximation to the curve one wishes to estimate. However, smoothing splines carry as many parameters as there are data, though overall their effect will be constrained, in some way, by the penalty functional. Hence, although penalized splines do not share the same good approximation properties of smoothing splines, the reduction in computational cost through the use of a low rank smoother can be enormous, specially for large sample sizes and models involving several smooths. Furthermore, penalized splines are more general than smoothing splines in that one can use as many or as few knots as desired. In this thesis we shall use the penalized spline approach to estimate the smooth curves of interest.

This chapter starts with a brief overview of the general penalized log-likelihood criterion. We describe standard quadratic penalty functionals based on the integral of a squared derivative of the curve, and some extensions of the single penalty case where a combination of roughness measures is used. Finally, we discuss the intrinsic link between penalized log-likelihood methods and Bayesian inference. For a complete treatment of penalized regression methods see Green & Silverman (1994) or Ruppert et al. (2003).

3.2 The Penalized Log-Likelihood Criterion

For simplicity we introduce penalized likelihood methods in the univariate case. Recall from Chapter 2 that the aim in nonparametric regression is to summarize the trend of a response Y as a function of the predictor measurement X by producing an estimate of the trend, $g(X)$, that is less variable than Y itself. For example, through

the regression model $Y = g(X) + \epsilon$, where ϵ is a zero-mean error component whose distribution is independent of the covariate X.

Following the work by Eilers & Marx (1996), the curve g is approximated by an overfitted spline function. Unconstrained estimation of g would result in a rapidly varying curve, translating the local variations in the observed data. In most applications, however, it is more plausible to take these local variations to be the result of random noise and to study the underlying, more slowly varying, trend in the data. This reflects the two conflicting goals in curve estimation: to obtain, simultaneously, a good fit to the data and a curve estimate that is not too wiggly or rough. The basic idea behind penalized regression methods is to quantify the notion of roughness of a curve through a suitable penalty functional and then to pose the estimation problem in a way that makes explicit the necessary compromise between bias and variability in curve fitting. Whittaker (1923) was the first to discuss the problem of balancing goodness-of-fit and smoothness. The idea of combining a relatively large number of knots with a penalty on the roughness of the fitted curve dates back to the work of O'Sullivan (1986).

Denote by D the available data and take $\ell(g \mid D)$ to be some appropriate log-likelihood function of the model at hand. Let $P(g\,;\lambda) = \lambda\, F(g)$, $\lambda \geq 0$, be a roughness penalty controlling for rapid variations in some feature of the curve g as defined by the functional $F(\cdot)$. All the penalties studied in this thesis are quadratic and satisfy $P(g\,;\lambda) \geq 0$, $\forall\, g$, λ. Penalized likelihood methods estimate g by maximizing the penalized log-likelihood criterion

$$\mathcal{J}(g) = \ell(g\,; D) - P(g\,;\lambda), \quad \lambda \geq 0\,. \tag{3.1}$$

The parameter λ in $P(g\,;\lambda)$ is usually called *smoothing parameter* and plays a central role in penalized likelihood methods. It governs the trade-off between fidelity to the data, as measured by $\ell(g \mid D)$, and smoothness, as measured by $P(g\,;\lambda)$. Large values of λ produce smoother curves while smaller values result in more wiggly curves. At the one extreme, as $\lambda \to +\infty$, the penalty term dominates the value of $\mathcal{J}(g)$, forcing the estimate $\widehat{g}$ to be the curve that yields a value of zero for $F(g)$. At the other extreme, as $\lambda \to 0$, the penalty term becomes irrelevant and the solution $\widehat{g}$ is found by maximizing the log-likelihood term $\ell(g\,;D)$ and hence will follow the data closely.

In what follows we present some of the most common penalty functionals used in the literature and their properties.

3.2.1 Single Penalty Models

Most spline smoothing methods rely upon a single quadratic penalty functional to attain the desired degree of smoothness when estimating the curve g of interest. Whittaker (1923) used differences of order d to smooth life tables arising in the actuarial sciences. Later, Schoenberg (1964a,b) worked on the same idea but replaced the dth order differences by integrals. Schoenberg's penalty measures roughness of a curve by the value of the integral of the square of the dth derivative of g, i.e.,

$$\lambda \int_l^r \left[g^{(d)}(x)\right]^2 dx\,, \quad \lambda \geq 0\,. \tag{3.2}$$

Hereafter, and for simplicity of notation, we shall suppress the use of integration limits, and assume that each integral is over a range that covers that of the vari-

able in question. Schoenberg (1964a,b) studied the penalty (3.2) in the context of the following least squares problem. Take data (y_i, x_i), $i = 1, \ldots, n$, such that $l < x_1 < \cdots < x_n < r$, and consider the regression model $y = g(x) + \epsilon$, $\epsilon \sim N(0, \sigma^2)$, for some smooth function g. As discussed in Section 3.2, unconstrained least squares estimation of g results in a curve interpolating the data points (y_i, x_i). Smoothness is attained by estimating g through the curve that minimizes the penalized residual sum of squares criterion,

$$\sum_{i=1}^{n} [y_i - g(x_i)]^2 + \lambda \int \left[g^{(d)}(x)\right]^2 dx \,, \tag{3.3}$$

with the parameter λ as in Section 3.2. The solution $\widehat{g}$ is a natural smoothing spline of degree $2d-1$ with knots at the x_i's, even though no constraint is imposed upon $\widehat{g}$ that it even be a spline. The limit of smoothness, when $\lambda \to \infty$, is a polynomial of degree $d-1$, for which the integral in (3.3) is zero. As $\lambda \to 0$ the resulting fitted curve $\widehat{g}$ becomes an interpolating spline.

In the context of penalized splines, Eilers & Marx (1996) proposed the use of what they called *difference penalties* for spline curves g in the span of a B-spline basis defined by equally spaced knots. If $\{\gamma_m\}_{m=1}^{\mathcal{K}_B}$ are the coefficients of such representation, then a difference penalty of order o is defined as

$$\lambda \sum_{m=o+1}^{\mathcal{K}_B} (\nabla^o \gamma_m)^2 \,, \quad \lambda \geq 0 \,, \tag{3.4}$$

with $\nabla^o \gamma_m = \underbrace{\nabla \ldots \nabla}_{o-1}(\nabla \gamma_m)$, and $\nabla \gamma_m = \gamma_m - \gamma_{m-1}$. The penalized residual sum of

squares criterion becomes

$$\sum_{i=1}^{n} [y_i - g(x_i)]^2 + \lambda \sum_{m=o+1}^{\mathcal{K}_B} (\nabla^o \gamma_m)^2 \ . \tag{3.5}$$

For large values of λ the fitted curve $\widehat{g}$ minimizing (3.5) will approach a polynomial of degree $o-1$ if the degree of the B-spline basis is equal to, or higher than, o. Eilers & Marx (1996) show that a second order difference penalty ($o = 2$) approximates the penalty functional (3.2) for $d = 2$. This results from discarding terms in (3.2) corresponding to integrals involving cross-products of second-order derivatives of neighboring B-spline functions. For non-equidistant knots, or for other orders of the derivative, the approximation in Eilers & Marx (1996) is no longer valid.

For completeness, we also discuss penalties for the truncated power basis introduced in Chapter 2. Let $\{\gamma_m\}_{m=2}^{\mathcal{K}-1}$ be the coefficients corresponding to the truncated power functions in the basis representation (2.3). For a spline g of degree s with knots $\{k_m\}_{m=1}^{\mathcal{K}}$, the size of the jump of the derivative of order s at the interior knots k_m, $m = 2, \ldots, \mathcal{K}-1$, is given by γ_m. Hence, smoothness can be achieved by shrinking the coefficients $\{\gamma_m\}_{m=2}^{\mathcal{K}-1}$ to zero, i.e., by defining the penalty

$$\lambda \sum_{m=2}^{\mathcal{K}-1} \gamma_m^2 \,, \quad \lambda \geq 0 \,. \tag{3.6}$$

Of particular interest in the context of regression problems is the case $d = 2$ in (3.2). The penalty functional becomes

$$\lambda \int g''(x)^2 dx \,, \quad \lambda \geq 0 \,, \tag{3.7}$$

and the solution to the optimization problem in (3.3) is a (natural) cubic smoothing spline. Note that (3.7) penalizes spline curves g whose curvature varies rapidly. The shrinkage limit is a linear function of x, for which the curvature is null. The linear regression model assumes g to be a linear function of x. Thus, the penalty in (3.7) acts by penalizing spline curves that depart from this 'baseline' linear model. This motivation behind (3.7) is one of the reasons why cubic splines estimated with the penalty (3.7) are a popular choice in nonparametric regression problems.

For a more general class of regression models, like those studied in this thesis, some transformation of the mean response value is necessary before it can be written as a sum of smooth terms, or the mean may not be defined at all. In this case, the log-likelihood function plays the role of the residual sum of squares term in (3.3) to form the penalized log-likelihood criterion (3.1).

3.2.2 Double Penalty Models

Single penalty models may have limitations concerning the right specification of the limit of smoothness. Consider, for example, the penalty (3.7). Its value is invariant under constant and linear shifts of the spline function g. Certain applications, however, may require one to further penalize constant and linear functions of x. Hence, in some situations it may be convenient to have multiple penalties over different characteristics of the spline curve g.

Let

$$P_d(g\,;\lambda) = \lambda \int \left[g^{(d)}(x)\right]^2 dx, \quad \lambda \geq 0\,, \tag{3.8}$$

and consider the cases $d = 1, 2$, corresponding to $P_1(g\,;\lambda) = \lambda \int g'(x)^2\, dx$ and

$P_2(g\,;\lambda) = \lambda \int g''(x)^2\,dx$ respectively. A *double penalty functional* is defined as:

$$P_{12}(g\,;\lambda_1,\lambda_2) = P_1(g\,;\lambda_1)+P_2(g\,;\lambda_2) = \lambda_1 \int g'(x)^2\,dx+\lambda_2 \int g''(x)^2\,dx\,,\ \ \lambda_1,\,\lambda_2 \geq 0\,. \tag{3.9}$$

As in the single penalty case, the penalty $P_{12}(g\,;\lambda_1,\lambda_2)$ can be subtracted to some appropriate log-likelihood function, replacing $P(g\,;\lambda)$ in (3.1). As both λ_1 and λ_2 become large, the corresponding fit $\widehat{g}$ converges to a polynomial of degree zero, i.e., a constant function of x. Hence, $P_{12}(g\,;\lambda_1,\lambda_2)$ is useful in situations where $g(x)$ generalizes constant functions of x.

Double penalization has been studied almost exclusively in the context of P-splines as developed by Eilers & Marx (1996). Let $\{\gamma_m\}_{m=1}^{\mathcal{K}_B}$ be the coefficients of the B-spline representation in (2.5). Aldrin (2006) investigated penalized cubic splines obtained with the double penalty

$$\lambda_1 \sum_{m=2}^{\mathcal{K}_B} (\nabla\gamma_m)^2 + \lambda_2 \sum_{m=3}^{\mathcal{K}_B} \left(\nabla^2\gamma_m\right)^2\,,\ \ \lambda_1,\,\lambda_2 \geq 0\,, \tag{3.10}$$

in the context of additive Gaussian regression models. Aldrin found that the penalty in (3.10) led to estimated models with better predictive ability compared to those using the single penalty (3.4) for $o = 2$. Eilers & Marx (2003) followed a similar approach within penalized signal regression. They replaced the first term in the penalty (3.10) by a ridge penalty, $\delta \sum_m (\gamma_m)^2$, with δ fixed to some small positive value. In the same spirit, Eilers & Goeman (2004) studied a modified version of the penalty in (3.10) where the two smoothing parameters were related through a monotonic function. More precisely, they took $\lambda_2 = \lambda$ and $\lambda_1 = \sqrt{\lambda}$. The choice of the squared root transformation for the smoothing parameter associated with

the penalty based upon the first derivative can be more easily understood in the following context: consider the fit of a smooth series, z, to noisy data sampled at equal distances, h. If $\nabla z \approx hz'$, then $\nabla^2 z \approx h^2 z''$. Hence, $\sum(\nabla z)^2 \approx h^2 \int z'^2$ and $\sum(\nabla^2 z)^2 \approx h^4 \int z''^2$. Since the penalty in (3.10) approximates a weighted sum of $\int z'^2$ and $\int z''^2$, we would expect weights, say, w^2 and w^4. Taking $\lambda = w^4$ we reach the desired result. Eilers & Goeman (2004) showed that this double penalty produced good estimates in situations where the true curve consisted of an essentially flat baseline process and sharp pulses. A detailed technical treatment and study of multiple quadratic penalty models can be found in Wood (2000, 2004).

3.3 Penalized Likelihood and Bayesian Inference

The optimization problem defined by the criterion $\mathcal{J}(g)$ in (3.1) can be cast in a Bayesian framework whatever the penalty functional we choose to use. Recall that, for a log-likelihood function $\ell(g\,;D)$ and a general penalty functional $P(g\,;\lambda) = \lambda\, F(g)$,

$$\mathcal{J}(g) = \ell(g\,;D) - P(g\,;\lambda)\,, \quad \lambda \geq 0\,.$$

The criterion $\mathcal{J}(g)$ has a natural interpretation as the log-posterior for g. Take $P(g\,;\lambda)$ to be minus the log-prior for g with hyperparameter λ, i.e., $p\,(g \mid \lambda) \propto \exp\{-P(g\,;\lambda)\}$. Then the posterior log-density of g given D is, up to an additive function of D and λ, given by $\mathcal{J}(g)$. Thus, the maximum a posteriori (MAP) estimator is $\widehat{g}$, the solution to the optimization problem (3.1).

All the penalty functionals discussed so far are quadratic. Let $\boldsymbol{\gamma}$ be the vector containing the elements of the parametrization defining g. For example, $\boldsymbol{\gamma}$ can be

the set of $\mathcal{K}_B$ coefficients of the B-spline representation of g in (2.5). Since g is linear in $\boldsymbol{\gamma}$, quadratic penalties in g define quadratic forms in $\boldsymbol{\gamma}$. Thus, $P(g\,;\lambda)$ can be written as $P(g\,;\lambda) = \lambda\, F(g) = \lambda\, \boldsymbol{\gamma}^{\mathsf{T}} \boldsymbol{P}\, \boldsymbol{\gamma}$, for some symmetric, positive semi-definite matrix $\boldsymbol{P}$, called the *penalty matrix*. Hence, the prior for the curve g becomes a prior over $\boldsymbol{\gamma}$, i.e.,

$$p\,(\boldsymbol{\gamma} \mid \lambda) \propto \exp\left(-\lambda\, \boldsymbol{\gamma}^{\mathsf{T}} \boldsymbol{P}\, \boldsymbol{\gamma}\right). \tag{3.11}$$

The prior in (3.11) resembles a multivariate Gaussian prior distribution with mean $\mathbf{0}$ and covariance $[\lambda\, \boldsymbol{P}]^{+}$, where $[\boldsymbol{A}]^{+}$ denotes the generalized inverse of $\boldsymbol{A}$. Since the eigenvalues of $\boldsymbol{P}$ corresponding to curves g that yield a value of zero for $F(g)$ are themselves zero, direct inversion of $\boldsymbol{P}$ leads to $+\infty$ eigenvalues. This is the reason for using $[\lambda\, \boldsymbol{P}]^{+}$ as the prior covariance matrix. The prior density $p\,(\boldsymbol{\gamma} \mid \lambda)$ in (3.11) is said to be partially improper (Green, 1987), where impropriety of the prior is equivalent to rank deficiency in $\boldsymbol{P}$. Take, for example, the penalty in (3.7). The corresponding penalty matrix $\boldsymbol{P}$ will have two zero eigenvalues associated with constant and linear functions of x. Hence, the prior will assign infinite variance to parameter vectors $\boldsymbol{\gamma}$ defining constant and linear functions of x.

The properties of the aforementioned Bayesian model are studied in, for example, Silverman (1985). In the context of cubic smoothing splines, Wahba (1978, 1983) considers Bayesian inference on an infinite dimensional function space by defining the underlying function g to be, a priori, the sum of a random linear function and an integrated Wiener process.

3.4 Summary

The penalized log-likelihood criterion for spline smoothing has been presented. We studied not only the standard case, involving a single penalty functional, but also an extension where a double penalty was subtracted from the log-likelihood function of the model of interest. Double penalty models overcome the limitations of standard single penalty functionals in specifying an appropriate limit of smoothness. This chapter ended with a discussion on the Bayesian interpretation of the penalized log-likelihood criterion.

Chapter 4

The Value-First Derivative Parametrization

4.1 Introduction

Spline functions are typically represented using elements in the span of a chosen spline basis. In Chapter 2 we presented two common examples of basis functions, the truncated power basis and the B-spline basis. There we discussed the strengths and weaknesses of each of these parametrizations. On the one hand, truncated power functions are easily manipulated but they lack numerical stability. On the other hand, B-splines have good numerical properties but their analytical manipulation is cumbersome except in the special case of equally spaced knots. This chapter explores an intuitive parametrization whose elements are easy to interpret and relate naturally to the spline function. Moreover, the parametrization is local in its nature and straightforward to handle analytically given any configuration of the knots. In this chapter we shall see that penalty functionals based upon integrated squared

derivatives of the spline curve, like the penalties discussed in Chapter 3, have simple, interpretable expressions in terms of the components of the parametrization we consider here. We refer to such parametrization as *value-first derivative parametrization* (VFDP), as we feel it is more appealing in the curve smoothing context of the current thesis. In the numerical analysis literature, spline functions represented by this parametrization are known as cubic Hermite splines. Good reference textbooks are Lancaster & Šalkauskas (1986), or Burden & Faires (2004), for example.

4.2 Definition of the Parametrization

Let $g(x)$, $x \in [l, r]$, be the function we wish to estimate using a cubic spline with knots $\{k_m\}_{m=1}^{\mathcal{K}}$ that cover the domain of x. From the definition of spline functions given in Chapter 2, we know that g agrees with a cubic polynomial within each knot interval $[k_m, k_{m+1})$. Such a polynomial can be uniquely defined by four conditions over its coefficients. For each knot k_m we define

$$a_m = g(k_m)\,, \quad b_m = g'(k_m)\,. \tag{4.1}$$

The parameters a_m, b_m, a_{m+1} and b_{m+1} define, according to (4.1), four equations over the coefficients of the polynomial that agrees with g within the knot interval $[k_m, k_{m+1})$. This means that, for $k_m \leq x < k_{m+1}$, we can write $g(x)$ in terms of a_m, b_m, a_{m+1} and b_{m+1}. If the four cubic polynomials in x are

$$\begin{aligned} \phi_{0m}(x) &= \frac{(u_m - \Delta_m)^2(2u_m + \Delta_m)}{\Delta_m^3}\,, & \phi_{1m}(x) &= \frac{u_m^2(3\Delta_m - 2u_m)}{\Delta_m^3}\,, \\ \psi_{0m}(x) &= \frac{u_m(u_m - \Delta_m)^2}{\Delta_m^2}\,, & \psi_{1m}(x) &= \frac{u_m^2(u_m - \Delta_m)}{\Delta_m^2}\,, \end{aligned} \tag{4.2}$$

where $u_m = x - k_m$ and $\Delta_m = k_{m+1} - k_m$, then it is straightforward to show that

$$g(x) = a_m\phi_{0m}(x) + b_m\psi_{0m}(x) + a_{m+1}\phi_{1m}(x) + b_{m+1}\psi_{1m}(x), \quad x \in [k_m,\, k_{m+1})\,, \tag{4.3}$$

for $m = 1, \ldots, \mathcal{K}-1$. Hence, the spline curve g is completely specified by the 4-dimensional vector of parameters $\boldsymbol{\alpha}_m = (a_m, b_m, a_{m+1}, b_{m+1})^\intercal$ within the knot interval $[k_m,\, k_{m+1})$, $m = 1, \ldots, \mathcal{K}-1$, and by the $2\mathcal{K}$-dimensional vector of parameters $\boldsymbol{\alpha} = (a_1, b_1, \ldots, a_\mathcal{K}, b_\mathcal{K})^\intercal$ within $[k_1, k_\mathcal{K})$. Thus, the space of cubic spline functions with knots $\{k_m\}_{m=1}^{\mathcal{K}}$ that are continuous and have continuous first derivative is a $2\mathcal{K}$-dimensional space.

Note that the definition of a_m and b_m in (4.1) automatically imposes g and g' to be continuous everywhere. However, g'' may be discontinuous across the knots, which brings additional flexibility to the fitting process. This is not the case in the parametrization used by Green & Silverman (1994), where the estimated curve is constrained to have continuous curvature throughout its domain. Cubic splines for which the second derivative is allowed to be discontinuous are referred to as *deficient splines* (Dikshit & Powar, 1981; Rana & Purohit, 1988). For a detailed study of deficient splines and their approximation properties the reader is referred to Dubeau & Savoie (1999).

The VFDP is easy to interpret as the parameters are naturally connected to the spline function g. The parameter a_m represents the image of the curve g at the point k_m in its domain, while b_m defines the rate of increase or decrease of g at that point. Hence, the set of $\mathcal{K}$ values a_m and b_m provides a rough visual estimate of g. This is useful when presenting the results to non-experts.

A general expression for $g(x)$ is readily available from (4.3). Let $\boldsymbol{\eta}_m(x)$ be the

4-dimensional row vector with components the four polynomials in (4.2), i.e.,

$$\boldsymbol{\eta}_m(x) = (\phi_{0m}(x), \psi_{0m}(x), \phi_{1m}(x), \psi_{1m}(x)) \ .$$

The value of the spline function $g(x)$ is:

$$g(x) = \sum_{m=1}^{\mathcal{K}-1} \mathbb{I}_m(x)\, \boldsymbol{\eta}_m(x)\, \boldsymbol{\alpha}_m \ , \quad x \in [k_1, k_{\mathcal{K}}) \ , \tag{4.4}$$

where $\mathbb{I}_m(x)$ is the indicator function associated with the knot interval $[k_m,\, k_{m+1})$:

$$\mathbb{I}_m(x) = \begin{cases} 1, & \text{if } k_m \leq x < k_{m+1} \\ 0, & \text{otherwise} \end{cases} \ .$$

Within each knot interval $[k_m,\, k_{m+1})$ the spline curve g is a linear combination of the four polynomials in $\boldsymbol{\eta}_m(x)$. The expression in (4.4) enables g to be plotted to any degree of accuracy desired. A visual display of the polynomial components of the VFDP is given in Figure 4.1.

One of the advantages of spline models when compared to single polynomial ones is local influence. The VFDP highlights this property since each parameter in $\boldsymbol{\alpha}$ affects the fitted curve only in the span of two consecutive knot intervals. For example, the influence of a_m and b_m on the value of g is restricted to the interval $[k_{m-1}, k_{m+1})$. Any form of correlation among the parameters is thus likely to be small.

As a final remark, note that the estimation of derivatives becomes trivial if the spline function is parametrized according to the VFDP. Derivative estimation can be useful in, for example, mechanics, where one may be interested in velocity

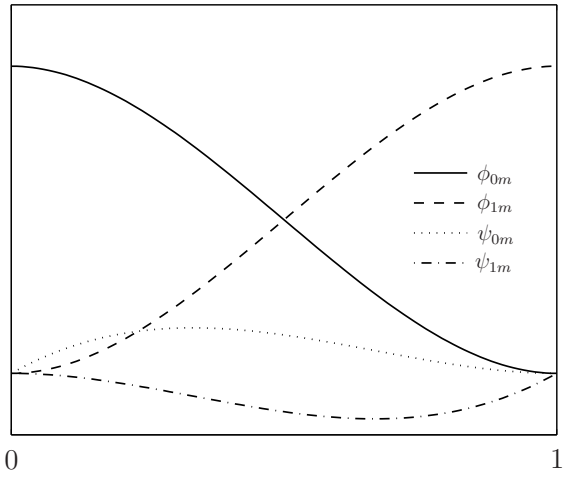

Figure 4.1: The four cubic polynomials in $\boldsymbol{\eta}_m$ for the knot interval $[k_m,\, k_{m+1}) = [0, 1)$.

estimates. A rough estimate of g' is provided by the elements $\{b_m\}_{m=1}^{\mathcal{K}}$ in the vector of parameters $\boldsymbol{\alpha}$ defining g. If a more precise estimate is required, then one only needs to differentiate the polynomials in $\boldsymbol{\eta}_m(x)$, $m = 1, \ldots, \mathcal{K}-1$, given in (4.2), and write

$$g'(x) = \sum_{m=1}^{\mathcal{K}-1} \mathbb{I}_m(x)\, \boldsymbol{\eta}'_m(x)\, \boldsymbol{\alpha}_m\,, \quad x \in [k_1, k_{\mathcal{K}})\ ,$$

where $\boldsymbol{\eta}'_m(x) = (\phi'_{0m}(x), \psi'_{0m}(x), \phi'_{1m}(x), \psi'_{1m}(x))$. The estimate of g'' will be a piecewise linear function with possible jumps at the interior knots $k_2, \ldots, k_{\mathcal{K}-1}$.

4.3 Penalty Implementation and Interpretation

In the discussion of the paper by Eilers & Marx (1996), Cox (1996) points out some of the drawbacks regarding the use of difference penalties. Cox (1996) notes that, even though difference penalties are computationally simpler to implement for a B-spline basis than the standard integral based penalties such as (3.8), they are not

as easy to interpret as the latter, specially regarding prior specification within a Bayesian framework. Parametrizing the curve g through the VFDP overcomes such drawbacks. As we shall see below, for $d = 1, 2$ the penalty in (3.8) can be written in a simple algebraic form owning to the local nature of the VFDP. In addition, we gain insight into the effect of the penalties on the estimated curve. This aids in the choice of an appropriate penalty functional, and thus of an appropriate prior distribution for the vector of spline parameters $\boldsymbol{\alpha}$, following the discussion in Section 3.3.

The VFDP differs from conventional parametrizations of cubic splines by defining g as a deficient spline and thus allowing g'' to be discontinuous at the interior knot points. However, the curvature of the spline function can still be used as a measure of its roughness, as defined by (3.8) for $d = 2$, by noting that the penalty functional $P_2(g\,;\lambda) = \lambda \int g''(x)^2\,dx$ can be re-expressed as

$$P_2(g\,;\lambda) = \lambda \sum_{m=1}^{\mathcal{K}-1} \int_{k_m}^{k_{m+1}} g''(x)^2\,dx = \sum_{m=1}^{\mathcal{K}-1} P_{2m}(g\,;\lambda)\,, \tag{4.5}$$

using the fact that $\{k_m\}_{m=1}^{\mathcal{K}}$ constitutes a partition of the domain of g, and that within each knot interval g'' is squared integrable. The local penalties $P_{2m}(g\,;\lambda)$ in (4.5) have simple, interpretable expressions as we shall see below.

For $m = 1, \ldots, \mathcal{K}-1$ we define

$$\begin{aligned} d_{m,m+1} &= a_{m+1} - (a_m + \Delta_m b_m)\,, \\ d_{m+1,m} &= a_m - (a_{m+1} - \Delta_m b_{m+1})\,. \end{aligned} \tag{4.6}$$

The quantities $d_{m,m+1}$ and $d_{m+1,m}$ are represented in Figure 4.2. The value of $d_{m,m+1}$ is the difference between the value of g at k_{m+1} and the tangent to g at k_m evaluated

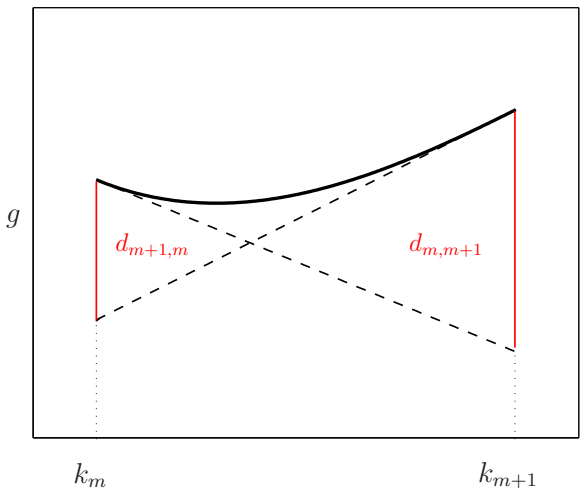

Figure 4.2: The quantities $d_{m,m+1}$ and $d_{m+1,m}$ for a cubic spline function g. The values of $d_{m,m+1}$ and $d_{m+1,m}$ are the lengths of the corresponding vertical red lines.

at k_{m+1}, and similar reasoning applies to $d_{m+1,m}$. It is also easy to see that there is a relationship between the degree of g within $[k_m,\, k_{m+1})$ and the values of $d_{m,m+1}$ and $d_{m+1,m}$: the curve g is a quadratic polynomial between k_m and k_{m+1} if and only if $d_{m,m+1} = d_{m+1,m}$; g is linear within $[k_m,\, k_{m+1})$ if and only if $d_{m,m+1} = d_{m+1,m} = 0$. A proof is given in Appendix A.

Shaw (1987) showed that we can write each functional $P_{2m}(g\,;\lambda)$ in (4.5) in terms of $d_{m,m+1}$ and $d_{m+1,m}$,

$$P_{2m}(g\,;\lambda) = \lambda\,\frac{3\,(d_{m,m+1} - d_{m+1,m})^2 + (d_{m,m+1} + d_{m+1,m})^2}{\Delta_m^3}\,, \quad m = 1, \ldots, \mathcal{K}{-}1\,. \tag{4.7}$$

The impact of the local penalty $P_{2m}(g\,;\lambda)$ on the portion of the spline curve g between $[k_m,\, k_{m+1})$ is now clear from (4.7). It penalizes generalizations of linear relationships, the strength of the penalization increasing as these generalizations

become more complex. Hence, linear polynomials yield a value of zero for $P_{2m}(g\,;\lambda)$. Parabolas are penalized only through the term $(d_{m,m+1}+d_{m+1,m})^2$. Cubic polynomials are fully penalized since both terms in the numerator of $P_{2m}(g\,;\lambda)$ are different from zero. How much these terms affect the estimated curve is determined by the value of the smoothing parameter λ as described in Section 3.2.

A similar decomposition holds for the penalty $P_1(g\,;\lambda)$ corresponding to $d=1$ in (3.8). Again we make use of the local structure of the VFDP and write

$$P_1(g\,;\lambda)=\lambda\sum_{m=1}^{\mathcal{K}-1}\int_{k_m}^{k_{m+1}} g'(x)^2\,dx=\sum_{m=1}^{\mathcal{K}-1}P_{1m}(g\,;\lambda)\,, \tag{4.8}$$

where

$$P_{1m}(g\,;\lambda)=\lambda\,\frac{(a_{m+1}-a_m)^2+\frac{1}{20}\,(d_{m,m+1}-d_{m+1,m})^2+\frac{1}{12}\,(d_{m,m+1}+d_{m+1,m})^2}{\Delta_m}\,. \tag{4.9}$$

The term $(a_{m+1}-a_m)^2$ penalizes linear functions of x. Thus, $P_{1m}(g\,;\lambda)$ is increasingly penalizing curves that grow in complexity compared to a constant function of x, $x\in[k_m,\,k_{m+1})$, for which $P_{1m}(g\,;\lambda)\equiv 0$.

Given $\boldsymbol{\alpha}$, evaluation of the $\mathcal{K}-1$ penalties $P_{2m}(g\,;\lambda)$ in (4.5) and $P_{1m}(g\,;\lambda)$ in (4.8) is straightforward using the expressions in (4.7) and (4.9) respectively, which are valid for any configuration of knots. The double penalty $P_{12}(g\,;\lambda_1,\lambda_2)$ in (3.9) is simply the sum of $P_1(g\,;\lambda_1)$ and $P_2(g\,;\lambda_2)$ above.

4.4 Computational Details

For concreteness, consider the regression model in (2.8) for the case $p = 1$:

$$Y = g(X) + \epsilon\,, \quad \epsilon \sim N(0, \sigma^2)\,.$$

Suppose we have observations $x_1, \ldots, x_n$ of the random variable X which defines the domain of g. Our aim is to find the parameter vector $\boldsymbol{\alpha}$ associated with the spline function g that maximizes the penalized log-likelihood criterion in (3.1). In order to characterize the solution we need some additional notation. We start by building the design matrix $\boldsymbol{X}$ associated with $x_1, \ldots, x_n$. Denote by $\boldsymbol{I}$ the $n \times (\mathcal{K}-1)$ incidence matrix whose ith row has zeros everywhere except for the column corresponding to the knot interval containing observation x_i, where it takes value 1. For each x_i we define the $(\mathcal{K}-1) \times (2\mathcal{K})$ matrix

$$\boldsymbol{\Omega}^i = \begin{pmatrix} \phi_{01}(x_i) & \psi_{01}(x_i) & \phi_{11}(x_i) & \psi_{11}(x_i) & 0 & 0 & 0 & \ldots \\ 0 & 0 & \phi_{02}(x_i) & \psi_{02}(x_i) & \phi_{12}(x_i) & \psi_{12}(x_i) & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where $\phi_{0m}(x_i)$, $\psi_{0m}(x_i)$, $\phi_{1m}(x_i)$ and $\psi_{1m}(x_i)$ are the polynomials in (4.2) evaluated at x_i. The ith row of the $n \times (2\mathcal{K})$ design matrix $\boldsymbol{X}$ is given by $\boldsymbol{I}_{i\bullet}\,\boldsymbol{\Omega}^i$, where $\boldsymbol{A}_{i\bullet}$ represents the ith row of the matrix $\boldsymbol{A}$. The vector of function evaluations $\boldsymbol{g} = (g(x_1), \ldots, g(x_n))^\intercal$ can be expressed as $\boldsymbol{X}\boldsymbol{\alpha}$, i.e., $g(x_i) = \boldsymbol{X}_{i\bullet}\,\boldsymbol{\alpha}$.

The local penalty $P_{2m}(g\,;\lambda)$ in (4.7) defines a quadratic form in the vector of parameters $\boldsymbol{\alpha}_m = (a_m, b_m, a_{m+1}, b_{m+1})^\intercal$, with coefficients the entries of the 4×4

symmetric, positive semi-definite matrix

$$\boldsymbol{P}_{2m} = \begin{pmatrix} \frac{12}{\Delta_m^3} & \frac{6}{\Delta_m^2} & -\frac{12}{\Delta_m^3} & \frac{6}{\Delta_m^2} \\ \frac{6}{\Delta_m^2} & \frac{4}{\Delta_m} & -\frac{6}{\Delta_m^2} & \frac{2}{\Delta_m} \\ -\frac{12}{\Delta_m^3} & -\frac{6}{\Delta_m^2} & \frac{12}{\Delta_m^3} & -\frac{6}{\Delta_m^2} \\ \frac{6}{\Delta_m^2} & \frac{2}{\Delta_m} & -\frac{6}{\Delta_m^2} & \frac{4}{\Delta_m} \end{pmatrix} .$$

Therefore, we can write:

$$P_{2m}(g\,;\lambda) = \lambda\, \boldsymbol{\alpha}_m{}^{\mathsf{T}} \boldsymbol{P}_{2m}\, \boldsymbol{\alpha}_m\,. \tag{4.10}$$

The two zero eigenvalues of $\boldsymbol{P}_{2m}$ are associated with constant and linear functions of x in $[k_m,\, k_{m+1})$, for which the corresponding parameter vector $\boldsymbol{\alpha}_m$ yields a value of zero for $P_{2m}(g\,;\lambda)$.

The matrix $\boldsymbol{P}_{2m}$ can be thought of as the *local penalty matrix* associated with the knot interval $[k_m,\, k_{m+1})$. The overall penalty matrix can be easily obtained from the set of the local matrices $\{\boldsymbol{P}_{2m}\}_{m=1}^{\mathcal{K}-1}$; we use the fact that each pair of parameters $\{a_m, b_m\}$ affects the spline g in the span of the knot intervals $[k_{m-1}, k_m)$ and $[k_m,\, k_{m+1})$. Hence, the overall penalty matrix, $\boldsymbol{P}_2$, will be the $(2\mathcal{K}) \times (2\mathcal{K})$

symmetric, positive semi-definite matrix:

$$\boldsymbol{P}_2 = \begin{pmatrix} \frac{12}{\Delta_1^3} & \frac{6}{\Delta_1^2} & -\frac{12}{\Delta_1^3} & \frac{6}{\Delta_1^2} & 0 & 0 & \dots \\ \frac{6}{\Delta_1^2} & \frac{4}{\Delta_1} & -\frac{6}{\Delta_1^2} & \frac{2}{\Delta_1} & 0 & 0 & \dots \\ -\frac{12}{\Delta_1^3} & -\frac{6}{\Delta_1^2} & \frac{12}{\Delta_1^3} + \frac{12}{\Delta_2^3} & -\frac{6}{\Delta_1^2} + \frac{6}{\Delta_2^2} & -\frac{12}{\Delta_2^3} & \frac{6}{\Delta_2^2} & \dots \\ \frac{6}{\Delta_1^2} & \frac{2}{\Delta_1} & -\frac{6}{\Delta_1^2} + \frac{6}{\Delta_2^2} & \frac{4}{\Delta_1} + \frac{4}{\Delta_2} & -\frac{6}{\Delta_2^2} & \frac{2}{\Delta_2} & \dots \\ 0 & 0 & -\frac{12}{\Delta_2^3} & -\frac{6}{\Delta_2^2} & \frac{12}{\Delta_2^3} + \frac{12}{\Delta_3^3} & -\frac{6}{\Delta_2^2} + \frac{6}{\Delta_3^2} & \dots \\ 0 & 0 & \frac{6}{\Delta_2^2} & \frac{2}{\Delta_2} & -\frac{6}{\Delta_2^2} + \frac{6}{\Delta_3^2} & \frac{4}{\Delta_2} + \frac{4}{\Delta_3} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \quad (4.11)$$

The entries of $\boldsymbol{P}_2$ are the coefficients of the quadratic form over $\boldsymbol{\alpha}$ defined by the sum of the local penalties $P_{2m}(g\,;\lambda)$ in (4.5), i.e.,

$$P_2(g\,;\lambda) = \sum_{m=1}^{\mathcal{K}-1} P_{2m}(g\,;\lambda) = \lambda\,\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{P}_2\,\boldsymbol{\alpha}, \quad (4.12)$$

The matrix $\boldsymbol{P}_2$ has rank $2\mathcal{K}-2$. The two zero eigenvalues correspond to linear and constant functions of x.

The $(2\mathcal{K}) \times (2\mathcal{K})$ matrix $\boldsymbol{P}_1$ associated with the quadratic form defined by the penalty functional $P_1(g\,;\lambda)$ in (4.8) can also be easily derived using the $\mathcal{K}-1$ expressions in (4.9). It will be a symmetric positive semi-definite matrix with rank $2\mathcal{K}-1$. The unique zero eigenvalue corresponds to constant functions of x. The penalty matrix for $P_{12}(g\,;\lambda_1,\lambda_2)$ in (3.9) follows directly from $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$.

4.5 Summary

This chapter presented a simple and intuitive parametrization for cubic spline functions, called value-first derivative parametrization (VFDP). Here we have developed the basic setup of the VFDP and have shown how standard integral based penalty functionals can be easily expressed in terms of the components of the VFDP. One interesting consequence of such representation is that the VFDP provided insight into the different levels of shrinkage applied by the penalty functional upon the spline curve.

Chapter 5

Simulation Study

5.1 Introduction

This chapter presents a small simulation study designed to compare the performance of penalized cubic spline models based upon the proposed methodology, the VFDP with integral penalties (Chapter 4), with the standard B-spline and difference penalties approach of Eilers & Marx (1996). We aim to compare the performance of four different estimators in terms of their value of mean squared error. The estimators are: VFDP splines with single and double penalty functionals, as described in Section 4.3, and B-splines with single and double difference penalties, as described in Aldrin (2006).

The framework for the simulation study will be that of the additive model introduced in Section 2.3. Given n observations on a response variable Y, denoted by $\boldsymbol{y} = (y_1, \ldots, y_n)^\intercal$, and on p selected covariates $X_1, \ldots, X_p$, denoted by $\boldsymbol{x}_j = (x_{1j}, \ldots, x_{nj})^\intercal$, $j = 1, \ldots, p$, consider the additive model

$$y_i = \beta_0 + \sum_{j=1}^{p} g_j(x_{ij}) + \epsilon_i \,, \quad \epsilon_i \sim N(0, \sigma^2) \,, \tag{5.1}$$

where the components $g_1, \ldots, g_p$ are unknown univariate smooth functions of the covariates, to be estimated by penalized cubic spline functions with knots $\{k_{jm}\}_{m=1}^{\mathcal{K}_j}$, $j = 1, \ldots, p$ respectively. Consider the parametrization of such cubic spline functions according to the VFDP as described in Chapter 4. We can write $g_j(x_{ij}) = \boldsymbol{X}_{j,i\boldsymbol{.}}\,\boldsymbol{\alpha}_j$, with $\boldsymbol{\alpha}_j$ the $(2\mathcal{K}_j)$-dimensional vector of spline parameters defined in Section 4.2, and $\boldsymbol{X}_j$ the $n \times (2\mathcal{K}_j)$ design matrix in Section 4.4, and re-express (5.1) in terms of $\boldsymbol{\alpha}_j$ as

$$y_i = \beta_0 + \sum_{j=1}^{p} \boldsymbol{X}_{j,i\boldsymbol{.}}\,\boldsymbol{\alpha}_j + \epsilon_i\,. \tag{5.2}$$

The identifiability constraints regarding the smooth terms in (5.1), discussed in Section 2.3, can be easily incorporated into the estimation process. The constraint that each smooth term should sum to zero over its observed covariate values can be translated into matrix form as $\boldsymbol{C}_j\boldsymbol{\alpha}_j = 0$, with $\boldsymbol{C}_j = \mathbf{1}^\intercal\boldsymbol{X}_j$, and $\mathbf{1}$ a $n \times 1$ vector of 1's. Let $\boldsymbol{C}_j^\intercal = \boldsymbol{Q}_j\boldsymbol{R}_j$ be the QR decomposition of $\boldsymbol{C}_j^\intercal$. The rightmost $(2\mathcal{K}_j) - 1$ columns of $\boldsymbol{Q}_j$ give the null space of $\boldsymbol{C}_j$, $\boldsymbol{Z}_j$ say. If $\boldsymbol{P}^j$, say, is the penalty matrix in the model associated with the smooth term g_j, then writing $\boldsymbol{\alpha}_j^{\boldsymbol{Z}_j} = \boldsymbol{Z}_j\boldsymbol{\alpha}_j$, $\boldsymbol{X}_j^{\boldsymbol{Z}_j} = \boldsymbol{X}_j\boldsymbol{Z}_j$, and $\boldsymbol{P}_{\boldsymbol{Z}_j}^j = \boldsymbol{Z}_j^\intercal\boldsymbol{P}^j\boldsymbol{Z}_j$, ensures that the constraints are met. For clarity of presentation, we shall drop the dependence upon $\boldsymbol{Z}_j$ and simply write $\boldsymbol{\alpha}_j$, $\boldsymbol{X}_j$ and $\boldsymbol{P}^j$.

This chapter starts with the description of the Bayesian inference methodology for estimation of the parameters β_0 and $\boldsymbol{\alpha}_j$ in the additive regression model (5.2). The prior specification follows from the discussion in Section 3.3. Because the posterior distribution of the model is not analytically tractable, Bayesian inference relies upon Markov chain Monte Carlo (MCMC) algorithms.

The simulation results are presented at the end of this chapter together with the

resultant conclusions.

5.2 Bayesian Inference via MCMC

In Chapter 3 we discussed the intrinsic link between the penalized log-likelihood criterion in (3.1) and Bayesian inference. Here we shall see that the Bayesian framework is particularly useful in the context of smoothing problems as the degree of smoothness can be estimated jointly with the spline parameters. The inference process for single penalty models is described first. The presence of the two smoothing parameters requires a different inference strategy to be developed for double penalty models.

5.2.1 Single Penalty Models

Section 3.3 pointed out the existence of a relationship between penalty functionals and roughness priors for the spline parameters. Given that all the penalties discussed so far can be written as quadratic forms in $\boldsymbol{\alpha}_j$, an appropriate choice is a Gaussian type prior for $\boldsymbol{\alpha}_j$. For the single penalty model based upon $P_2(g_j\,;\lambda_j)$ (corresponding to $d = 2$ in (3.8)) the prior is

$$p\left(\boldsymbol{\alpha}_j \mid \lambda_j\right) \propto \lambda_j^{\mathrm{rk}(\boldsymbol{P}_2^j)/2} \exp\left(-\frac{\lambda_j}{2}\,\boldsymbol{\alpha}_j^{\intercal}\boldsymbol{P}_2^j\,\boldsymbol{\alpha}_j\right), \tag{5.3}$$

with $\boldsymbol{P}_2^j$ given by (4.11). Note that (5.3) defines a partially improper prior as $\boldsymbol{P}_2^j$ is rank deficient (Green, 1987). The intercept term β_0 in (5.2) is assigned a diffuse prior, i.e., $\beta_0 \propto const.$

For the hyperparameter λ_j a gamma prior (the conjugate prior of (5.3)) is as-

sumed, i.e., $\lambda_j \sim G(s_j, r_j)$. The constants s_j and r_j are usually chosen so that the prior is vague, and therefore expresses our ignorance regarding the shape of g_j, while yielding a proper posterior distribution. Typical choices include $r_j = 10^{-4}$, 10^{-5}, or 10^{-6}, with $s_j = 1$ or $s_j = r_j$.

Given observed data D, let $L(\beta_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p ; D)$ be the likelihood function of the additive model in (5.2). The joint posterior distribution is given by

$$p(\boldsymbol{\theta} \mid D) \propto L(\beta_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p ; D) \prod_j \lambda_j^{\mathrm{rk}(\boldsymbol{P}_2^j)/2} \exp\left(-\frac{\lambda_j}{2} \boldsymbol{\alpha}_j^\mathsf{T} \boldsymbol{P}_2^j \boldsymbol{\alpha}_j \right) \times \prod_j \lambda_j^{s_j - 1} \exp(-r_j \lambda_j), \quad (5.4)$$

where $\boldsymbol{\theta} = (\beta_0, \boldsymbol{\alpha}_1, \lambda_1, \ldots, \boldsymbol{\alpha}_p, \lambda_p)^\mathsf{T}$ is the vector of all the parameters in the model. The posterior density in (5.4) is analytically intractable. Therefore, inference relies upon Monte Carlo estimates obtained using MCMC simulation techniques based upon draws from full conditionals of blocks of parameters given the other parameters in the data. The structure of the blocks follows directly from that of the joint posterior in (5.4). The idea for block moves is that the corresponding likelihoods will contain more information, leading to less correlation and better convergence.

Straightforward calculations show that the full conditionals for $\boldsymbol{\alpha}_j$, $j = 1, \ldots, p$, are multivariate Gaussian with covariance matrix and mean

$$\boldsymbol{\Sigma}_j = \left[\frac{1}{\sigma^2} \boldsymbol{X}_j^\mathsf{T} \boldsymbol{X}_j + \lambda_j \boldsymbol{P}_2^j \right]^{-1}, \qquad \boldsymbol{m}_j = \boldsymbol{\Sigma}_j \frac{1}{\sigma^2} \boldsymbol{X}_j^\mathsf{T} \left(\boldsymbol{y} - \beta_0 - \sum_{l \neq j} \boldsymbol{X}_l \boldsymbol{\alpha}_l \right).$$

Likewise, a new value β_0 is drawn from the Gaussian distribution with parameters

$$\Sigma_{\beta_0} = \sigma^2 \, [\mathbf{1}^\intercal \, \mathbf{1}]^{-1} \, , \qquad m_{\beta_0} = [\mathbf{1}^\intercal \, \mathbf{1}]^{-1} \, \mathbf{1}^\intercal \left(\boldsymbol{y} - \sum_{j=1}^{p} \boldsymbol{X}_j \, \boldsymbol{\alpha}_j \right) , \tag{5.5}$$

where $\mathbf{1}$ is a $n \times 1$ vector of 1's. For the semiparametric model in (2.9), the intercept β_0 is embedded in the vector of parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_q)^\intercal$. The parameters of the full conditional for $\boldsymbol{\beta}$ are similar to those in (5.5), with $\mathbf{1}$ replaced by the design matrix $\boldsymbol{V}$ with ith row $\boldsymbol{V}_{i\bullet} = (1, v_{i1}, \ldots, v_{iq})$, where v_{il} is the observed value of the covariate V_l for individual i.

Once $\boldsymbol{\alpha}_j$ has been updated, a new value for the hyperparameter λ_j is obtained by sampling from its full conditional, a gamma distribution with parameters

$$\check{s}_j = s_j + \frac{\mathrm{rk}(\boldsymbol{P}_2^j)}{2} \, , \quad \check{r}_j = r_j + \frac{1}{2} \, \boldsymbol{\alpha}_j^\intercal \, \boldsymbol{P}_2^j \, \boldsymbol{\alpha}_j \, . \tag{5.6}$$

Since all the full conditionals above are known distributions, a Gibbs sampler (Geman & Geman, 1984) can be used to successively update the parameters of the model (see Appendix B for a description of the Gibbs sampler).

5.2.2 Double Penalty Models

Double penalty models specify the following form for the penalty functional:

$$P_{12}(g_j \, ; \lambda_j^1, \lambda_j^2) = \lambda_j^1 \int g_j'(x)^2 dx + \lambda_j^2 \int g_j''(x)^2 dx \, , \quad \lambda_j^1, \, \lambda_j^2 \geq 0 \, .$$

Hence, the prior density for the spline parameters $\boldsymbol{\alpha}_j$ is now defined as

$$p\left(\boldsymbol{\alpha}_j \mid \lambda_j^1, \lambda_j^2\right) \propto \exp\left(-\frac{1}{2}\,\boldsymbol{\alpha}_j^{\intercal}\left[\lambda_j^1 \boldsymbol{P}_1 + \lambda_j^2 \boldsymbol{P}_2\right]\boldsymbol{\alpha}_j\right).$$

The hyperparameters λ_j^1 and λ_j^2 deserve special attention here. Since they are associated with the same spline function, their values are likely to be correlated, making independence a priori an implausible assumption. However, it is not clear how one should elicit a joint prior for λ_j^1 and λ_j^2. We therefore resort to empirical Bayes methods as proposed in Ruppert & Carroll (2000). The main idea behind empirical Bayes techniques is to estimate hyperparameters in a prior using the data at hand and then to plug-in those estimates in the prior as though they were known.

Let $\boldsymbol{\lambda}_j = \left\{\lambda_j^1, \lambda_j^2\right\}$, $j = 1, \ldots, p$. In what follows we shall make explicit the dependence upon the value of the smoothing parameter pair $\boldsymbol{\lambda}_j$. We use Akaike's information criterion (AIC) to select the smoothing parameters $\boldsymbol{\lambda}_j$,

$$\mathrm{AIC}\left(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_p\right) = -2\left[\ell(\beta_0, \boldsymbol{\alpha}_1(\boldsymbol{\lambda}_1), \ldots, \boldsymbol{\alpha}_p(\boldsymbol{\lambda}_p)\,; D) + \mathrm{tr}(\boldsymbol{R}(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_p))\right], \quad (5.7)$$

where ℓ is the log-likelihood function and $\boldsymbol{R}$ is the $n \times n$ smoother (or hat) matrix of the additive model in (5.2), whose trace provides an estimate of the total number of degrees of freedom in the model (Hastie & Tibshirani, 1990b). We estimate the parameters $(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_p)$ by

$$(\widehat{\boldsymbol{\lambda}}_1, \ldots, \widehat{\boldsymbol{\lambda}}_p) = \arg\min \mathrm{AIC}\left(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_p\right).$$

Simultaneous minimization of $\mathrm{AIC}\left(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_p\right)$ is very intensive computationally. We propose to overcome this by using an adaptive algorithm that estimates each

$\boldsymbol{\lambda}_j$ individually by minimizing the global criterion $\text{AIC}\,(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_p)$. In practice, the smoothing parameters in $\boldsymbol{\lambda}_j$ are varied over a pre-specified grid. For each pair $\boldsymbol{\lambda}_j$ we estimate $\boldsymbol{\alpha}_j$ using *backfitting* (Friedman & Stuetzle, 1981; Hastie & Tibshirani, 1986). We then select $\widehat{\boldsymbol{\lambda}}_j$ that minimizes $\text{AIC}\,(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_p)$. Each iteration of the algorithm updates all p smoothing parameter pairs $\boldsymbol{\lambda}_j$, $j = 1, \ldots, p$. The algorithm stops when $\text{AIC}\,(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_p)$ converges. The foregoing iterative scheme resembles an algorithm developed by Hastie & Tibshirani (1990b), called *BRUTO*, which combines backfitting and smoothing parameter selection. A similar idea is followed by Ruppert & Carroll (2000) in the context of adaptive smoothing.

The iterative AIC algorithm described above provided good results as we shall see in later chapters of this thesis. More efficient methods that avoid the use of backfitting, such as those in Wood (2004), could also be employed.

The matrix $\boldsymbol{R}$ in (5.7) is the additive fit operator that produces $\widehat{\boldsymbol{g}}_+ = \boldsymbol{R}\boldsymbol{y}$ at convergence of the backfitting algorithm, where $\boldsymbol{g}_+ = \sum_j \boldsymbol{g}_j$, with $\boldsymbol{g}_j = \boldsymbol{X}_j\boldsymbol{\alpha}_j$. As Hastie & Tibshirani (1990b) point out, estimation of $\boldsymbol{R}$ is formidable except in very special cases (e.g., when a single smooth term exists). Hastie & Tibshirani (1990b) report on the good properties of the following approximation to $\text{tr}(\boldsymbol{R})$:

$$\text{tr}\,(\boldsymbol{R}) \approx 1 + \sum_{j=1}^{p} \left[\text{tr}(\widetilde{\boldsymbol{S}}_j) - 1\right], \tag{5.8}$$

where $\widetilde{\boldsymbol{S}}_j = \boldsymbol{X}_j \left[\frac{1}{\sigma^2}\boldsymbol{X}_j^\top\boldsymbol{X}_j + \lambda_j^1\boldsymbol{P}_1^j + \lambda_j^2\boldsymbol{P}_2^j\right]^{-1} \frac{1}{\sigma^2}\boldsymbol{X}_j^\top$ is the smoother matrix associated with the fitted curve $\boldsymbol{g}_j$. We therefore use the approximation in (5.8) in the AIC criterion of (5.7).

Given $\widehat{\boldsymbol{\lambda}}_1, \ldots, \widehat{\boldsymbol{\lambda}}_p$, Kneib & Fahrmeir (2007) use a Newton-Raphson type algorithm to obtain posterior mode estimates of $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p$ in an empirical Bayes frame-

work in the context of hazard regression models. Here we follow the hybrid approach in Fahrmeir et al. (2004), who combine empirical Bayes estimates of the smoothing parameters with estimates for the spline parameters obtained from MCMC outputs. This allows straightforward access to functionals of posterior estimates. Hence, given $\widehat{\boldsymbol{\lambda}}_1, \ldots, \widehat{\boldsymbol{\lambda}}_p$, estimation of $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p$ proceeds as in the single penalty case described in Subsection 5.2.1. The joint posterior distribution of the model is now

$$p\left(\beta_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p \mid D, \widehat{\boldsymbol{\lambda}}_1, \ldots, \widehat{\boldsymbol{\lambda}}_p\right) \propto L(\beta_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p\,; D) \times \prod_j \exp\left(-\frac{1}{2}\,\boldsymbol{\alpha}_j^{\intercal}\left[\widehat{\lambda}_j^1 \boldsymbol{P}_1 + \widehat{\lambda}_j^2 \boldsymbol{P}_2\right]\boldsymbol{\alpha}_j\right),$$

and the variance matrix of the full conditional distribution for $\boldsymbol{\alpha}_j$ has the form

$$\boldsymbol{\Sigma}_j = \left[\frac{1}{\sigma^2}\,\boldsymbol{X}_j^{\intercal}\boldsymbol{X}_j + \widehat{\lambda}_j^1\,\boldsymbol{P}_1 + \widehat{\lambda}_j^2\,\boldsymbol{P}_2\right]^{-1}.$$

5.3 Simulation Results

In this section we present the results of the simulation study we conducted in order to compare the four estimators described in Section 5.1. We started by considering two functions, $g_1(x) = \sin(2x) + 2\exp(-16x^2)$, from Denison et al. (1998a), and $g_2(x) = \frac{1}{0.72}\sin(x)$, from Lang & Brezger (2004), with $-2 \leq x \leq 2$. These are functions with high (g_1) and moderate (g_2) level of curvature. To create simulated data we rescaled both functions so that their support was the unit interval, and then evaluated them at 200 points generated uniformly in $[0, 1]$. Zero-mean Gaussian noise was added with the error variance σ^2 taking the values $\sigma = 0.3,\ 1$, corresponding to medium and low signal-to-noise ratio respectively.

We estimated g_1 and g_2 using cubic spline functions with 20 equally spaced knots, parametrized according to both the VFDP and the B-splines representation. Single penalty estimates were based upon the penalty $P_2(g_j\,;\lambda_j)$, and double penalty estimates upon $P_{12}(g_j\,;\lambda_j^1,\lambda_j^2)$. For the B-splines approach these penalties were replaced by the corresponding difference penalties (3.4), with $o = 2$, and (3.10) respectively. All computations were carried out on a Pentium IV 3.00 Ghz PC running Windows XP and MATLAB 7.0.1 (The MathWorks, 2008).

For each combination of single/double penalty and function/variance we simulated 200 replications. For single penalty estimates we used $s_j = 1$ and $r_j = 10^{-4}$ in the prior for λ_j, though a sensitive analysis showed no significant dependence of the results on the values of s_j and r_j. For each replicate, posterior mean estimates were computed based upon the output of a chain of length 200,000 (after an initial burn-in period of length 2,000) obtained with the sampling schemes of Subsection 5.2.1 and Subsection 5.2.2. Convergence of a chain is determined by examining the plot of the its path.

Bayesian inference for penalized splines represented through a B-spline basis as proposed by Eilers & Marx (1996) is similar. Details on prior specifications and the Gibbs sampler can be found in Lang & Brezger (2004). The B-spline basis is computed using the algorithm which is described in the Appendix in Eilers & Marx (1996). In what follows, 'SP' stands for single penalty, while 'DP' stands for double penalty.

The quality of the fit was measured by the logarithm of the empirical mean

squared error (MSE) given by:

$$\log_{10} \mathrm{RSS} = \log_{10} \left\{ \sum_{i=1}^{n} (g(x_i) - \widehat{g}(x_i))^2 \right\},$$

where g is the true function, and $\widehat{g}$ is the estimate of the true function, with lower values of $\log_{10}$ RSS indicating a better performance.

Figure 5.1 displays the spline estimates of the curve g_1 (Function 1), for the case $\sigma = 0.3$, together with the true function. The plot in Figure 5.1 suggests

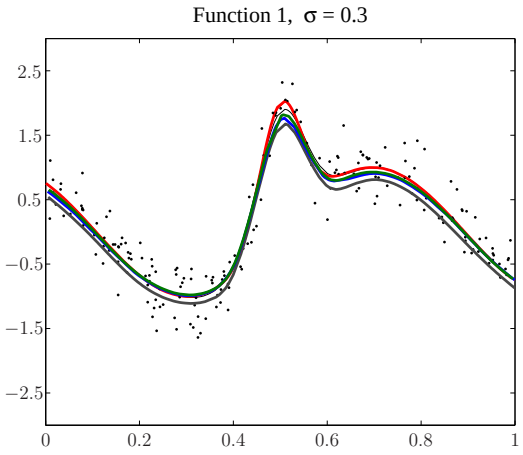

Figure 5.1: Estimated splines for g_1 with $\sigma = 0.3$; VFDP/SP (red), B-splines/SP (grey), VFDP/DP (blue), B-splines/DP (green), true curve (black); the dots represent a typical data set.

that the four methods perform equally well. The boxplots in Figure 5.2 show the simulations results corresponding to the function g_1 in both signal-to-noise ratio case scenarios. They corroborate the findings in Figure 5.1, namely that the four estimators considered here perform equally well in terms of their MSE value.

Figure 5.3 and Figure 5.4 contain the simulation results for the curve g_2 (Function 2). The four spline estimates represented in Figure 5.3 are almost undistinguishable

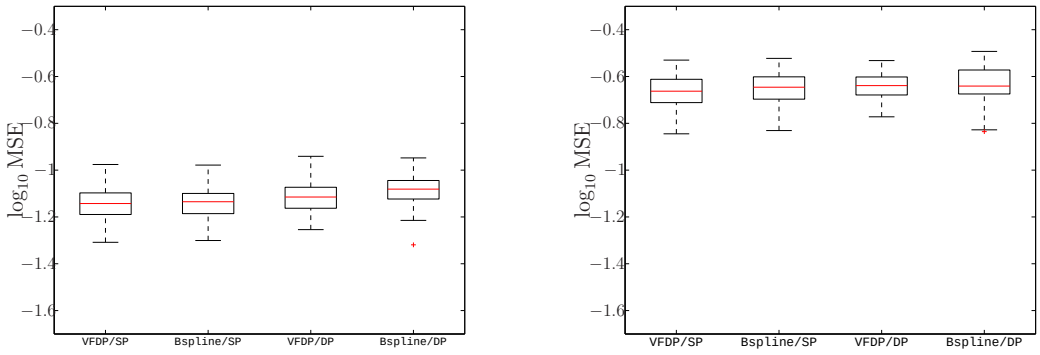

Figure 5.2: Boxplots of $\log_{10}$ *MSE for the four estimators of function* g_1*. The left panel corresponds to medium signal-to-noise ratio (*$\sigma = 0.3$*) and the right panel to low signal-to-noise ratio (*$\sigma = 1$*). From left to right the boxplots in the respective graphs refer to* VFDP *splines with single penalty, Bayesian P-splines with single penalty,* VFDP *splines with double penalty, and Bayesian P-splines with double penalty.*

and very close to the true function g_2, also displayed in Figure 5.3. However, the boxplots represented in the left panel of Figure 5.4 suggest that Bayesian P-splines combined with a double penalty functional perform slightly worst compared with the remaining three estimators.

We also considered the more demanding situation of trying to estimate a highly oscillating function. For this purpose we considered the following function (Function 3)

$$g_3(x) = \sqrt{x(1-x)}\,\sin\left(\frac{2\pi(1+2^{(9-4j)/5})}{x+2^{(9-4j)/5}}\right),$$

for $j = 3$ (low spatial variability) and $j = 5$ (high spatial variability). We mainly refer to Lang & Brezger (2004), who compared a variety of estimators using the function g_3 above. We take $\mathcal{K} = 40$ and $\sigma = 0.3$ as in Lang & Brezger (2004). The results for the 200 replicates can be found in Figure 5.5. They suggest that all estimators, apart from B-splines with a double penalty, perform equally well in

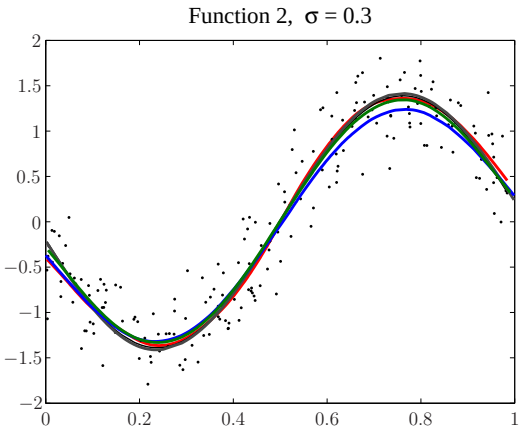

Figure 5.3: Estimated splines for g_2 with $\sigma = 0.3$; VFDP/SP (red), B-splines/SP (grey), VFDP/DP (blue), B-splines/DP (green), true curve (black); the dots represent a typical data set.

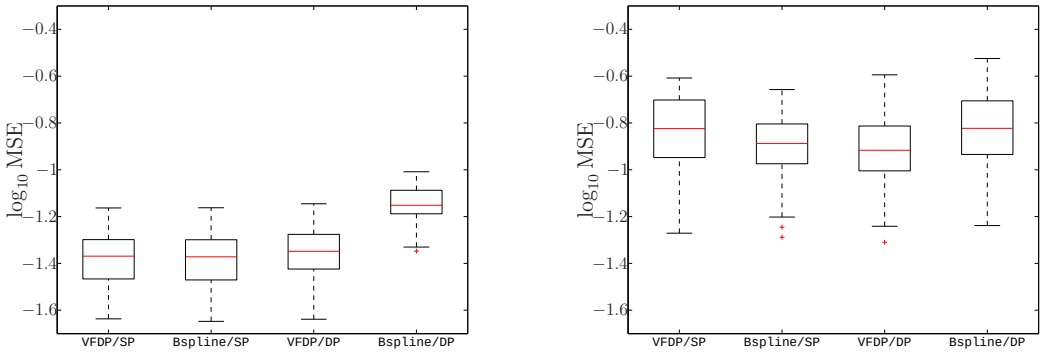

Figure 5.4: Boxplots of $\log_{10}$ *MSE for the four estimators of function g_2. The left panel corresponds to medium signal-to-noise ratio ($\sigma = 0.3$) and the right panel to low signal-to-noise ratio ($\sigma = 1$). From left to right the boxplots in the respective graphs refer to* VFDP *splines with single penalty, Bayesian P-splines with single penalty,* VFDP *splines with double penalty, and Bayesian P-splines with double penalty.*

terms of their value of MSE, both for the low and high spatial variability scenario. The estimated spline functions on the left (top and bottom panels) are practically undistinguishable from each other for the four estimators. Nevertheless, VFDP

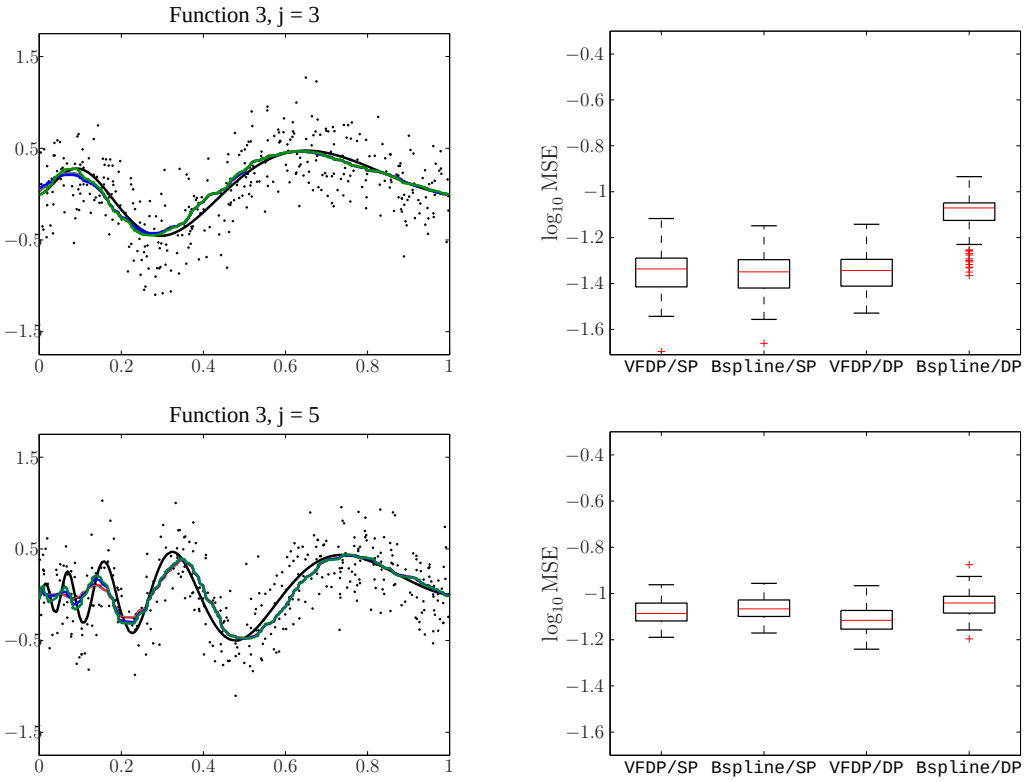

Figure 5.5: Simulation results for the function g_3 with $j = 3$ (top panel) and $j = 5$ (bottom panel). The plots on the left represent estimated splines together with a typical data set (dotted points); VFDP/SP (red), B-splines/SP (grey), VFDP/DP (blue), B-splines/DP (green), true curve (black);. The boxplots on the right display the values of MSE for the four estimators under study; from left to right the boxplots in the respective graphs refer to VFDP *splines with single penalty, Bayesian P-splines with single penalty,* VFDP *splines with double penalty, and Bayesian P-splines with double penalty.*

splines with a double penalty functional seem to have a slight advantage in the case $j = 5$. In contrast, B-splines combined with a double difference penalty showed a poor performance when $j = 3$, which agrees with the results found for the function g_2 when $\sigma = 0.3$. This is not surprising given that, for $j = 3$, the functions g_3 and g_2 exhibit similar behaviour.

5.4 Summary

The Bayesian additive model has been described in detail. Inference schemes for both single and double penalty models have been described. The results from a simulation study comparing the VFDP and the B-spline basis representation were presented. We also compared single and double penalty estimates. Overall, the VFDP was competitive with the B-splines approach, particularly if a double penalty functional was used.

Chapter 6

The VFDP in Generalized Additive Models

6.1 Introduction

In additive models the response variable is assumed to have a normal distribution with mean given by the sum of smooth functions of the predictors. Generalized additive models (Hastie & Tibshirani, 1986, 1990a) extend additive models in two directions: i) the response variable has distribution other than normal, and ii) the relationship between the mean response value and the predictors need not be linear. This chapter illustrates how generalized additive models (GAMs) can be represented using (cubic) penalized splines parametrized through the VFDP and estimated within a Bayesian framework.

GAMs and generalized linear models (Nelder & Wedderburn, 1972) share essentially the same theoretical background, as we shall see in the next section. Hence, we start this chapter by laying down the basic ingredients associated with generalized linear models (GLMs). We then review the local scoring algorithm for GAMs. The

algorithm is derived as a tool for maximizing the penalized log-likelihood criterion of Chapter 3. A stochastic version of the local scoring algorithm is discussed. Building on an idea first introduced by Gamerman (1997), Brezger & Lang (2006) propose an efficient Markov chain Monte Carlo (MCMC) sampling algorithm based on local scoring type proposals. Here we follow the same approach to perform Bayesian inference. This chapter concludes with an application of the proposed methodology to the analysis of a data set on union membership. The book by Hastie & Tibshirani (1990b) is the standard reference for GAMs. Wood (2006a) provides a comprehensive treatment of GAMs with several examples of applications.

6.2 From GLMs to GAMs - Concepts and Definitions

In a generalized linear model (GLM) the response variable Y is assumed to have a distribution from the *exponential family of distributions*, i.e., Y has probability density function, or probability mass function, of the form

$$f_Y(y) = \exp\{[y\theta - b(\theta)]/\zeta + c(y,\zeta)\},$$

where ζ is a *scale parameter* and θ is the *natural parameter*. The specific form of the distribution is determined by the functions b and c. Many well-known distributions belong to the exponential family, for example, the Bernoulli distribution, commonly associated with logistic regression. In this case, $b(\theta) = \log\left(1 + e^{\theta}\right)$, $\zeta = 1$, and $c(y,\zeta) \equiv 0$.

Let $\mathbb{E}[Y] = \mu$. Straightforward calculations show that $\mu = b'(\theta)$ and that $V(Y) = b''(\theta)\,\zeta$. Consider the set of predictors $X_1, \ldots, X_p$. In a GLM it is assumed that there

exists a transformation of μ such that

$$h(\mu) = \eta = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \,. \tag{6.1}$$

The function h is monotone and differentiable and is usually called the *link function*. It maps the domain of μ to the set inhabited by the *linear predictor* η in (6.1), usually the real line. One particular choice for h, which simplifies the algebra, is the canonical link function, which corresponds to $h(\cdot) = b'^{-1}(\cdot)$, i.e., $\eta = \theta$. As an example, for the Bernoulli distribution, $\mu = \Pr(Y = 1)$ is the mean and the canonical link is the *logit*, i.e., $h(\mu) = \text{logit}(\mu) \equiv \log\{\mu/(1-\mu)\}$. Estimation of μ does not involve ζ, so for simplicity this is assumed known and equal to one.

Generalized additive models (GAMs) extend GLMs in the same way as additive models extend linear models. The linear predictor $\eta = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$ specifies that $X_1, \ldots, X_p$ act linearly on the transformed mean $h(\mu)$. A GAM differs from a GLM in that the linear predictor η takes the form

$$\eta = \beta_0 + \sum_{j=1}^{p} g_j(X_j) \,, \tag{6.2}$$

where g_j, $j = 1, \ldots, p$, are univariate smooth functions. These functions will not be given a parametric form but instead will be estimated in a nonparametric fashion.

GAMs retain the additivity property of the familiar multiple linear regression model in (2.6), and so they can be interpreted 'variable-by-variable'. Holding all other component functions in (6.2) constant, a plot of X_j versus $g_j(X_j)$ reveals the nature of any non-linearities in the effect of X_j on the transformed mean of the response variable Y, $h(\mu)$, assuming that the additive model in (6.2) holds. Hence,

GAMs seem to strike a sensible compromise between ease of interpretation and flexibility.

A GAM can also contain parametric terms, in similar fashion to the semiparametric additive model described in Section 2.3. In this case the linear predictor η has the form

$$\eta = \beta_0 + \beta_1 V_1 + \cdots + \beta_q V_q + \sum_{j=1}^{p} g_j(X_j)\,, \tag{6.3}$$

where $\boldsymbol{V} = (V_1, \ldots, V_q)^\intercal$ is the vector of covariates entering linearly in the GAM. In what follows we shall focus on the GAM in (6.2) in order to describe the local scoring algorithm for estimation of the component functions g_j. The algorithm for the semiparametric model (6.3) is essentially the same.

6.3 Penalized Maximum Likelihood Estimation - The Local Scoring Algorithm

One way to derive the local scoring algorithm is to maximize a penalized likelihood criterion (Hastie & Tibshirani, 1990b). Let Y be a random variable satisfying the properties of a GAM. Let $X_1, \ldots, X_p$ be p possibly relevant explanatory variables. Given a set of n independent realizations, we write $\boldsymbol{y} = (y_1, \ldots, y_n)^\intercal$, and $\boldsymbol{x}_j = (x_{1j}, \ldots, x_{nj})^\intercal$, $j = 1, \ldots, p$. Furthermore, let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^\intercal$, with $\mu_i = \mathbb{E}[Y_i]$, and $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)^\intercal$, where the linear predictor for observation i takes the form

$$\eta_i = \beta_0 + \sum_{j=1}^{p} g_j(x_{ij})\,. \tag{6.4}$$

Assume that the canonical link function is used so that $\eta_i = \theta_i$. For each Y_i, the log-likelihood function is

$$\ell_i(g_1, \ldots, g_p\,; D) = y_i\eta_i - b(\eta_i) + c(y_i)\,,$$

where D represents the observed data. The log-likelihood function for all the Y_i's is

$$\ell(g_1, \ldots, g_p\,; D) = \sum_{i=1}^{n} \ell_i(g_1, \ldots, g_p\,; D) = \sum_{i=1}^{n} \{y_i\eta_i - b(\eta_i) + c(y_i)\}\ . \tag{6.5}$$

Consider the problem of estimating the component functions g_j through penalized likelihood methods using the criterion in (3.1). We start by representing each g_j in (6.4) using a cubic spline with $\mathcal{K}_j$ knots and parametrized through the VFDP, as described in Chapter 4. For simplicity, we focus here upon the single penalty case, for which the penalty is $P_2(g_j\,; \lambda_j)$. Take $\boldsymbol{\alpha}_j$, $\boldsymbol{X}_j$ and $\boldsymbol{P}_2^j$ to be, respectively, the parameter vector, design and penalty matrix associated with the function g_j, and satisfying the same identifiability constraints as those for the additive model of Section 5.1. Thus, $g_j(x_{ij}) = \boldsymbol{X}_{j,i\boldsymbol{.}}\,\boldsymbol{\alpha}_j$, and $P_2(g_j\,; \lambda_j) = \lambda_j\,\boldsymbol{\alpha}_j^{\intercal}\boldsymbol{P}_2^j\,\boldsymbol{\alpha}_j$. So, we can write

$$\eta_i = \beta_0 + \sum_{j=1}^{p} \boldsymbol{X}_{j,i\boldsymbol{.}}\,\boldsymbol{\alpha}_j\,, \tag{6.6}$$

and hence express the log-likelihood in (6.5) as a function of $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p$. Estimation of the curves g_j is therefore posed in terms of the estimation of the corresponding spline parameters: Over all the p-tuples of parameter vectors $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p$ defining continuous cubic spline functions with continuous first derivatives, find the one that

maximizes

$$\mathcal{J}(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p) = \ell(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p; D) - \tfrac{1}{2} \sum_{j=1}^{p} \lambda_j \, \boldsymbol{\alpha}_j^{\mathsf{T}} \boldsymbol{P}_2^j \, \boldsymbol{\alpha}_j \,. \tag{6.7}$$

The constant $\frac{1}{2}$ in (6.7) is included for convenience. The Newton-Raphson algorithm, or some related method, is the standard choice to maximize the optimization criterion in (6.7). Its estimating equation requires the score vector and the observed information matrix. The score for the parameter vector $\boldsymbol{\alpha}_j$ is

$$\begin{aligned}
\boldsymbol{U}_j &= \frac{\partial \mathcal{J}}{\partial \boldsymbol{\alpha}_j} \\
&= \sum_i \frac{\partial \ell_i}{\partial \boldsymbol{\alpha}_j} - \lambda_j \boldsymbol{P}_2^j \, \boldsymbol{\alpha}_j \\
&= \sum_i \{ y_i \, \boldsymbol{X}_{j,i\bullet} - \mu_i \, \boldsymbol{X}_{j,i\bullet} \} - \lambda_j \boldsymbol{P}_2^j \, \boldsymbol{\alpha}_j \\
&= \boldsymbol{X}_j^{\mathsf{T}} (\boldsymbol{y} - \boldsymbol{\mu}) - \lambda_j \boldsymbol{P}_2^j \, \boldsymbol{\alpha}_j \,.
\end{aligned}$$

The $(2\mathcal{K}_j p) \times (2\mathcal{K}_j p)$ observed information matrix has blocks

$$\begin{aligned}
\boldsymbol{\mathcal{I}}_{lj} &= -\frac{\partial^2 \mathcal{J}}{\partial \boldsymbol{\alpha}_l \partial \boldsymbol{\alpha}_j^{\mathsf{T}}} \\
&= -\sum_i \left\{ -\frac{\partial \mu_i}{\partial \eta_i} \, \boldsymbol{X}_{j,i\bullet} \, \boldsymbol{X}_{l,i\bullet}^{\mathsf{T}} \right\} \\
&= \boldsymbol{X}_l^{\mathsf{T}} \boldsymbol{W} \boldsymbol{X}_j \,,
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{\mathcal{I}}_{jj} &= -\frac{\partial^2 \mathcal{J}}{\partial \boldsymbol{\alpha}_j \partial \boldsymbol{\alpha}_j^{\mathsf{T}}} \\
&= \boldsymbol{X}_j^{\mathsf{T}} \boldsymbol{W} \boldsymbol{X}_j + \lambda_j \boldsymbol{P}_2^j \,,
\end{aligned}$$

where $\boldsymbol{W} = \text{diag}\,(w_1, \ldots, w_n)$ is a *weight matrix*, with weights $w_i = \partial\mu_i/\partial\eta_i$. Let $\widehat{\boldsymbol{\alpha}}_1^{[c]}, \ldots, \widehat{\boldsymbol{\alpha}}_p^{[c]}$ be the current estimates of $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p$. The Newton-Raphson step to update from $\widehat{\boldsymbol{\alpha}}_1^{[c]}, \ldots, \boldsymbol{\alpha}_p^{[c]}$ to $\widehat{\boldsymbol{\alpha}}_1^{[c+1]}, \ldots, \widehat{\boldsymbol{\alpha}}_p^{[c+1]}$ is

$$\underline{\widehat{\boldsymbol{\alpha}}}^{[c+1]} = \underline{\widehat{\boldsymbol{\alpha}}}^{[c]} + \left[\boldsymbol{\mathcal{I}}^{[c]}\right]^{-1} \underline{\boldsymbol{U}}^{[c]}\,, \tag{6.8}$$

with $\underline{\boldsymbol{\alpha}} = (\boldsymbol{\alpha}_1^\intercal, \ldots, \boldsymbol{\alpha}_p^\intercal)^\intercal$, and $\underline{\boldsymbol{U}} = (\boldsymbol{U}_1^\intercal, \ldots, \boldsymbol{U}_p^\intercal)^\intercal$. The iteration in (6.8) can be re-expressed as

$$\begin{pmatrix} \boldsymbol{X}_1^\intercal \boldsymbol{W}^{[c]} \boldsymbol{X}_1 + \lambda_1 \boldsymbol{P}_2^1 & \boldsymbol{X}_1^\intercal \boldsymbol{W}^{[c]} \boldsymbol{X}_2 & \ldots & \boldsymbol{X}_1^\intercal \boldsymbol{W}^{[c]} \boldsymbol{X}_p \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{X}_p^\intercal \boldsymbol{W}^{[c]} \boldsymbol{X}_1 & \boldsymbol{X}_p^\intercal \boldsymbol{W}^{[c]} \boldsymbol{X}_2 & \ldots & \boldsymbol{X}_p^\intercal \boldsymbol{W}^{[c]} \boldsymbol{X}_p + \lambda_p \boldsymbol{P}_2^p \end{pmatrix} \begin{pmatrix} \widehat{\boldsymbol{\alpha}}_1^{[c+1]} - \widehat{\boldsymbol{\alpha}}_1^{[c]} \\ \vdots \\ \widehat{\boldsymbol{\alpha}}_p^{[c+1]} - \widehat{\boldsymbol{\alpha}}_p^{[c]} \end{pmatrix}$$

$$= \begin{pmatrix} \boldsymbol{X}_1^\intercal (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{[c]}) - \lambda_1 \boldsymbol{P}_2^1 \widehat{\boldsymbol{\alpha}}_1^{[c]} \\ \vdots \\ \boldsymbol{X}_p^\intercal (\boldsymbol{y} - \widehat{\boldsymbol{\mu}}^{[c]}) - \lambda_p \boldsymbol{P}_2^p \widehat{\boldsymbol{\alpha}}_p^{[c]} \end{pmatrix}, \tag{6.9}$$

where $\boldsymbol{W}^{[c]}$ and $\boldsymbol{\mu}^{[c]}$ are evaluated at $\widehat{\boldsymbol{\alpha}}_1^{[c]}, \ldots, \widehat{\boldsymbol{\alpha}}_p^{[c]}$. Define the vector of *adjusted response variables* $\boldsymbol{z}^{[c]} = (z_1^{[c]}, \ldots, z_n^{[c]})^\intercal$, with $z_i^{[c]} = \widehat{\eta}_i^{[c]} + \left[w_i^{[c]}\right]^{-1} (y_i - \widehat{\mu}_i^{[c]})$, and the smoothing matrices $\boldsymbol{S}_j = (\boldsymbol{X}_j^\intercal \boldsymbol{W}^{[c]} \boldsymbol{X}_j + \lambda_j \boldsymbol{P}_2^j)^{-1} \boldsymbol{X}_j^\intercal \boldsymbol{W}^{[c]}$, $j = 1, \ldots, p$. Then (6.9) can be written as

$$\begin{pmatrix} \widehat{\boldsymbol{\alpha}}_1^{[c+1]} \\ \vdots \\ \widehat{\boldsymbol{\alpha}}_p^{[c+1]} \end{pmatrix} = \begin{pmatrix} \boldsymbol{S}_1 \left(\boldsymbol{z}^{[c]} - \beta_0 - \sum_{l \neq 1} \boldsymbol{X}_l \, \widehat{\boldsymbol{\alpha}}_l^{[c+1]}\right) \\ \vdots \\ \boldsymbol{S}_p \left(\boldsymbol{z}^{[c]} - \beta_0 - \sum_{l \neq p} \boldsymbol{X}_l \, \widehat{\boldsymbol{\alpha}}_l^{[c+1]}\right) \end{pmatrix}. \tag{6.10}$$

In principle, the system of equations in (6.10) could be solved directly using, for example, a QR decomposition. However, the computational cost associated with such procedures is usually high, making them an inefficient choice. Hastie & Tibshirani (1986) advocate the use of the backfitting algorithm to solve the system (6.10). The local scoring algorithm is thus defined as follows:

Local Scoring Algorithm

(1) `Initialize:` $\widehat{\beta}_0 = h(\bar{y})$, $\widehat{\boldsymbol{\alpha}}_j^{[c]} = \mathbf{0}$, `with` $\bar{y} = (1/n)\sum_i y_i$.

(2) `Update: Construct weights`

$$w_i^{[c]} = \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^{[c]}.$$

`Construct the adjusted response variable`

$$z_i^{[c]} = \widehat{\eta}_i^{[c]} + \left[w_i^{[c]}\right]^{-1}(y_i - \widehat{\mu}_i^{[c]}),$$

`with` $\widehat{\eta}_i^{[c]} = \widehat{\beta}_0 + \sum_{j=1}^{p} \boldsymbol{X}_{j,i.}\widehat{\boldsymbol{\alpha}}_j^{[c]}$, `and` $\widehat{\mu}_i^{[c]} = h^{-1}(\widehat{\eta}_i^{[c]})$.

Backfitting Algorithm

`Cycle:` $j = 1, \ldots, p,\ 1, \ldots, p,\ 1, \ldots, p$

$$\widehat{\boldsymbol{\alpha}}_j^{[c+1]} = \boldsymbol{S}_j(\boldsymbol{z}^{[c]} - \widehat{\beta}_0 - \sum_{l \neq j} \boldsymbol{X}_l \, \widehat{\boldsymbol{\alpha}}_l^{[c+1]})$$

`until` RSS $= \sum_{i=1}^{n}(z_i^{[c]} - \widehat{\beta}_0 - \sum_j \boldsymbol{X}_{j,i\bullet} \, \widehat{\boldsymbol{\alpha}}_j^{[c+1]})^2$ `converges.`

`Compute convergence criterion`

$$\Gamma = \frac{\sum_j \|\widehat{\boldsymbol{g}}_j^{[c+1]} - \widehat{\boldsymbol{g}}_j^{[c]}\|}{\sum_j \|\widehat{\boldsymbol{g}}_j^{[c]}\|},$$

`with` $\|\boldsymbol{g}_j\|$ `the norm of the vector of evaluations` $\boldsymbol{g}_j = \boldsymbol{X}_j \, \boldsymbol{\alpha}_j$.

(3) `Repeat step` (2) `until` Γ `is less than some small threshold.`

Note that there are two nested loops in the algorithm. In the inner (backfitting) loop, $\boldsymbol{z}$ is held fixed and the $\boldsymbol{\alpha}_j$'s are re-estimated, while in the outer (Newton-Raphson) loop, $\boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{z}$ and $\boldsymbol{W}$ are updated.

Let $\widetilde{\boldsymbol{S}}_j = \boldsymbol{X}_j \boldsymbol{S}_j$, $j = 1, \ldots, p$, be the smoothing matrices associated with the vector of evaluations $\boldsymbol{g}_j$. Hastie & Tibshirani (1990b) prove existence and uniqueness results for the backfitting algorithm when the smoothing matrices $\widetilde{\boldsymbol{S}}_1, \ldots, \widetilde{\boldsymbol{S}}_p$ are symmetric with eigenvalues in $[0, 1]$ and there is no *concurvity*. Concurvity can be seen as the analogue of collinearity for function spaces. Let $\mathcal{M}_1(\widetilde{\boldsymbol{S}}_j)$ be the space spanned by the eigenvectors of $\widetilde{\boldsymbol{S}}_j$ with eigenvalue 1. Concurvity exists if and only if the spaces $\mathcal{M}_1(\widetilde{\boldsymbol{S}}_j)$ are linearly dependent, i.e., there exist $f_j \in \mathcal{M}_1(\widetilde{\boldsymbol{S}}_j)$, not all

zero, satisfying $\sum_j f_j = 0$. In the case of penalized cubic splines, the eigenspaces $\mathcal{M}_1(\widetilde{\boldsymbol{S}}_j)$ correspond to linear functions of the predictor. Hence, concurvity exits only if the predictors are collinear.

The smoothing matrices $\widetilde{\boldsymbol{S}}_1, \ldots, \widetilde{\boldsymbol{S}}_p$ corresponding to the system of equations in (6.10) are not symmetric. This is generally the case for smoothing matrices arising in the context of GAMs due to the presence of the weight matrix $\boldsymbol{W}$. However, the same existence and uniqueness results apply, as one can simply map the problem to the unweighted case. For a thorough proof see Hastie & Tibshirani (1990b). In what follows, a stochastic version of the local scoring algorithm for Bayesian inference is described.

6.4 Bayesian GAMs

Consider the Bayesian estimation of the GAM in (6.6) using both single and double penalty models. The prior specifications for the parameters $\boldsymbol{\alpha}_j$ and β_0 in (6.6) are the same as those described in Subsection 5.2.1, as is the prior on the hyperparameter λ_j in the single penalty model, and hence we omit them here but refer the reader to the aforementioned part of this thesis. Let D represent the observed data, and take $L\left(\beta_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p \,;\, D\right)$ to be the likelihood function of the GAM with linear predictor defined in (6.6). For the single penalty model the joint posterior distribution of the GAM is

$$p\left(\boldsymbol{\theta} \mid D\right) \propto L\left(\beta_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p \,;\, D\right) \prod_j \lambda_j^{\mathrm{rk}(\boldsymbol{P}_2^j)/2} \exp\left(-\frac{\lambda_j}{2}\, \boldsymbol{\alpha}_j^{\intercal} \boldsymbol{P}_2^j\, \boldsymbol{\alpha}_j\right) \times \prod_j \lambda_j^{s_j - 1} \exp\left(-\, r_j \lambda_j\right), \quad (6.11)$$

with $\boldsymbol{\theta} = (\beta_0, \boldsymbol{\alpha}_1, \lambda_1, \ldots, \boldsymbol{\alpha}_p, \lambda_p)^\intercal$ the vector of all the parameters in the model. The posterior density in (6.11) is, again, analytically intractable and so inference relies upon Monte Carlo estimates obtained using an MCMC algorithm. Because the full conditional for $\boldsymbol{\alpha}_j$ is not of standard form, its value can not be updated using a Gibbs sampler. Below we describe an alternative updating scheme based upon the local scoring algorithm.

We explore the high-dimensional posterior distribution in (6.11) using a hybrid approach that combines the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) with the Gibbs sampler (see Appendix B for details of these algorithms). The main idea is to approximate the full conditional of $\boldsymbol{\alpha}_j$, which is not available, by a Gaussian distribution obtained by accomplishing one local scoring step, as proposed in Brezger & Lang (2006). This process is repeated in every iteration of the algorithm. The full conditional for the hyperparameter λ_j is readily available and so a Gibbs step is used to update its value.

Denote by $\boldsymbol{\alpha}_j^c$ and λ_j^c the current value of the parameters associated with the curve $g_j(x_j)$. Let $\boldsymbol{\eta}^c$ be the current predictor based upon the spline parameters $\boldsymbol{\alpha}_j^c$ and intercept β_0^c. Recall the local scoring algorithm described in Section 6.3. A new value $\boldsymbol{\alpha}_j^p$ is proposed by drawing from the multivariate Gaussian proposal distribution $q\left(\boldsymbol{\alpha}_j^c, \boldsymbol{\alpha}_j^p\right)$ with covariance matrix and mean

$$\boldsymbol{\Sigma}_j = \left[\boldsymbol{X}_j^\intercal \boldsymbol{W}^c \boldsymbol{X}_j + \lambda_j^c \boldsymbol{P}_2^j\right]^{-1}, \quad \boldsymbol{m}_j = \boldsymbol{\Sigma}_j \boldsymbol{X}_j^\intercal \boldsymbol{W}^c \left(\boldsymbol{z}^c - \beta_0^c - \sum_{l \neq j} \boldsymbol{X}_l \boldsymbol{\alpha}_l^c\right). \quad (6.12)$$

The matrix $\boldsymbol{W}^c$ and the vector $\boldsymbol{z}^c$ contain the usual weights and adjusted response variables for the local scoring algorithm. They depend upon the current value of the predictor, $\boldsymbol{\eta}^c$, which in turn depends upon the current state of the parameter vector

$\boldsymbol{\alpha}_j^c$. We use the modified algorithm of Brezger & Lang (2006), which replaces $\boldsymbol{\alpha}_j^c$ by the current posterior mode approximation $\boldsymbol{m}_j^c$ for computing $\boldsymbol{W}^c$ and $\boldsymbol{z}^c$. The vector $\boldsymbol{m}_j^c$ is the mean of the proposal distribution corresponding to the last iteration of the sampler. The proposal scheme is now independent of the current value of $\boldsymbol{\alpha}_j$, i.e., $q\left(\boldsymbol{\alpha}_j^c, \boldsymbol{\alpha}_j^p\right) \equiv q\left(\boldsymbol{\alpha}_j^p\right)$. Hence, there is considerable reduction in computational cost because $\boldsymbol{\Sigma}_j$ and $\boldsymbol{m}_j$ do not need to be recomputed when evaluating the proposal density $q\left(\boldsymbol{\alpha}_j^p, \boldsymbol{\alpha}_j^c\right)$. The proposed parameter vector $\boldsymbol{\alpha}_j^p$ is accepted with probability

$$\xi(\boldsymbol{\alpha}_j^c, \boldsymbol{\alpha}_j^p) = \min\left\{1, \frac{p\left(\boldsymbol{\alpha}_j^p \mid \cdot\right) q\left(\boldsymbol{\alpha}_j^c\right)}{p\left(\boldsymbol{\alpha}_j^c \mid \cdot\right) q\left(\boldsymbol{\alpha}_j^p\right)}\right\}, \tag{6.13}$$

where $p\left(\boldsymbol{\alpha}_j \mid \cdot\right)$ is the full conditional for $\boldsymbol{\alpha}_j$. If the proposal is accepted we set $\boldsymbol{\alpha}_j^c = \boldsymbol{\alpha}_j^p$, otherwise we keep the current state $\boldsymbol{\alpha}_j^c$. Lastly, we set $\boldsymbol{m}_j^c = \boldsymbol{m}_j$.

The iterative update for the intercept term β_0 follows similar steps to those described above for the spline parameters $\boldsymbol{\alpha}_j$. The proposal density is the Gaussian distribution with variance and mean given by

$$\Sigma_{\beta_0} = \left[\mathbf{1}^\intercal \boldsymbol{W}^c \mathbf{1}\right]^{-1}, \qquad m_{\beta_0} = \Sigma_{\beta_0} \mathbf{1}^\intercal \boldsymbol{W}^c \left(\boldsymbol{z}^c - \sum_{j=1}^{p} \boldsymbol{X}_j \boldsymbol{\alpha}_j^c\right), \tag{6.14}$$

with $\mathbf{1}$ a $n \times 1$ vector of 1's. For the semiparametric model (6.3), the parameters of the proposal for $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_q)^\intercal$ are the same as those in (6.14) with $\mathbf{1}$ replaced by the design matrix $\boldsymbol{V}$ with ith row $\boldsymbol{V}_{i\cdot} = (1, v_{i1}, \ldots, v_{iq})$.

In the applications we consider, convergence to the stationary distribution is usually very fast, even with poor starting values for the $\boldsymbol{\alpha}_j$'s. Brezger & Lang (2006) propose to run the local scoring algorithm with small enough smoothing parameters in order to obtain initial estimates for the $\boldsymbol{\alpha}_j$'s. We did not find this

necessary here.

Once $\boldsymbol{\alpha}_j$ has been updated, a new value for the hyperparameter λ_j is obtained through a Gibbs step by sampling from its full conditional, a gamma distribution with parameters given by (5.6).

Below we describe the MCMC sampling scheme in detail.

MCMC Algorithm

(1) Initialize: $\beta_0^c = h(\bar{y})$, $\boldsymbol{\alpha}_j^c = \mathbf{0}$. Set these as the current posterior mode estimates $m_{\beta_0}^c$ and $\boldsymbol{m}_j^c$. Fix λ_j (e.g., $\lambda_j = 0.1$). Store the current linear predictor $\boldsymbol{\eta}^c$.

(2) For $j = 1, \ldots, p$ do:

- ⋆ Compute the likelihood $L(\beta_0^c, \ldots, \boldsymbol{\alpha}_j^c, \ldots; D)$.
- ⋆ Replace $\boldsymbol{\alpha}_j^c$ in $\boldsymbol{\eta}^c$ by the current posterior mode estimate $\boldsymbol{m}_j^c$.
- ⋆ Sample $\boldsymbol{\alpha}_j^p$ from the Gaussian proposal $q(\boldsymbol{\alpha}_j^c, \boldsymbol{\alpha}_j^p)$ with variance $\boldsymbol{\Sigma}_j$ and mean $\boldsymbol{m}_j$ given in (6.15).
- ⋆ Replace $\boldsymbol{m}_j^c$ in $\boldsymbol{\eta}^c$ by $\boldsymbol{\alpha}_j^p$.
- ⋆ Compute the likelihood $L(\beta_0^c, \ldots, \boldsymbol{\alpha}_j^p, \ldots; D)$.
- ⋆ Accept $\boldsymbol{\alpha}_j^p$ as the new value for $\boldsymbol{\alpha}_j$ with probability (6.13). If $\boldsymbol{\alpha}_j^p$ is rejected, exchange $\boldsymbol{\alpha}_j^p$ in $\boldsymbol{\eta}^c$ by $\boldsymbol{\alpha}_j^c$.
- ⋆ Set $\boldsymbol{m}_j^c = \boldsymbol{m}_j$.

(3) Update β_0 by similar steps as for the update of $\boldsymbol{\alpha}_j$.

(4) For $j = 1, \ldots, p$ update the smoothing parameters λ_j by drawing from the gamma distribution with parameters given in (5.6).

(5) Repeat step (2) until the chain converges.

In general, the sampling scheme above yields very high acceptance rates, around 95% or above. This is due to the good approximation of the proposal to the posterior distribution.

If a double penalty model is to be estimated, then the empirical Bayes approach described in Chapter 5 can be implemented for GAMs. We rely upon the local scoring algorithm described in Section 6.3 to obtain, for each value of $\boldsymbol{\lambda}_j = \left\{\lambda_j^1, \lambda_j^2\right\}$, estimates for the spline parameters $\boldsymbol{\alpha}_j$. Having estimated the smoothing parameters in the double penalty functionals, estimation of the spline parameters $\boldsymbol{\alpha}_j$ is performed using the Metropolis-Hastings algorithm described above, but with the variance of the proposal distribution for $\boldsymbol{\alpha}_j$ given by

$$\boldsymbol{\Sigma}_j = \left[\boldsymbol{X}_j^{\intercal} \boldsymbol{W}^c \boldsymbol{X}_j + \widehat{\lambda}_j^1 \boldsymbol{P}_1 + \widehat{\lambda}_j^2 \boldsymbol{P}_2\right]^{-1}. \tag{6.15}$$

6.5 Union Membership Data

The proposed methodology is illustrated with an application to a data set given in Berndt (1991) and analyzed in Ruppert et al. (2003). These data consist of a random sample of 533 persons from the Current Population Survey (CPS) undertaken in 1985 in the United States, with information on wages and other characteristics of the individuals, including sex, age, race, number of years of education, region of residence and union membership. Interest lies in the relationship between the

probability that an individual is a union member and their wage. The variable `union` is binary (1 if member, 0 otherwise) as are the variables `south` (1 if person lives in the south, 0 if person lives elsewhere) and `gender` (1 if female, 0 if male). The variable `race` is categorical with three categories: `race` = 1 if Other, `race` = 2 if Hispanic, `race` = 3 if White. The variables `education`, `age` (in number of years) and `wage` (in \$/hour) are continuous.

We start by considering the following semiparametric GAM:

$$\begin{aligned} \texttt{union} &\sim \text{Bernoulli}\,(\mu)\,, \\ \text{logit}(\mu) &= \beta_0 + \beta_1\, V_1 + \beta_2\, V_2 + \beta_3\, V_3 + \beta_4\, V_4 + g_1(X_1) + g_2(X_2) + g_3(X_3)\,, \end{aligned} \tag{6.16}$$

with

$$(V_1, V_2, V_3, V_4)^{\mathsf{T}} = (\texttt{race}_{\text{o}},\, \texttt{race}_{\text{h}},\, \texttt{gender},\, \texttt{south})^{\mathsf{T}}\,, \text{ and}$$
$$(X_1, X_2, X_3)^{\mathsf{T}} = (\texttt{wage},\, \texttt{age},\, \texttt{education})^{\mathsf{T}}\,.$$

The covariates $\texttt{race}_{\text{o}}$ and $\texttt{race}_{\text{h}}$ represent the dummy variables associated with the categorical covariate `race` with the base category taken to be White. So, for example, $\texttt{race}_{\text{o}} = 1$, if `race` = Other, and $\texttt{race}_{\text{o}} = 0$ otherwise; likewise for $\texttt{race}_{\text{h}}$.

The smooth terms g_1, g_2, and g_3 in (6.16) are modeled nonparametrically. Each is represented using a cubic spline function parametrized according to the VFDP with 20 knots placed at equally spaced quantiles of the observed values of the corresponding covariate.

We start our analysis of these data with the single penalty model based upon the functional $P_2(g\,;\lambda) = \lambda \int g''(x)^2\, dx$. Let $\boldsymbol{\alpha}_j$ be the parameter vector associated

with the spline g_j, $j = 1, 2, 3$. The prior definitions follow those described in Subsection 5.2.1. For the parameters s_j and r_j in the prior for λ_j the values $s_j = r_j = 10^{-4}$ yielded the best results, though a sensitivity analysis revealed that the results were robust to the choice of s_j and r_j. The estimated effects in Table 6.1 and Figure 6.1 were obtained using the output of a chain of length 100,000 for the spline parameters $\boldsymbol{\alpha}_j$, $j = 1, 2, 3$, and for the linear effect parameters β_u, $u = 0, \ldots, 4$ (after an initial burn-in period of length 2,000). Convergence of the chain was determined by examining the plot of its path.

Table 6.1: Posterior mean estimates of the linear effects in the GAM fit with a single penalty functional.

Parameter	Posterior Mean (95% C.I.)
β_0	-1.5 (-1.9, -1.1)
β_1	0.8 (0.1, 1.4)
β_2	-0.6 (-0.6, 1.8)
β_3	-0.7 (-1.3, -0.2)
β_4	-0.5 (-1.2, 0.1)

C.I., Credible Interval.

The plots in Figure 6.1 show the estimated probability that an individual is a union member as a function of the covariate in the horizontal axis with all the other variables in the model fixed at their observed means. Though the pointwise credibility intervals are very wide for certain values of the covariates, generally one can say that the probability of union membership is a linear function of `age`, and that the effects of `wage` and `education` are nearly linear. Older individuals have higher probability of being a union member. The probability of an individual being

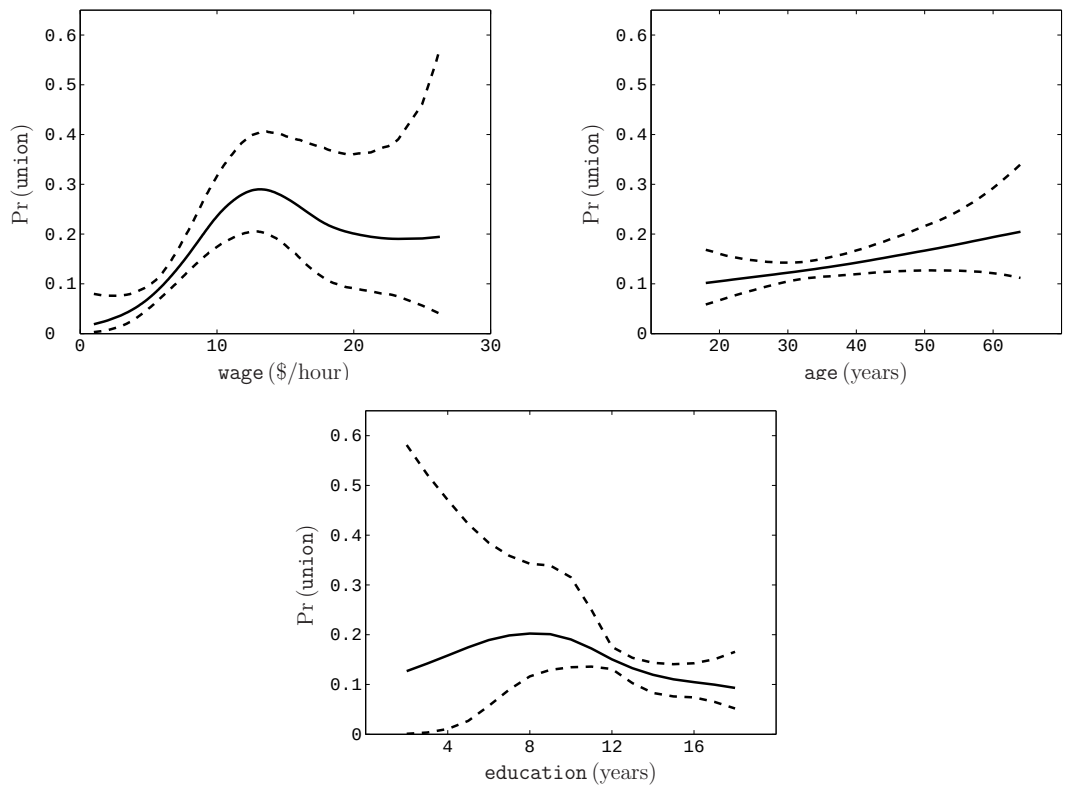

Figure 6.1: Posterior mean estimates of the smooth terms in the GAM fit (solid lines) with a single penalty functional, together with 95% *pointwise credible intervals (dashed lines).*

a union member increases with `wage` up to \$15 per hour and decreases as the level of education rises.

Finally, Table 6.1 shows that the probability of being a union member is significantly higher for individuals of other race compared to white individuals, and significantly lower for females.

The plots in Figure 6.1 show evidence towards a simplified semiparametric model with a linear effect for `age`. To investigate this further we computed the value of the Akaike's information criterion (AIC) for the model in (6.16) estimated using a single penalty as above, and for the model with a linear effect for `age`. The

results are presented in Table 6.2 (rows 1 and 2). The results show that the linear effect for `age` is not statistically significant, and that the model with a nonlinear effect for `age` provides a better fit to the data in terms of lower AIC. Note that the latter is essentially a linear function of age with a very small slope. Hence, the AIC analysis suggests a model excluding the covariate `age`. We went on to

Table 6.2: Degrees of freedom (d.f.) and Akaike's information criterion (AIC) for different model specifications, together with the posterior mean estimate of the linear effect for covariate `age`*.*

Penalty	Effect of `age`	d.f.	AIC	Posterior Mean (95% C.I.)
SP	linear	9.7	458.9	0.2e-1 (-0.1e-1, 0.4e-1)
SP	nonlinear	9.6	458.7	.
DP	nonlinear	8.5	455.9	.
DP	linear	9.9	458.4	0.2e-1 (-0.2e-1, 0.4e-1)

SP, Single Penalty; DP, Double Penalty; C.I., Credible Interval.

analyze the model in (6.16) using the double penalty functional in (3.9), which shrinks the estimated smooth terms towards a constant. In this case, a grid for the values of the two smoothing parameters in the penalty has to be specified. The grid 10^{-5}, $10^{-4}, \ldots, 10^{4}$, 10^{5} provided satisfactory results. The corresponding AIC value can be found in Table 6.2 (row 3). A comparison of this with the AIC values in rows 1 and 2 of Table 6.2 suggests that double penalty estimates of the components of the model in (6.16) provide a better fit to the data compared to single penalty ones. These can be found in Table 6.3 and Figure 6.2. The estimated effects are very similar to those obtained with the single penalty model. In particular, the effect of `age` still appears to be linear. However, now the slope is shrunk towards zero, which may explain the better fit obtained with the double penalty model. Moreover, the

added penalty functional yields a more stable estimate for the effect of `education`, with narrower pointwise credible intervals. We also considered the double penalty

Table 6.3: Posterior mean estimates of the linear effects in the GAM fit with a double penalty functional.

Parameter	Posterior Mean (95% C.I.)
β_0	-1.5 (-1.9, -1.1)
β_1	0.8 (0.1, 1.4)
β_2	-0.6 (-0.6, 1.8)
β_3	-0.7 (-1.3, -0.2)
β_4	-0.5 (-1.1, 0.1)

C.I., Credible Interval.

model with a linear effect for covariate `age`, for which the AIC value can be found in Table 6.2 (row 4). This model performs better than the single penalty models but worst than the double penalty model with a smooth term for `age`.

The analysis above suggests that, even though the effect of `age` on the probability of union membership appears to be linear from a graphical point of view, a model with a nonlinear effect for `age` provides a better fit to the data.

6.6 Summary

We applied the proposed methodology in the context of GAMs. We have described an MCMC scheme for updating the spline parameters based upon the local scoring algorithm. We presented a real data analysis where a semiparametric logistic regression model was fitted to data obtained from the 1985 CPS in the U.S. The smooth terms in the model were parametrized using the VFDP and estimated using both a

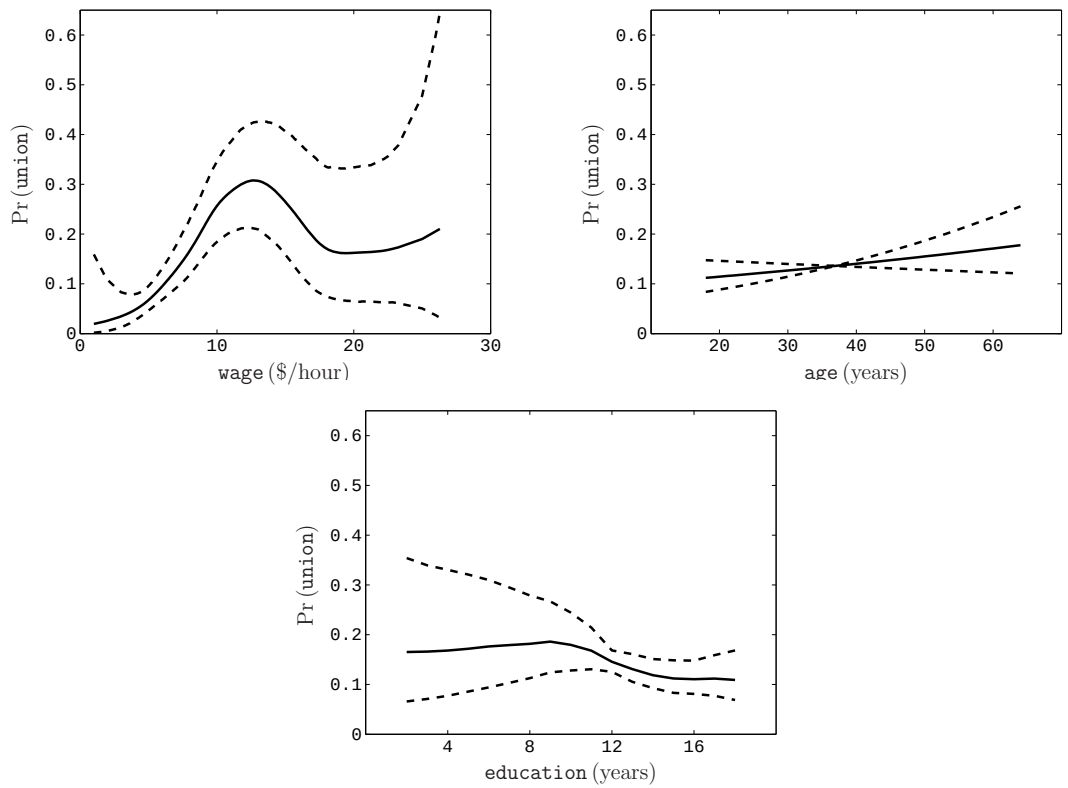

Figure 6.2: Posterior mean estimates of the smooth terms in the GAM fit (solid lines) with a double penalty functional, together with 95% *pointwise credible intervals (dashed lines).*

single and a double penalty functional. The latter resulted in more stable estimates and provided a better fit to the data.

Chapter 7

The VFDP in Survival Analysis

7.1 Introduction

The focus of interest in survival analysis is the time to an event. Throughout we use the term *failure* to describe the event of interest, and refer to the time to failure as a *survival time*. To determine the survival time, we need to define two time points: the time of origin, i.e. the time at which an original event, such as birth, occurs and the time of failure, i.e. the time at which the final event, such as death, occurs. A subject is said to be at risk if the original event has occurred, but the final event has not. An important area of application of survival analysis is medicine, where, for example, interest may centre on whether a new treatment lengthens the life of a cancer patient, relative to those who receive existing treatments. Other areas of application include the social sciences or engineering.

A key characteristic that distinguishes survival analysis from other areas in statistics is that survival data are usually censored or incomplete in some way. For example, a patient may be lost to follow-up for some reason unrelated to his or her disease, so that it is unknown whether or not he or she died from the cause under

study. There are several types of censoring mechanisms. These usually depend upon the type of application or upon the specific design of the study. We shall denote by T the non-negative random variable whose value corresponds to the time to the event of interest.

This chapter starts by outlining some important notions in survival analysis. We then proceed to describe one of the most common regression models for survival data, the *Cox proportional hazards model* (Cox, 1972). In its usual form, the Cox proportional hazards (PH) model is a semiparametric model in which dependence upon explanatory variables is modeled explicitly in parametric fashion but no specific probability distribution is assumed for the survival times. Hence, inference is based on a *partial likelihood* function rather than on the full likelihood.

Here focus is on a generalization of the PH model that overcomes the assumption of proportional hazards. Essentially, covariate effects are allowed to vary with follow-up time. This may happen if, for example, a treatment gradually loses its effectiveness with time. The *Cox model with time-varying regression coefficients*, or *nonproportional hazards model*, is a complex setting that has attracted a lot of attention in recent years (see, e.g., Kauermann, 2005; Martinussen & Scheike, 2006). The time-varying regression coefficients are represented using penalized cubic spline functions estimated within a Bayesian framework based upon the partial likelihood of the model. The spline curves are parametrized through the VFDP. The inference process is similar to that described in Chapter 6 for GAMs but changes in the proposal densities have to be made due to the specific form of the partial likelihood. Sargent (1997) uses Bayesian dynamic linear models to account for possible time dependencies of covariate effects. Lambert & Eilers (2005) also use Bayesian cubic penalized splines to estimate time-varying regression coefficients but build a Poisson

approximation to the full likelihood function.

The fact that the VFDP is not bound to any particular grid of knots becomes particularly useful in survival analysis. The choice of the knots on the time axis can be based upon complete information only, thus disregarding censored survival times. This is convenient in situations of high censoring for which data can be sparse. We place the knots not at equally spaced points in time but at equally spaced quantiles of the observed failure times, thus ensuring roughly the same amount of information between knots (Gray, 1992). Finally, an application of the proposed methodology is presented based on the well known data set on primary biliary cirrhosis (PBC) which can be found in Fleming & Harrington (1991, Appendix D). The analysis therein showed that the effect of some of the covariates seems to vary with time. We shall see that the VFDP is a flexible tool that is simple to use yet able to yield smooth, clinically plausible estimates of covariate effects.

The book by Collett (2003) provides a comprehensive analysis of the most relevant methodologies in the framework of survival analysis.

7.2 Concepts and Definitions in Survival Analysis

7.2.1 Survival and Hazard Functions

Let the continuous random variable T represent the survival times of individuals in some population. All functions in this chapter, unless otherwise stated, are defined over the interval $[0, +\infty)$. Let $f_T(t)$ denote the probability density function of T. Then the probability of failure before a specific time t is given by the cumulative

probability distribution

$$\mathcal{F}(t) = \Pr(T \leq t) = \int_0^t f_T(u)\, du\,.$$

The *survival function* is defined as the probability of an individual surviving beyond time t. Thus, it is given by

$$S(t) = \Pr(T \geq t) = 1 - \mathcal{F}(t)\,.$$

We note that $S(t)$ is a monotonic decreasing function with $S(0) = 1$ (there can not be a failure before time 0) and $S(+\infty) = \lim_{t\to+\infty} S(t) = 0$ (asymptotically all events realize). A central concept in survival analysis is the *hazard function* of T, defined loosely as the probability of failure in an infinitesimally small time interval between t and $(t + \varepsilon t)$, given survival up to time t,

$$h(t) = \lim_{\varepsilon t\to 0} \frac{\Pr(t \leq T < t + \varepsilon t \mid T \geq t)}{\varepsilon t} = \frac{f_T(t)}{S(t)}\,. \tag{7.1}$$

The functions $f_T(t)$, $\mathcal{F}(t)$, $S(t)$ and $h(t)$ give mathematically equivalent specifications of the distribution of T. It is easy to derive expressions for $S(t)$ and $f_T(t)$ in terms of $h(t)$. Since $f_T(t) = -\frac{d}{dt}S(t)$, (7.1) implies that

$$h(t) = -\frac{d}{dt}\log\{S(t)\}\,. \tag{7.2}$$

Hence

$$S(t) = \exp\{-H(t)\}\,, \quad \text{where} \quad H(t) = \int_0^t h(u)\, du\,, \tag{7.3}$$

or

$$H(t) = -\log\{S(t)\}\,.$$

The function $H(t)$ is called *cumulative hazard function*. Since $S(+\infty) = 0$, it follows that $H(+\infty) = \lim_{t\to+\infty} H(t) = +\infty$. Thus, the hazard function $h(t)$ has the properties

$$h(t) \geq 0\,, \quad \text{and} \quad \int_0^{+\infty} h(t)\,dt = +\infty\,. \tag{7.4}$$

Finally, in addition to (7.3), it follows from (7.2) that

$$f_T(t) = h(t)\exp\{-H(t)\}\,.$$

All the functions above can also be derived for discrete or mixed discrete-continuous survival times, but here we shall focus upon the continuous case.

7.2.2 Censoring Mechanisms

With some exceptions, the censoring mechanisms in most observational studies are unknown. Two such exceptions are the so-called *Type I censoring* and *Type II censoring*. Type II designs are studies in which n independent variables are observed until there have been r failures, so the first r order statistics $0 < T_{(1)} < \cdots < T_{(r)}$ are observed. All that is known about the $n-r$ remaining observations is that they exceed $T_{(r)}$. This scheme is typically used in industrial life-testing. In a Type I censoring design, a random variable T is watched until a pre-determined time c. If $T \leq c$, we observe the value t of T, but if $T > c$, we know only that T is larger than c.

In medical and epidemiological studies, the censoring time c is often random rather than fixed. Under random censoring we suppose that the ith of n independent units has an associated censoring time C_i drawn from a distribution G, independent of its survival time T_i^0. The time actually observed is $T_i = \min\{T_i^0, C_i\}$, and it is known whether $T_i = T_i^0$ or $T_i = C_i$, an event indicated by δ_i. Thus a pair (t_i, δ_i) is observed for each unit, with $\delta_i = 1$ if t_i is the survival time and $\delta_i = 0$ if t_i is the censoring time. This type of censoring, known as *right censoring*, is important in medical applications, where a patient may either die of a cause unrelated to the reason they are being studied, or withdraw from the study or be lost to follow-up, or the study may end before his or her survival time is observed.

The assumption that T_i^0 and C_i are uninformative with respect to each other is critical and may induce serious bias in the analysis if not taken into account properly. It implies that estimation of the parameters in the distribution $\mathcal{F}(t)$ of T^0 is independent of having knowledge of the distribution G of the random censoring variable C. Thus, G is usually left unspecified.

Other types of random censoring mechanisms are also possible. *Left censoring* occurs when the time of origin is not known exactly, for example if time to death from a disease is observed, but the time of infection is unknown. If an observation is both right and left censored, we say that the observation is *doubly censored*. In some applications, the time of the event may be known only up to a time interval, especially when the time is established by periodical examinations. These observations are called *interval censored*. In this thesis we shall only be concerned with data subject to right censoring.

7.3 Proportional Hazards Model and Partial Likelihood

In medical applications the focus of interest is typically on how treatments or certain characteristics of the units affect survival, the form of the survival distribution being of secondary importance. This suggests that one seeks inferences that will be valid for any such distribution.

The hazard function depends in general on both time and a set of covariates. The Cox proportional hazards (PH) model (Cox, 1972) separates these components by specifying that the hazard at time t for an individual whose p-dimensional covariate vector is $\boldsymbol{x} = (x_1, \ldots, x_p)^\intercal$ is given by

$$h(t \mid \boldsymbol{x}, \boldsymbol{\beta}) = h_0(t) \exp\left(\boldsymbol{\beta}^\intercal \boldsymbol{x}\right), \tag{7.5}$$

where $h_0(t)$ is called the *baseline hazard* and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\intercal$ is a p-dimensional vector of regression coefficients. The function $h_0(t)$ is the hazard corresponding to the reference levels for the explanatory variables in $\boldsymbol{x}$ and hence the name 'baseline'. The second term in (7.5) is written in exponential form to ensure it remains positive (recall (7.4) in Subsection 7.2.1). Finally, note that the model in (7.5) assumes that the effect of the covariates on the hazard is multiplicative.

For a single binary covariate with values $x = 0$ if the exposure is absent and $x = 1$ if the exposure is present, the *hazard ratio* or *relative hazard* for presence vs. absence of exposure is

$$\frac{h(t \mid x = 1, \beta)}{h(t \mid x = 0, \beta)} = \exp\left(\beta\right), \tag{7.6}$$

which does not depend upon the time t since the baseline hazard cancels out. A one-unit increase in a continuous covariate will also result in the hazard ratio given

in (7.6). More generally, the model in (7.5) implies that the hazard ratio for two individuals does not involve $h_0(t)$ and hence their hazards are proportional. This PH assumption is strong and must be checked in practice. Later in this chapter we shall study an extension of the model in (7.5) which attempts to deal with situations where the PH assumption is violated.

Suppose that data are available on N individuals, and assume from these that we have n distinct failure times and $N-n$ right censored survival times. For simplicity, we assume here that only one individual fails at each time, so that there are no ties in the data. Hence, the available data consists of the independent triplets $D = (t_i, \delta_i, \boldsymbol{x}_i)_{i=1}^N$, with δ_i as before. The vector $\boldsymbol{x}_i$ contains the measurements on p selected time-constant covariates for individual i. The observed failure times are $\tilde{t}_1 < \cdots < \tilde{t}_n$. Consider a parametric model under which the survival time has density $f_T(t \mid \boldsymbol{x}, \boldsymbol{\beta})$, survival function $S(t \mid \boldsymbol{x}, \boldsymbol{\beta})$, and hazard and cumulative hazard functions $h(t \mid \boldsymbol{x}, \boldsymbol{\beta})$ and $H(t \mid \boldsymbol{x}, \boldsymbol{\beta})$, respectively. We shall assume throughout that censoring is uninformative about $\boldsymbol{\beta}$. The likelihood contribution from individual i is

$$f_T(t_i \mid \boldsymbol{x}_i, \boldsymbol{\beta}) \quad \text{if } \delta_i = 1, \quad \text{and} \quad S(t_i \mid \boldsymbol{x}_i, \boldsymbol{\beta}) \quad \text{if } \delta_i = 0\,.$$

Using the relationship in (7.1), the overall log-likelihood function may be written as

$$\ell(\boldsymbol{\beta}\,; D) = \sum_{i=1}^{N} \left[\delta_i \log\left\{h(t_i \mid \boldsymbol{x}_i, \boldsymbol{\beta})\right\} + \log\left\{S(t_i \mid \boldsymbol{x}_i, \boldsymbol{\beta})\right\}\right] .$$

For the PH model in (7.5) the log-likelihood takes the form

$$\ell(\boldsymbol{\beta}, h_0(t)\,; D) = \sum_{i=1}^{N} \left[\delta_i(\log\left\{h_0(t)\right\} + \boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{x}_i) - \exp\left(\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{x}_i\right) \int_0^{t_i} h_0(u)\, du\right] .$$

The action of the covariates being of primary interest, we seek a likelihood on which to base inference for $\boldsymbol{\beta}$, regardless of $h_0(t)$. Partial likelihood based inference was first proposed by Cox (1972) together with the PH model in (7.5), and later developed in more detail in Cox (1975). The basic idea was to treat the baseline hazard $h_0(t)$ as an infinite-dimensional nuisance parameter when estimating the regression coefficients $\boldsymbol{\beta}$. To motivate further the use of the partial likelihood note that, if the hazard function was entirely arbitrary, then inference could only be based upon events where failures actually occurred, because the hazard might in principle be zero at every other time. Thus it suffices to estimate the *baseline cumulative hazard function*, $H_0(t) = \int_0^t h_0(u)\,du$, by a step function, i.e., $H_0(t) = \sum_{i:\,t_i \le t} h_i$, where $h_i = h_0(t_i) > 0$ only at observed failures. Let R_i denote the *risk set* of individuals for whom the event of interest did not occur by the instant before t_i, i.e., all individuals except those who have previously failed or been censored. Then the log-likelihood is

$$\begin{aligned}\ell(\boldsymbol{\beta}, h_1, \ldots, h_N\,; D) &= \sum_{i=1}^{N} \left[\delta_i (\log\{h_i\} + \boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{x}_i) - \exp(\boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{x}_i) \sum_{u=1}^{i} h_u \right] \\ &= \sum_{i=1}^{N} \left[\delta_i (\log\{h_i\} + \boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{x}_i) - h_i \sum_{l \in R_i} \exp(\boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{x}_l) \right] .\end{aligned}$$

The last equality follows from the fact that

$$\begin{aligned}\sum_{i=1}^{N} \left[\exp(\boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{x}_i) \sum_{u=1}^{i} h_u \right] &= \sum_{i=1}^{N} \left[h_i \left\{ \exp(\boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{x}_i) + \exp(\boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{x}_{i+1}) + \cdots + \exp(\boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{x}_N) \right\} \right] \\ &= \sum_{i=1}^{N} \left[h_i \sum_{l \in R_i} \exp(\boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{x}_l) \right] .\end{aligned}$$

Suppose $\boldsymbol{\beta}$ is fixed. The quantities h_i have maximum likelihood estimators $\widehat{h}_i =$

$\delta_i / \sum_{l \in R_i} \exp(\boldsymbol{\beta}^\intercal \boldsymbol{x}_l)$, which are positive only when $\delta_i = 1$. The profile log-likelihood for $\boldsymbol{\beta}$ is

$$\ell_p(\boldsymbol{\beta}\,; \widehat{h}_1, \ldots, \widehat{h}_N, D) = \sum_{i=1}^{N} \delta_i \log \left(\frac{\exp(\boldsymbol{\beta}^\intercal \boldsymbol{x}_i)}{\sum_{l \in R_i} \exp(\boldsymbol{\beta}^\intercal \boldsymbol{x}_l)} \right),$$

with corresponding profile likelihood

$$L_p(\boldsymbol{\beta}\,; D) = \prod_{f=1}^{n} \frac{\exp(\boldsymbol{\beta}^\intercal \tilde{\boldsymbol{x}}_f)}{\sum_{l \in R_f} \exp(\boldsymbol{\beta}^\intercal \boldsymbol{x}_l)}, \tag{7.7}$$

where $\tilde{\boldsymbol{x}}_f$ is the value of the p-dimensional covariate vector for the individual who fails at $\tilde{t}_f$, $f = 1, \ldots, n$, and D now contains the estimates $\widehat{h}_1, \ldots, \widehat{h}_N$. Note that the product is taken over the individuals for whom event times have been recorded. Hence, individuals for whom the survival times are censored do not contribute to the numerator of (7.7). Furthermore, the value of (7.7) depends only upon the ranking of the failures, since this determines the risk set at each failure. The expression in (7.7) is known as *partial likelihood*, but it can be treated as an ordinary likelihood (Andersen & Gill, 1982). The partial maximum likelihood estimate of $\boldsymbol{\beta}$ can be obtained by maximizing (7.7) with respect to $\boldsymbol{\beta}$. This can be accomplished using numerical methods such as the Newton-Raphson algorithm. A Bayesian justification of the use of the partial likelihood is provided by Kalbfleisch (1978).

7.4 Nonproportional Hazards Model

7.4.1 Model Specification

The Cox's PH model in (7.5) may not reflect all the important aspects of the data and hence may give misleading summaries. There are many ways in which such

model can fail. The functional form of the individual covariates may be misspecified, the use of the function 'exp' may not be appropriate, meaning that the relationship between the hazard and the predictor $\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{x}$ may not be log-linear, and the regression coefficients may not be constant with time (the PH assumption). In this chapter we shall focus upon models that generalize the PH assumption. The latter implies that the hazard ratio is constant over time. While this may be reasonable in some settings it fails in others. For example, in practice one often encounters covariate effects, such as treatment effects, that are weakened with time.

A natural extension of the Cox model that accommodates time-varying covariate effects is the *nonproportional hazards model*

$$h(t \mid \boldsymbol{x}, \boldsymbol{\beta}(t)) = h_0(t) \exp\left(\boldsymbol{\beta}(t)^{\mathsf{T}}\, \boldsymbol{x}\right), \tag{7.8}$$

where $\boldsymbol{\beta}(t) = (\beta_1(t), \ldots, \beta_p(t))^{\mathsf{T}}$ is the p-dimensional vector of time-varying regression coefficients associated with the vector of static covariates $\boldsymbol{x} = (x_1, \ldots, x_p)^{\mathsf{T}}$. Each component $\beta_j(t)$ of $\boldsymbol{\beta}(t)$ is a smooth function of t which defines the logarithm of the hazard ratio at time t corresponding to a unit increase in x_j. We call $\beta_j(t)$ a *coefficient function*. The model in (7.8) has been studied by a number of authors, for example Zucker & Karr (1990), Grambsch & Therneau (1994), Gray (1994), and more recently by Kauermann (2005), Tian et al. (2005), and Martinussen & Scheike (2006). An additional advantage of a model which is not constrained by the PH assumption, is that the model itself could be used as a test for proportional hazards: constant covariate effects would be indicative of proportional hazards.

We model each coefficient function $\beta_j(t)$ in (7.8) using a cubic spline with $\mathcal{K}$ knots parametrized according to the VFDP. A goal of our approach is to avoid

making any assumptions about the baseline hazard function. In principle, an extra spline function could be used to estimate $\log\{h_0(t)\}$, and inference be based on an approximate full likelihood function. This is the approach followed in, for example, Kneib & Fahrmeir (2007). However, because our main interest lies in the coefficient functions $\beta_j(t)$, $j = 1, \ldots, p$, we shall instead treat $h_0(t)$ as a nuisance parameter and resort to partial likelihood based inference as originally proposed by Cox (1972). Furthermore, using the partial likelihood function simplifies the implementation of the MCMC algorithm described in the next section.

The partial likelihood function for the nonproportional hazards model in (7.8) becomes:

$$L_p(\boldsymbol{\beta}(t)\,; D) = \prod_{f=1}^{n} \frac{\exp\left(\boldsymbol{\beta}(\tilde{t}_f)^\intercal\, \tilde{\boldsymbol{x}}_f\right)}{\sum_{l \in R_f} \exp\left(\boldsymbol{\beta}(\tilde{t}_f)^\intercal\, \boldsymbol{x}_l\right)}\,, \tag{7.9}$$

with $\tilde{\boldsymbol{x}}_f$ as in (7.7). Because only information at observed failures contributes to the partial-likelihood, we place the knots $\{k_m\}_{m=1}^{\mathcal{K}}$ in the following way: we fix $k_1 = 0$, the time origin, and $k_{\mathcal{K}} = \max\{t_i\}$. For the remaining $\mathcal{K} - 2$ interior knots we take k_m, $m = 2, \ldots, \mathcal{K} - 1$, to be the $(m-1)/(\mathcal{K}-1)$ quantile of the observed failures. This ensures roughly the same amount of information between knots and thus stable estimates (Gray, 1992). Let $\boldsymbol{T}$ be the $n \times (2\mathcal{K})$ design matrix of the VFDP representation of the coefficient functions $\beta_j(t)$, $j = 1, \ldots, p$, as described in Section 4.4. Since the partial likelihood depends upon $\beta_j(t)$ only through $\beta_j(\tilde{t}_f)$, $f = 1, \ldots, n$, we build $\boldsymbol{T}$ using the aforementioned knot configuration $\{k_m\}_{m=1}^{\mathcal{K}}$ and the set of observed failures $\{\tilde{t}_f\}_{f=1}^{n}$. Hence, $\beta_j(\tilde{t}_f) = \boldsymbol{T}_{f\boldsymbol{.}}\, \boldsymbol{\alpha}_j$, with $\boldsymbol{\alpha}_j$ as defined in

Section 4.2. The partial likelihood in (7.9) can thus be rewritten as

$$L_p(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p \,; D) = \prod_{f=1}^{n} \frac{\exp\left(\sum_{j=1}^{p} \boldsymbol{T}_{f\boldsymbol{.}}\, \boldsymbol{\alpha}_j\, \tilde{x}_{fj}\right)}{\sum_{l \in R_f} \exp\left(\sum_{j=1}^{p} \boldsymbol{T}_{f\boldsymbol{.}}\, \boldsymbol{\alpha}_j\, x_{lj}\right)} \,. \tag{7.10}$$

The flexibility of the model (7.8) may, in some situations, not be needed for all covariates. Therefore we also consider the important semiparametric version of the model:

$$h(t \mid \boldsymbol{x}, \boldsymbol{\beta}(t), \boldsymbol{v}, \boldsymbol{\gamma}) = h_0(t) \exp\left(\boldsymbol{\beta}(t)^{\mathsf{T}} \boldsymbol{x} + \boldsymbol{\gamma}^{\mathsf{T}} \boldsymbol{v}\right), \tag{7.11}$$

where $\boldsymbol{v}$ is a q-dimensional vector of covariates with time-constant effects $\boldsymbol{\gamma}$. The semiparametric model (7.11) has the ability to summarize covariate effects as much as the data suggests. Also, survival analysis is typically a low informative framework due to the presence of censoring. Hence, for small sized data the fully nonparametric version of the extended Cox model in (7.8), with all covariate effects being time-varying, may further be difficult to fit. The semiparametric model (7.11) can thus achieve a more reasonable compromise between model complexity and size of the data.

7.4.2 Bayesian Inference

Consider the nonproportional hazards model in (7.8). Estimation of the coefficient functions $\beta_j(t)$ proceeds in the same way as that for the functions $g_j(x_j)$ in the GAM regression setting described in Chapter 6. Again we consider penalization of a spline curve $\beta(t)$ using both the single penalty $P_2(\beta(t)\,;\lambda) = \lambda \int \beta''(t)^2\, dt$, and the double penalty $P_{12}(\beta(t)\,;\lambda_1, \lambda_2) = \lambda_1 \int \beta'(t)^2\, dt + \lambda_2 \int \beta''(t)^2\, dt$. Let us focus, for

now, on the single penalty case. The prior specifications are described in detail in Subsection 5.2.1. The posterior distribution for the nonproportional hazards model (7.8) is

$$p\,(\boldsymbol{\theta} \mid D) \propto L_p(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p\,; D) \prod_j \lambda_j^{\mathrm{rk}(\boldsymbol{P}_2)/2} \exp\left(-\frac{\lambda_j}{2}\,\boldsymbol{\alpha}_j^{\intercal} \boldsymbol{P}_2\,\boldsymbol{\alpha}_j\right) \times \prod_j \lambda_j^{s_j - 1} \exp\left(-r_j \lambda_j\right), \quad (7.12)$$

where $\boldsymbol{\theta}$ is the vector of all the parameters in the model, and $L_p(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p\,; D)$ is the partial likelihood in (7.10).

We resort to MCMC simulation techniques to sample from the analytically intractable posterior in (7.12). The updating scheme based upon the local scoring algorithm proposed by Brezger & Lang (2006) (see Section 6.4) needs to be modified to account for the presence of the risk sets in the partial likelihood function (Hastie & Tibshirani, 1993). To see why note that here we wish to find the set of p spline parameter vectors $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p$ that maximizes the penalized partial log-likelihood criterion

$$\mathcal{J}_p(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p) = \ell_p(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p\,; D) - \tfrac{1}{2} \sum_{j=1}^{p} \lambda_j\,\boldsymbol{\alpha}_j^{\intercal} \boldsymbol{P}_2^{j}\,\boldsymbol{\alpha}_j\,. \quad (7.13)$$

Hastie & Tibshirani (1993) derive a modified local scoring algorithm to optimize a criterion similar to that in (7.13), in the case where the coefficient functions are modeled as cubic smoothing splines. Here we follow the same approach in order to find the solution to (7.13). Below we describe this algorithm in detail.

Let $\rho(f, i) = \sum_{j=1}^{p} \beta_j(\tilde{t}_f)\, x_{ij} = \sum_{j=1}^{p} \boldsymbol{T}_{f\boldsymbol{\cdot}}\,\boldsymbol{\alpha}_j\, x_{ij}$. The score vector associated with

(7.13) is now:

$$\begin{aligned}
\boldsymbol{U}_j &= \frac{\partial \mathcal{J}_p}{\partial \boldsymbol{\alpha}_j} \\
&= \sum_f \left\{ \boldsymbol{T}_{f\bullet}\, \tilde{x}_{fj} - \frac{\sum_{l \in R_f} \boldsymbol{T}_{f\bullet}\, x_{lj} \exp\left[\rho(f,l)\right]}{\sum_{l \in R_f} \exp\left[\rho(f,l)\right]} \right\} - \lambda_j \boldsymbol{P}_2^j\, \boldsymbol{\alpha}_j \\
&= \boldsymbol{T}^{\intercal}\, (\tilde{\boldsymbol{x}}_j - \bar{\boldsymbol{x}}_j) - \lambda_j \boldsymbol{P}_2^j\, \boldsymbol{\alpha}_j \,,
\end{aligned}$$

where $\tilde{\boldsymbol{x}}_j$ is the n-dimensional vector whose fth component is the value of the covariate j associated with $\tilde{t}_f$, and $\bar{\boldsymbol{x}}_j$ has fth component the weighted mean of x_j in the risk set R_f, i.e., $\bar{x}_{fj} = \sum_{l \in R_f} \zeta_{fl}\, x_{lj}$, $f = 1, \ldots, n$, where the weights ζ_{fl} represent the model probabilities

$$\zeta_{fl} = \frac{\exp\left[\rho(f,l)\right]}{\sum_{r \in R_f} \exp\left[\rho(f,r)\right]} = \frac{\exp\left(\sum_j \boldsymbol{T}_{f\bullet}\, \boldsymbol{\alpha}_j\, x_{lj}\right)}{\sum_{r \in R_f} \exp\left(\sum_j \boldsymbol{T}_{f\bullet}\, \boldsymbol{\alpha}_j\, x_{rj}\right)} \,.$$

Similar calculations show that the observed information matrix for the penalized partial log-likelihood in (7.13) has components

$$\begin{aligned}
\boldsymbol{\mathcal{I}}_{uj} &= -\frac{\partial^2 \mathcal{J}_p}{\partial \boldsymbol{\alpha}_u \partial \boldsymbol{\alpha}_j^{\intercal}} \\
&= \sum_f \left\{ \frac{\left(\sum_{l \in R_f} \boldsymbol{T}_{f\bullet}^{\intercal}\, \boldsymbol{T}_{f\bullet}\, x_{lu}\, x_{lj}\, \exp\left[\rho(f,l)\right]\right)}{\left(\sum_{l \in R_f} \exp\left[\rho(f,l)\right]\right)} - \right. \\
&\qquad \left. - \frac{\left(\sum_{l \in R_f} \boldsymbol{T}_{f\bullet}^{\intercal}\, x_{lj}\, \exp\left[\rho(f,l)\right]\right)\left(\sum_{l \in R_f} \boldsymbol{T}_{f\bullet}\, x_{lu}\, \exp\left[\rho(f,l)\right]\right)}{\left(\sum_{l \in R_f} \exp\left[\rho(f,l)\right]\right)^2} \right\} \\
&= \boldsymbol{T}^{\intercal}\, \boldsymbol{W}_{uj}\, \boldsymbol{T} \,,
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{\mathcal{I}}_{jj} &= -\frac{\partial^2 \mathcal{J}_p}{\partial \boldsymbol{\alpha}_j \partial \boldsymbol{\alpha}_j^{\intercal}} \\
&= \sum_f \left\{ \frac{\left(\sum_{l \in R_f} \boldsymbol{T}_{f\cdot}^{\intercal} \boldsymbol{T}_{f\cdot} \, x_{lj}^2 \, \exp\left[\rho(f,l)\right] \right)}{\left(\sum_{l \in R_f} \exp\left[\rho(f,l)\right] \right)} - \right. \\
&\qquad \left. - \frac{\left(\sum_{l \in R_f} \boldsymbol{T}_{f\cdot}^{\intercal} \, x_{lj} \, \exp\left[\rho(f,l)\right] \right) \left(\sum_{l \in R_f} \boldsymbol{T}_{f\cdot} \, x_{lj} \, \exp\left[\rho(f,l)\right] \right)}{\left(\sum_{l \in R_f} \exp\left[\rho(f,l)\right] \right)^2} \right\} + \lambda_j \boldsymbol{P}_2^j \\
&= \boldsymbol{T}^{\intercal} \boldsymbol{W}_{jj} \boldsymbol{T} + \lambda_j \boldsymbol{P}_2^j \,,
\end{aligned}$$

where the weight matrix $\boldsymbol{W}_{uj} = \operatorname{diag}(w_{uj1}, \ldots, w_{ujn})$ has elements the weighted covariances of (x_u, x_j) in the risk sets R_f,

$$w_{ujf} = \sum\nolimits_{l \in R_f} \zeta_{fl} \, x_{lu} \, x_{lj} - \left(\sum\nolimits_{l \in R_f} \zeta_{fl} \, x_{lu} \right) \left(\sum\nolimits_{l \in R_f} \zeta_{fl} \, x_{lj} \right), \qquad f = 1, \ldots, n \,.$$

The local scoring algorithm results in the following system of equations:

$$\begin{pmatrix} \boldsymbol{\alpha}_1 \\ \vdots \\ \boldsymbol{\alpha}_p \end{pmatrix} = \begin{pmatrix} \boldsymbol{S}_1 \left(\boldsymbol{z}_1 - [\boldsymbol{W}_{11}]^+ \sum_{j \neq 1} \boldsymbol{W}_{1j} \boldsymbol{T} \, \boldsymbol{\alpha}_j \right) \\ \vdots \\ \boldsymbol{S}_p \left(\boldsymbol{z}_p - [\boldsymbol{W}_{pp}]^+ \sum_{j \neq p} \boldsymbol{W}_{pj} \boldsymbol{T} \, \boldsymbol{\alpha}_j \right) \end{pmatrix}, \tag{7.14}$$

where

$$\boldsymbol{z}_j = [\boldsymbol{W}_{jj}]^+ \left(\tilde{\boldsymbol{x}}_j - \bar{\boldsymbol{x}}_j \right) + [\boldsymbol{W}_{jj}]^+ \sum_{u=1}^{p} \boldsymbol{W}_{ju} \boldsymbol{T} \, \boldsymbol{\alpha}_u \,,$$

and

$$\boldsymbol{S}_j = \left[\boldsymbol{T}^{\intercal} \boldsymbol{W}_{jj} \boldsymbol{T} + \lambda_j \, \boldsymbol{P}_2 \right]^{-1} \boldsymbol{T}^{\intercal} \boldsymbol{W}_{jj} \,.$$

The equations in (7.14) are used to define the parameters of the proposal density for the MCMC algorithm. Hence, if $\boldsymbol{\alpha}_j^c$ is the current vector of parameters defining the spline $\beta_j(t)$, a new value $\boldsymbol{\alpha}_j^p$ is proposed by drawing from the multivariate Gaussian proposal distribution $q\left(\boldsymbol{\alpha}_j^c, \boldsymbol{\alpha}_j^p\right)$ with covariance matrix and mean

$$\boldsymbol{\Sigma}_j = \left[\boldsymbol{T}^\intercal \boldsymbol{W}_{jj}^c \boldsymbol{T} + \lambda_j^c \boldsymbol{P}_2\right]^{-1}, \quad \boldsymbol{m}_j = \boldsymbol{\Sigma}_j \boldsymbol{T}^\intercal \boldsymbol{W}_{jj}^c \left(\boldsymbol{z}_j^c - \left[\boldsymbol{W}_{jj}^c\right]^+ \sum_{u \neq j} \boldsymbol{W}_{ju}^c \boldsymbol{T} \boldsymbol{\alpha}_u^c\right). \tag{7.15}$$

The complexity of the expressions in (7.15) means that the MCMC scheme will be more expensive computationally compared to that within the GAM regression context of Chapter 6. However, the resultant chains have good mixing properties and convergence is usually fast. The smoothing parameters λ_j are updated through Gibbs steps as described in Section 6.4. For the semiparametric model in (7.11) a similar MCMC scheme to the one described above is available (see Section 6.4).

Double penalty models can be particularly useful in the context of nonproportional hazards models. The coefficient functions $\beta_j(t)$ in (7.8) generalize constant functions of t corresponding to the proportional hazards model. It therefore seems reasonable to consider penalty functionals that shrink towards a constant (Gray, 1992). This can be achieved using the double penalty $P_{12}(\beta_j(t)\,; \lambda_j^1, \lambda_j^2)$. Further, Gray (1992) argues that cubic splines with the penalty $P_2(\beta_j(t)\,; \lambda) = \frac{\lambda}{2} \int \beta_j''(t)^2\, dt$ may result in unstable estimates at the right tail because data there tend to be sparse due to censoring. The inference process for double penalty models mimics the one described in Subsection 5.2.2 and Section 6.4, apart from the changes to the updating scheme for the spline parameter vectors $\boldsymbol{\alpha}_j$ describe above.

7.4.3 Predicting Individual Survival

In survival analysis it is usually of interest to predict the survival experience of an individual with given covariate vector $\boldsymbol{x}$. More specifically, we would like to know this individual's survival probability beyond some time point t. This information is summarized in the survival function

$$S(t \mid \boldsymbol{x}, \boldsymbol{\beta}) = \Pr\left(T \geq t \mid \boldsymbol{x}, \boldsymbol{\beta}\right) = \exp\left\{-\int_0^t h\left(u \mid \boldsymbol{x}, \boldsymbol{\beta}\right) du\right\}.$$

For the nonproportional hazards model in (7.8) we have that

$$S(t \mid \boldsymbol{x}, \boldsymbol{\beta}(t)) = \exp\left\{-\int_0^t h_0(u) \exp\left(\boldsymbol{\beta}(u)^{\intercal}\boldsymbol{x}\right) du\right\}. \tag{7.16}$$

An estimate of $S(t \mid \boldsymbol{x}, \boldsymbol{\beta}(t))$ can be obtained from estimates of $h_0(t)$ and $\beta_j(t)$, $j = 1, \ldots, p$. The parameter vector $\boldsymbol{\alpha}_j$ defining the coefficient function $\beta_j(t)$ is estimated using the Bayesian framework based upon the partial likelihood described in Subsection 7.4.2, where the baseline hazard $h_0(t)$ was treated as a nuisance parameter. Sinha et al. (2003), within a partial likelihood setting, obtain a Bayesian estimate of the cumulative baseline hazard by taking it to be, a priori, a gamma process. Here we follow a much simpler approach by using maximum likelihood techniques.

The full likelihood of the model in (7.8) can be written in terms of the hazard and survival functions as follows

$$L(\boldsymbol{\beta}(t), h_0(t)\,; D) = \prod_{f=0}^{n} \left\{ \prod_{r \in F_f} h(\tilde{t}_f \mid \tilde{\boldsymbol{x}}_r, \boldsymbol{\beta}(\tilde{t}_f))\, S(\tilde{t}_f \mid \tilde{\boldsymbol{x}}_r, \boldsymbol{\beta}(\tilde{t}_f)) \prod_{c \in C_f} S(t_c \mid \boldsymbol{x}_c, \boldsymbol{\beta}(t_c)) \right\}, \tag{7.17}$$

where F_f is the set of labels associated with individuals who fail at $\tilde{t}_f$, and C_f is the set of labels corresponding to individuals censored in $(\tilde{t}_f, \tilde{t}_{f+1})$, $f = 0, \ldots, n$, with $\tilde{t}_0 = 0$ and $\tilde{t}_{n+1} = \max\{t_i\}$. Following Breslow (1972), we adopt the convention that all censored observations are censored at the preceding observed failure time. The set F_0 is empty and F_f, $f = 1, \ldots, n$, has only one element since we assume that no tied observations exist in the data.

In the discussion of Cox's paper, Breslow (1972) suggested taking $h_0(t)$ to be a left continuous step function with jumps possibly only at the points in time where failures occurred, i.e.

$$h_0(t) = h_{0f}, \qquad \tilde{t}_{f-1} < t \leq \tilde{t}_f\,, \quad f = 1, \ldots, n\,.$$

Note that Breslow's model for the baseline hazard follows essentially from the same argument used to derive the partial likelihood in Section 7.3, i.e., that the gaps between failures contribute no information about the regression coefficients.

Let the last observed failure before time t be $\tilde{t}_L$. The integral in (7.16) can be written as

$$\int_0^t h_0(u) \exp\left(\boldsymbol{\beta}(u)^\intercal \boldsymbol{x}\right) du = \sum_{f:\, \tilde{t}_f \leq \tilde{t}_L} \int_{\tilde{t}_{f-1}}^{\tilde{t}_f} h_{0f} \exp\left(\boldsymbol{\beta}(u)^\intercal \boldsymbol{x}\right) du\,. \tag{7.18}$$

The integrals in (7.18) are not analytically tractable and therefore have to be estimated through numerical integration. Here we simply follow Kauermann (2005) and apply the trapezium rule

$$\int_{\tilde{t}_{f-1}}^{\tilde{t}_f} \exp\left(\boldsymbol{\beta}(u)^\intercal \boldsymbol{x}\right) du \approx \left(\tilde{t}_f - \tilde{t}_{f-1}\right) \frac{\exp\left(\boldsymbol{\beta}(\tilde{t}_f)^\intercal \boldsymbol{x}\right) + \exp\left(\boldsymbol{\beta}(\tilde{t}_{f-1})^\intercal \boldsymbol{x}\right)}{2}\,. \tag{7.19}$$

If $\widehat{\boldsymbol{\alpha}}_j$ is the posterior mean estimate obtained from the MCMC algorithm of Subsection 7.4.2, we replace the components $\beta_j(\tilde{t}_{f-1})$ and $\beta_j(\tilde{t}_f)$ of $\boldsymbol{\beta}(t)$ in (7.19) by their estimates, $\boldsymbol{T}_{f-1,\boldsymbol{.}}\,\widehat{\boldsymbol{\alpha}}_j$ and $\boldsymbol{T}_{f\boldsymbol{.}}\,\widehat{\boldsymbol{\alpha}}_j$ respectively. Conditional on $\widehat{\boldsymbol{\alpha}}_1, \ldots, \widehat{\boldsymbol{\alpha}}_p$, differentiation of (7.17) with respect to h_{0f}, $f = 1, \ldots, n$, yields the profile maximum likelihood estimate

$$\widehat{h}_{0f} = \left[(\tilde{t}_f - \tilde{t}_{f-1}) \sum_{l \in R_f} \frac{\exp\left(\sum_{j=1}^{p} \boldsymbol{T}_{f\boldsymbol{.}}\,\widehat{\boldsymbol{\alpha}}_j\, x_{lj}\right) + \exp\left(\sum_{j=1}^{p} \boldsymbol{T}_{f-1,\boldsymbol{.}}\,\widehat{\boldsymbol{\alpha}}_j\, x_{lj}\right)}{2} \right]^{-1},$$

resulting in the following estimate for $S(t \mid \boldsymbol{x}, \widehat{\boldsymbol{\beta}}(t))$

$$\widehat{S}(t \mid \boldsymbol{x}, \widehat{\boldsymbol{\beta}}(t)) = \prod_{f:\, \tilde{t}_f \le \tilde{t}_L} \exp\left\{ - \frac{\exp\left(\sum_{j=1}^{p} \boldsymbol{T}_{f\boldsymbol{.}}\,\widehat{\boldsymbol{\alpha}}_j\, x_j\right) + \exp\left(\sum_{j=1}^{p} \boldsymbol{T}_{f-1,\boldsymbol{.}}\,\widehat{\boldsymbol{\alpha}}_j\, x_j\right)}{\sum_{l \in R_f} \left[\exp\left(\sum_{j=1}^{p} \boldsymbol{T}_{f\boldsymbol{.}}\,\widehat{\boldsymbol{\alpha}}_j\, x_{lj}\right) + \exp\left(\sum_{j=1}^{p} \boldsymbol{T}_{f-1,\boldsymbol{.}}\,\widehat{\boldsymbol{\alpha}}_j\, x_{lj}\right)\right]} \right\}.$$

7.5 Primary Biliary Cirrhosis Data

We apply the proposed methodology to the PBC data set described in Fleming & Harrington (1991, Appendix D). It results from a Mayo Clinic trial in primary biliary cirrhosis (PBC) of the liver conducted between 1974 and 1984. PBC is a fatal chronic liver disease of unknown cause. The data set contains measurements on 418 individuals. Besides the patient's survival time and censoring indicator, 17 potential prognostic factors were recorded. These include clinical, biochemical, serologic, and histologic measurements made at the time of randomization to one of the two treatments: placebo or D-penicillamine. See Fleming & Harrington (1991) for a detailed description of the study. We focus here upon the five covariates found

to be important by Fleming & Harrington (1991): age (`age` - in years), edema (`edema` - 0 if no edema, 1 if edema is present), log (albumin) (`albumin` - in gm/dl), log (bilirubin) (`bilirubin` - in mg/dl), and log (prothrombin time) (`protime` - in seconds).

In the original analysis based on the Cox's PH model, Fleming & Harrington (1991) found evidence that both `edema` and `protime` did not satisfy the PH assumption: their initial effects disappeared with time, thus contradicting the PH assumption. Several authors have since relied on the PBC data set to illustrate a wide range of regression tools, mainly within a Cox regression setting. Abrahamowicz et al. (1996) focused on the time-varying effect of the predictor `protime`. They used regression splines to model the hazard ratio as a flexible function of time and found essentially the same pattern as Fleming & Harrington (1991). Tian et al. (2005) developed a kernel-weighted partial likelihood approach. Through the use of diagnostic tools the authors reached the conclusion that the effect of `bilirubin` also varied with time.

We start our analysis by considering the fully nonparametric model in (7.8). Hence, the hazard model is

$$h\left(t \mid \boldsymbol{x}, \boldsymbol{\beta}(t)\right) = h_0(t) \exp\left(\boldsymbol{\beta}(t)^{\mathsf{T}} \boldsymbol{x}\right),$$

where $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5)^{\mathsf{T}} = (\texttt{age}, \texttt{edema}, \texttt{albumin}, \texttt{bilirubin}, \texttt{protime})^{\mathsf{T}}$, and $\boldsymbol{\beta}(t) = (\beta_1(t), \beta_2(t), \beta_3(t), \beta_4(t), \beta_5(t))^{\mathsf{T}}$.

In total there were 160 deaths (approximately 60% censoring). We exclude the three individuals for whom information on one or more of the above selected covariates is missing. We center all continuous covariates around their means and

randomly break the ties at each iteration of the MCMC sampler.

All five covariate effects are modeled as time-varying regression coefficients using cubic splines with $\mathcal{K} = 8$ knots and represented through the VFDP. Let $\boldsymbol{\alpha}_j$ be the parameter vector associated with the spline $\beta_j(t)$, $j = 1, \ldots, 5$. In the single penalty case we need to set the value for the parameters in the prior for λ_j as discussed in Subsection 6.4. We investigated how sensitive the results were with respect to the choice of s_j and r_j and concluded that, at least in the application we considered here, the values of these parameters affected the posterior distribution of the smoothing parameters. Even though the value of $\check{s}_j$ in (5.6) in the full conditional for λ_j is dominated by $\mathrm{rk}(\boldsymbol{P}_2)$, the same is not true for $\check{r}_j$. Note that any change in the effect of a prognostic factor on the patient's survival is likely to be very mild, yielding $\boldsymbol{\alpha}_j^{\mathrm{T}} \boldsymbol{P}_2 \boldsymbol{\alpha}_j \approx 0$ in (5.6). Hence, $\check{r}_j$ is influenced by its value a priori. Values of $r_j = 10^{-6}$ or larger yield too small posterior estimates for λ_j, resulting in less smooth effects. For values of r_j smaller than 10^{-6} the posterior for λ_j becomes quite robust. The best results in terms of the visual display of the estimates were obtained setting $s_j = 1$ and $r_j = 10^{-7}$, $j = 1, \ldots, 5$. The posterior distribution of the model is still proper (Hennerfeind et al., 2006).

For double penalty models, we have to pre-specify a grid of values for λ_j^1 and λ_j^2. The grid $10^{-5}, 10^{-4}, \ldots, 10^4, 10^5$ provided satisfactory results. We denote by $\widehat{\beta}_j(t)^{\mathrm{SP}}$ and $\widehat{\beta}_j(t)^{\mathrm{DP}}$ the posterior mean estimates of the time-varying regression coefficient functions obtained using the single and double penalty models based upon the penalty functionals $P_2(\beta_j(t)\,;\lambda_j)$ and $P_{12}(\beta_j(t)\,;\lambda_j^1, \lambda_j^2)$ respectively.

The estimated coefficient functions in Figure 7.1 were obtained using the output of a chain of length 100,000 for the spline parameters $\boldsymbol{\alpha}_j$, $j = 1, \ldots, 5$ (after an initial burn-in period of length 2,000). Convergence of the chain was determined by

examining the plot of its path.

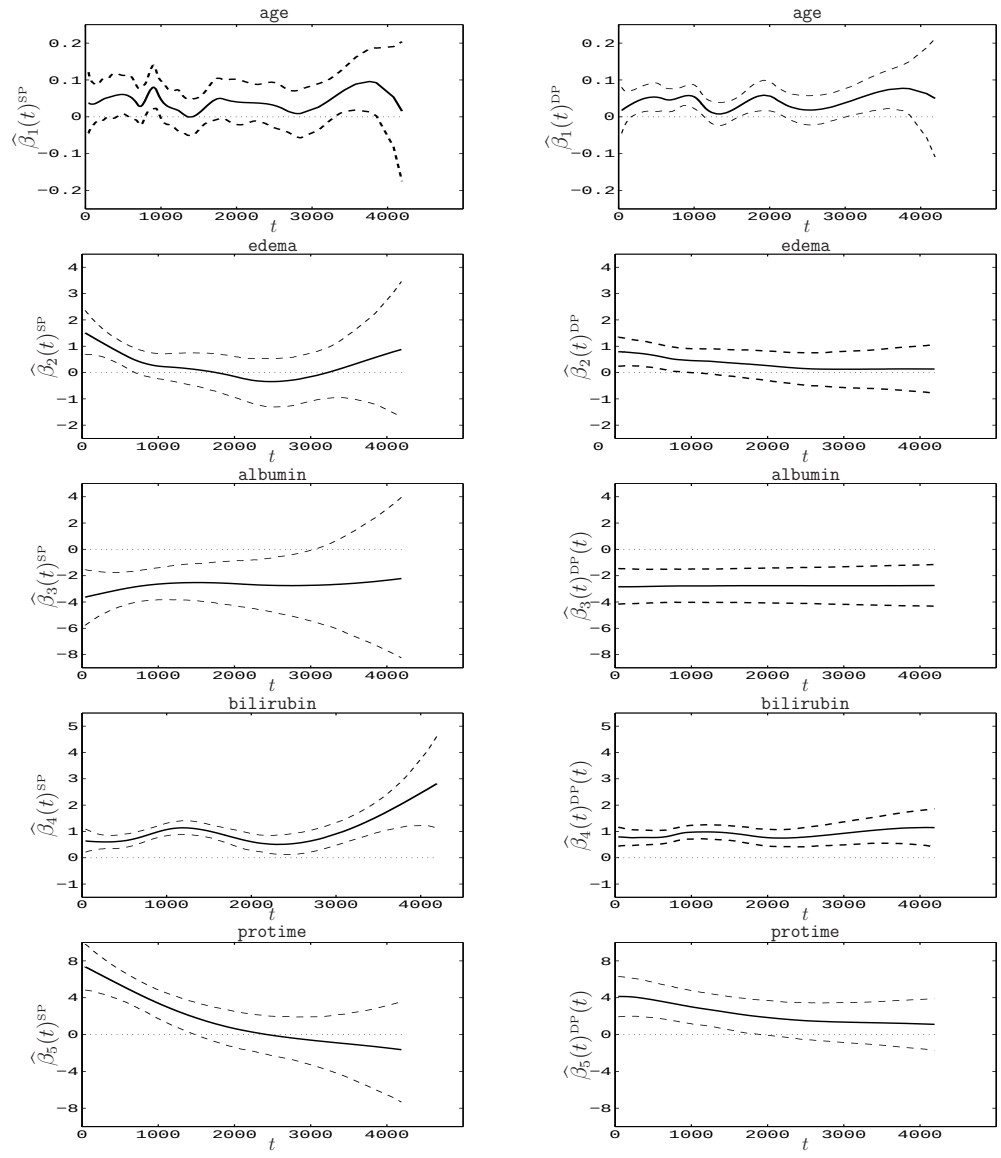

Figure 7.1: Posterior mean estimates of the time-varying regression coefficients as a function of time t (in days) for the PBC data set (solid line), together with 95% *pointwise credibility intervals (dashed line) using both the single (left column) and double (right column) penalty models.*

The plots in Figure 7.1 suggest that the effects of `age` and `albumin` on survival are

essentially constant throughout the study period. Older patients, with lower values of `albumin`, have worst survival prognosis. The regression coefficients associated with covariates `edema`, `bilirubin` and `protime` seem to vary with time. For both `edema` and `protime` this variation can be characterized as a loss of prognosis ability as the follow-up time increases. Initially, the presence of edema and larger values of prothrombin time have a negative effect on survival, but this eventually vanishes as time progresses. Regarding `bilirubin`, higher values lead to worst survival, especially around 1,000 days of follow-up.

The AIC criterion can be used to check whether time-varying effects lead to an improved fit compared to time-constant ones. In Table 7.1 the values for the AIC criterion for a selected number of models are shown. The semiparametric models

Table 7.1: Degrees of freedom (d.f.) and Akaike's information criteria (AIC) for different model specifications.

Penalty	`edema`	`protime`	`albumin`	`bilirubin`	`age`	d.f.	AIC
SP	TV	TV	TV	TV	TV	20.3	1503.3
SP	TV	TV	Const	TV	TV	20.7	1489.3
SP	TV	TV	TV	Const	TV	17.5	1526.2
SP	TV	TV	Const	Const	Const	7.1	1517.1
SP	Const	TV	Const	Const	Const	2.6	1518.6
SP	TV	Const	Const	Const	Const	4.4	1533.3
DP	TV	TV	TV	TV	TV	26.2	1507.5
DP	TV	TV	Const	TV	TV	26.4	1493.6
DP	TV	TV	TV	Const	TV	19.2	1526.4
.	Const	Const	Const	Const	Const	1.0	1538.2

SP, Single Penalty; DP, Double Penalty; TV, Time-varying; Const, Constant.

in Table 7.1, involving both time-constant and time-varying effects, correspond to versions of the model in (7.11). The model with all covariates but `albumin` having time-varying regression coefficients yields the lowest value for the AIC criterion, and thus seems to provide the best fit to the data (rows 2 and 8 in Table 7.1). In particular, time-varying effects for `edema` and `protime` lead to a better fit in terms of AIC compared to a proportional hazards model (rows 5, 6 and 10 in Table 7.1). The AIC results in Table 7.1 also seem to suggest that the effect of `bilirubin` on the hazard does change with time as suggested from the plots in Figure 7.1. Models with time-varying effect of `bilirubin` yield a lower value than models with time-constant effect (rows 1, 3, 7 and 9 in Table 7.1).

In general, the 95% pointwise credible intervals are narrower at the tails for double penalty estimates when compared to those obtained with a single penalty functional. This can arise, as here, when the data are compatible with a (nearly) time-independent predictor effect, in which case the double penalty shrinks the estimate towards a horizontal line.

Note that the estimate $\widehat{\beta}_3(t)^{\text{DP}}$ clearly supports the proportional hazards assumption, as suggested in Table 7.1, whereas $\widehat{\beta}_3(t)^{\text{SP}}$ starts off constant but becomes non-significant at the end of the follow-up. The fact that proportional hazards models arise as the smoothing limit of the double penalty model in (3.9) may help to explain this apparent difference between $\widehat{\beta}_3(t)^{\text{SP}}$ and $\widehat{\beta}_3(t)^{\text{DP}}$. The foregoing discussion highlights one further advantage of a model which is not constrained by the PH assumption. The model in

The seeming increase in risk for patients with high values of `bilirubin` around 1,000 days is probably the result of the large number of deaths between 500 and 1,500 days of follow-up (69, almost 44% of the total number of deaths). These correspond

to patients that tend to have higher values of `bilirubin` than average. This is illustrated in Figure 7.2 (left plot), where the values of `bilirubin` corresponding to failures are plotted against time. A 'lowess' smooth is also shown. The increase in the values of `bilirubin` around day 1,000 is clear from this plot.

The estimates for the effect of `age` show considerable variation, even when a double penalty functional is used. The fact that the values for covariate `age` are evenly spread around their mean might explain the apparent excessive uncertainty when estimating $\beta_1(t)$. This can be seen in Figure 7.2 (right plot), which displays the values of `age` associated with observed failures vs. the survival time, together with a 'lowess' smooth which is essentially flat. Nevertheless, one can still plausibly fit a constant function of time.

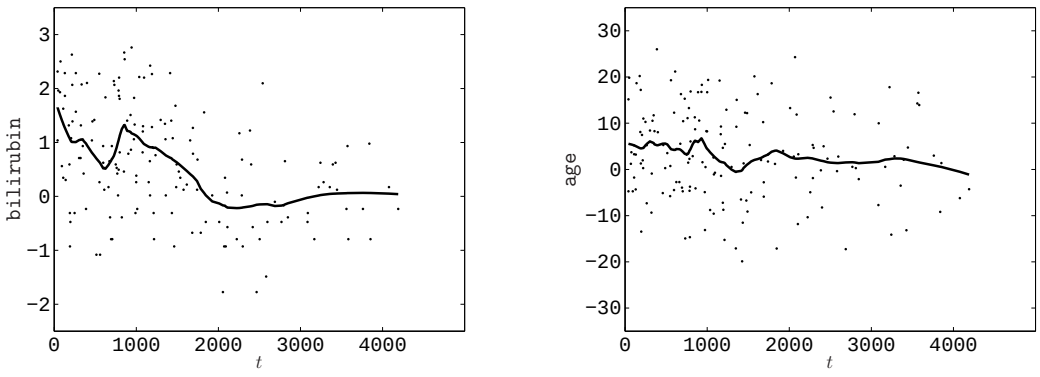

Figure 7.2: ***bilirubin*** *(left plot) and* ***age*** *(right plot) corresponding to observed failures vs time t (in days). The solid lines in both plots correspond to 'lowess' smooths.*

Overall, coefficient functions estimated using the double penalty model seem to have a longer, more stable effect on survival than those obtained under the single penalty formulation. Figure 7.3 represents the survival curve estimates for an average patient with and without edema for the two penalized spline models we studied.

The loss of prognosis ability of the covariate `edema` is clear for the single penalty model estimate (Figure 7.3, left plot). Here, the two curves start off apart, with the presence of edema leading to lower survival probability, but eventually collide later in the follow up, when `edema` is no longer a significant predictor. This effect is less visible for the double penalty estimate, as the two survival curves remain apart throughout the observation period (Figure 7.3, right plot).

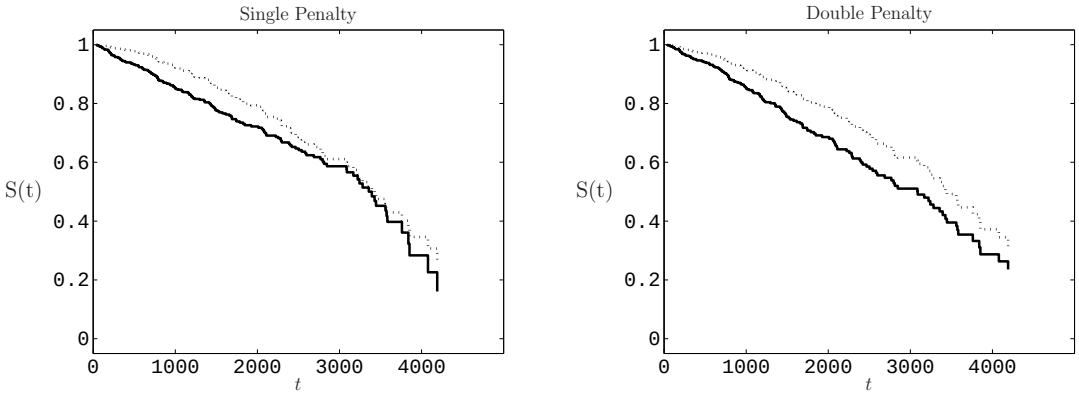

Figure 7.3: Estimated survival function for a 51-year-old patient with 3.5 gm/dl albumin, 1.7 mg/dl bilirubin, 10.6 seconds of prothrombin time with edema (solid line), and no edema (dotted line), using the single penalty (left plot) and double penalty (right plot) models.

Note that the penalty $P_1(\beta(t)\,;\lambda) = \lambda \int \beta'(t)^2\,dt$ also shrinks the estimated coefficient functions towards a constant function of time. However, it resulted in spline estimates considerably more wiggly than those obtained using the double penalty $P_{12}(\beta(t)\,;\lambda_1,\lambda_2)$. In addition, the credible intervals were very wide, even compared to the ones associated with the single penalty model based upon $P_2(\beta(t)\,;\lambda)$. This is illustrated in Figure 7.4, where plots of the estimated coefficient functions for the predictors `edema` and `protime` are presented. Hence, we conclude that explicit penalization of the curvature of the cubic spline function leads to better estimates.

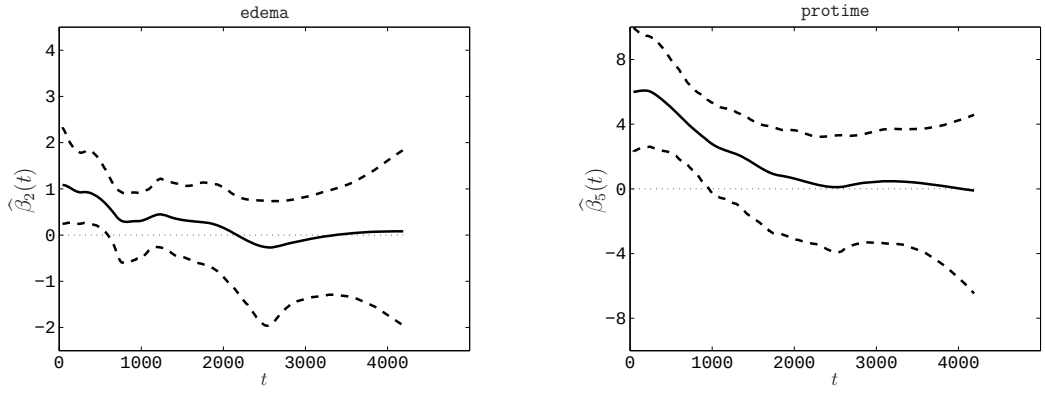

Figure 7.4: Posterior mean estimates of the time-varying regression coefficients for covariates `edema` *and* `protime` *using the penalty functional* $P_1(\beta(t)\,;\lambda)$ *(solid line), together with* 95% *pointwise credibility intervals (dashed line).*

7.6 Summary

The proposed methodology has been applied to the framework of nonproportional hazards model. Covariate effects were modeled as smooth functions of time through the VFDP. Bayesian inference was based upon the partial likelihood and we derived the parameters of the proposal density in the MCMC algorithm. The baseline hazard has been conveniently approximated by a piecewise constant function. We have considered both single and double penalty models. The latter yielded estimates with better behavior at the right tail of the distribution.

Chapter 8

Further Topics

This thesis has dealt so far with univariate spline functions estimated using a single smoothing parameter for each penalty functional within a penalized likelihood context. Whilst these have proved useful in a variety of applications, of which we gave two distinct examples in Chapter 6 and Chapter 7, there are situations where one may be interested in generalizations of the aforementioned methodology. Also, Bayesian inference for double penalty models as described in Chapter 5 relied upon empirical Bayes methods. Though this resulted in plausible estimated spline functions, a full Bayesian analysis taking into account all the uncertainty in the model is desirable.

Below we propose three directions for future research that attempt to generalize further the material presented in the current thesis by focusing upon the aforementioned limitations of the proposed methodology.

8.1 Spatially Adaptive Smoothing

Penalized splines with a single smoothing parameter can model nonlinear relationships quite well, provided that the curvature is not too heterogeneous. The nonlinearity present in the examples studied in this thesis is adequately handled through smoothers of this type and the models enjoy the benefits of simplicity and ease of fit of the proposed methodology. Yet in some application areas, such as speech recognition and neuroscience, varying amounts of curvature are the norm. For example, the function to be estimated may exhibit spatial heterogeneity in that it oscillates rapidly in some regions and is rather smooth in others. A literature on *spatially adaptive smoothing* has emerged to deal with such data.

Several authors have proposed penalized spline models with spatially adaptive smoothing. The approaches differ in the way the adaptive smoothing parameter $\lambda(x)$ is modeled as a function of the variable x of interest. Ruppert & Carroll (2000) estimate $\lambda(x)$ at a set of *subknots* and use linear interpolation to define $\lambda(x)$ everywhere on the domain of x. A Bayesian version of this procedure is given in Baladandayuthapani et al. (2005). Crainiceanu et al. (2007) extended the adaptive smoothing idea to adaptive error variance in models with heterogeneous error dispersion.

Pintore et al. (2006) model $\lambda(x)$ as a piecewise constant function with jumps at a set of subknots. They show that such choice for $\lambda(x)$ is convenient from a computational point of view since it provides closed-form solutions in a reproducing kernel Hilbert space framework (Wahba, 1990).

To model $\lambda(x)$ as a piecewise constant function is also a convenient choice if the spline function to be estimated is represented using the VFDP. In this case take $\mathcal{R}$,

the number of different regions of smoothing, to be such that $\mathcal{K} - 1 = c\,\mathcal{R}$, $c \in \mathbb{N}$. Each such region spreads over c adjacent knot intervals. In this case, the set of subknots is a subset of the original $\mathcal{K}$ knots. Let λ_u be the value of the smoothing parameter within the region $\mathcal{R}_u$, $u = 1, \ldots, \mathcal{R}$, and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{\mathcal{R}})$. Recall the penalty decomposition in (4.5). In the foregoing adaptive smoothing scenario, (4.5) can be re-expressed as follows:

$$P_2(g\,;\boldsymbol{\lambda}) = \sum_{u=1}^{\mathcal{R}} \lambda_u \left(\sum_{m=c\,(u-1)+1}^{c\,u} \boldsymbol{\alpha}_m{}^{\mathsf{T}} \boldsymbol{P}_{2m}\, \boldsymbol{\alpha}_m \right) ,$$

so that, for example, when $u = 1$ we have $\sum_{m=1}^{c} \boldsymbol{\alpha}_m{}^{\mathsf{T}} \boldsymbol{P}_{2m}\, \boldsymbol{\alpha}_m$, and when $u = 2$, $\sum_{m=c+1}^{2c} \boldsymbol{\alpha}_m{}^{\mathsf{T}} \boldsymbol{P}_{2m}\, \boldsymbol{\alpha}_m$.

Define $\boldsymbol{P}_2(\boldsymbol{\lambda})$ as the penalty matrix incorporating the smoothing parameters in $\boldsymbol{\lambda}$, and $\sum_{m=c\,(u-1)+1}^{c\,u} \boldsymbol{\alpha}_m{}^{\mathsf{T}} \boldsymbol{P}_{2m}\, \boldsymbol{\alpha}_m = \boldsymbol{\alpha}_u^{\mathsf{T}} \boldsymbol{P}_{2u}\, \boldsymbol{\alpha}_u$. The prior specification for the vector of spline parameters $\boldsymbol{\alpha}$ remains unchanged, i.e.,

$$p\,(\boldsymbol{\alpha} \mid \boldsymbol{\lambda}) \propto \exp\left(-\tfrac{1}{2}\, \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{P}_2(\boldsymbol{\lambda})\, \boldsymbol{\alpha}\right).$$

For the set of smoothing parameters in $\boldsymbol{\lambda}$ a correlated gamma process is assumed. Given $\lambda_1, \ldots, \lambda_{u-1}$, the prior for λ_u is

$$\lambda_u \mid \lambda_1, \ldots, \lambda_{u-1} \sim G\left(\omega, \frac{\omega}{\lambda_{u-1}}\right), \quad u = 1, \ldots, \mathcal{R}\,, \tag{8.1}$$

with $\lambda_0 = 1$, so that $\mathbb{E}[\lambda_u \mid \lambda_1, \ldots, \lambda_{u-1}] = \lambda_{u-1}$. The prior in (8.1) induces smoothness a priori across the parameters λ_u. The smaller the value of ω the less information a priori is assumed for smoothing the λ_u's.

Let $L(\boldsymbol{\alpha}\,; D)$ be the likelihood of the model at hand for a given data set D. The full conditional for $\boldsymbol{\alpha}$ remains unchanged,

$$p\left(\boldsymbol{\alpha} \mid D, \boldsymbol{\lambda}\right) \propto L(\boldsymbol{\alpha}\,; D) \exp\left(-\tfrac{1}{2}\,\boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{P}_2(\boldsymbol{\lambda})\,\boldsymbol{\alpha}\right),$$

whereas the full conditional for the smoothing parameters λ_u is now

$$p\left(\lambda_u \mid \boldsymbol{\alpha}, \boldsymbol{\lambda}_{(u)}\right) \propto f(\lambda_u) = \begin{cases} \lambda_u^{\frac{\mathrm{rk}(\boldsymbol{P}_{2u})}{2} - 1} \exp\left(-\lambda_u \left\{\tfrac{1}{2}\,\boldsymbol{\alpha}_u^{\mathsf{T}} \boldsymbol{P}_{2u}\,\boldsymbol{\alpha}_u + \frac{\omega}{\lambda_{u-1}}\right\}\right. \\ \qquad\qquad \left. - \frac{\omega}{\lambda_u}\,\lambda_{u+1}\right), \quad u = 1, \ldots, \mathcal{R}{-}1\,, \\ \lambda_{\mathcal{R}}^{\frac{\mathrm{rk}(\boldsymbol{P}_{2\mathcal{R}})}{2} + \omega} \\ \qquad \times \exp\left(-\lambda_{\mathcal{R}} \left\{\tfrac{1}{2}\,\boldsymbol{\alpha}_{\mathcal{R}}^{\mathsf{T}} \boldsymbol{P}_{2\mathcal{R}}\,\boldsymbol{\alpha}_{\mathcal{R}} + \frac{\omega}{\lambda_{\mathcal{R}-1}}\right\}\right), \quad u = \mathcal{R}\,, \end{cases} \tag{8.2}$$

where $\boldsymbol{\lambda}_{(u)}$ denotes the vector excluding the uth element, λ_u. The full conditional distribution in (8.2) is not of standard form. In order to sample from it one could use, for example, the ratio-of-uniforms method (Wakefield et al., 1991) as proposed in Qiou et al. (1999), since both $f(\lambda_u)$ and $\lambda_u^2 f(\lambda_u)$ are unimodal on the interval $(0, +\infty]$. We intend to explore the foregoing setting as well as related variants, specially regarding the prior specification for the smoothing parameters λ_u.

8.2 Bivariate Smoothing

The additive models described in Chapter 6 and Chapter 7 have several attractive features. For example, the individual component functions can be plotted separately to visualize the effect of each predictor upon the expected response. However, it is often the case where the values of two or more covariates in a regression model

are dependent upon each other. Hence, interaction terms are called for in order to obtain a better fit to the data.

Bivariate smoothing deals with interaction components of the form $g(X_u, X_v)$, where X_u and X_v are random variables whose values are likely to be correlated. As in the univariate smoothing scenario, no assumptions are made regarding the shape of the surface $g(X_u, X_v)$ other than it is smooth on some sense. Bivariate smoothing is of central interest in a number of application areas such as geography or public health.

The two most common approaches to bivariate smoothing are thin-plate splines (e.g., Wahba, 1990; Green & Silverman, 1994; Wood, 2003) and tensor products of spline basis functions (see, e.g., Gray, 1992; Wood, 2006b; Brezger & Lang, 2006).

An approach based upon a generalization of the ideas leading to the VFDP of Chapter 4 defines a *bicubic* surface $g(t, w)$ as:

$$g(t, w) = \sum_{i=0}^{3} \sum_{j=0}^{3} \nu_{ij}\, t^i w^j \,. \tag{8.3}$$

The surface $g(t, w)$ is continuous everywhere and has continuous first derivative, but its second derivative is allowed discontinuities at certain well defined regions of its domain.

As in the univariate case, the representation in (8.3) is not convenient from a computational point of view. Let $\{k_m^t\}_{m=1}^{\mathcal{K}_t}$ and $\{k_u^w\}_{u=1}^{\mathcal{K}_w}$ be sets of knots covering the domains of the variables t and w respectively. Within the rectangle $[k_m^t, k_{m+1}^t) \times$

$[k_u^w, k_{u+1}^w)$, the surface g is completely defined by the 16 parameters:

$$\begin{aligned} g_{ij} &= g(k_i^t, k_j^w)\,, \qquad & g_{ij}^t &= \frac{\partial g(k_i^t, k_j^w)}{\partial t}\,, \\ g_{ij}^w &= \frac{\partial g(k_i^t, k_j^w)}{\partial w}\,, \qquad & g_{ij}^{tw} &= \frac{\partial^2 g(k_i^t, k_j^w)}{\partial t \partial w}\,, \end{aligned} \tag{8.4}$$

with $i = m, m+1$, and $j = u, u+1$. The parameters g_{ij} define the height of the surface at the point with coordinates $\left\{k_i^t, k_j^w\right\}$. The values of g_{ij}^t and g_{ij}^w correspond to the tangents, in the t and w direction respectively, at the point $\left\{k_i^t, k_j^w\right\}$. Finally, the value of g_{ij}^{tw} controls the shape of the interior of the surface in the neighbourhood of the corner $\left\{k_i^t, k_j^w\right\}$.

Alternatively, the representation using the parameters in (8.4) can be seen as a tensor product of the elements in (4.2). For ease of notation let us redefine the polynomials in (4.2) as: $H_0^m(t) = \phi_{0m}(t)$, $H_1^m(t) = \psi_{0m}(t)$, $H_2^m(t) = \phi_{1m}(t)$, and $H_3^m(t) = \psi_{1m}(t)$, for $t \in [k_m^t, k_{m+1}^t)$; likewise for the variable w and the knot interval $[k_u^w, k_{u+1}^w)$. Within the rectangle $[k_m^t, k_{m+1}^t) \times [k_u^w, k_{u+1}^w)$, the tensor product representation of the surface g is:

$$g(t, w) = \sum_{i=0}^{3} \sum_{j=0}^{3} c_{ij}\, H_i^m(t)\, H_j^u(w)\,. \tag{8.5}$$

Note that, for example, $g\left(k_m^t, k_u^w\right) = c_{00} = g_{mu}$, and $g(k_m^t, k_{u+1}^w) = c_{02} = g_{mu+1}$, and that $\partial g\left(k_m^t, k_u^w\right)/\partial t = c_{11} = g_{mu}^t$, so that the coefficients c_{ij} of the tensor product representation of g in (8.5) are exactly the parameters in (8.4).

Smoothness can be attained by means of a suitable penalty functional as in the univariate case. For example, one could, within each rectangle $[k_m^t, k_{m+1}^t) \times [k_u^w, k_{u+1}^w)$, define the penalty on the curvature of g along the directions defined by

t and w:

$$P_2\left(g\,;\lambda\right) = \int_{k_m^t}^{k_{m+1}^t} \int_{k_u^w}^{k_{u+1}^w} \lambda_t \left(\frac{\partial^2 g}{\partial t^2}\right)^2 + \lambda_w \left(\frac{\partial^2 g}{\partial w^2}\right)^2 dw\, dt\,, \tag{8.6}$$

where λ_t and λ_w control smoothing in the direction defined by t and w respectively. See Wood (2006b) for details and the motivation behind the penalty in (8.6). It is likely that interpretable, simple expressions based upon the parameters in (8.4) exist for the penalty functional (8.6) and it would certainly be interesting to investigate this matter further.

8.3 Full Bayesian Inference for Double Penalty Models

The empirical Bayes methodology for double penalty functionals described in Chapter 5 does not take into account the uncertainty inherent to the estimation of the smoothing parameters when constructing pointwise credibility intervals for the estimated spline functions. Further research on a full Bayesian analysis for double penalty models is therefore desirable.

Chapter 5 exposed the difficulties in eliciting a prior for the smoothing parameter pairs $\{\lambda_1, \lambda_2\}$ associated with the penalty functional in (3.9). One possible strategy would be to ignore, a priori, the possible dependency between the values of λ_1 and λ_2, and to place independent, non-informative priors over λ_1 and λ_2. For example, a uniform with a large variance on the transformed parameters $\log\left(\lambda_1\right)$ and $\log\left(\lambda_2\right)$. However, the Gibbs sampler is no longer available to obtain draws from the posterior distribution of λ_1 and λ_2, as in the single penalty case. Hence, we need to define some sort of acceptance-rejection algorithm to sample from the posterior of interest. The

performance of such algorithm is likely to be hindered by the correlation between λ_1 and λ_2. One could try to reduce correlation by means of a reparametrization, for example, by replacing $\{\lambda_1, \lambda_2\}$ for $\{\lambda, p\}$, with $\lambda_1 = p\lambda$ and $\lambda_2 = (1-p)\lambda$, $p \in [0, 1]$. This is ongoing work and further research is required.

Chapter 9

Summary and Conclusions

9.1 Summary of the Thesis

Nonparametric regression methods based upon penalized splines (Eilers & Marx, 1996) are continually gaining in popularity (e.g., Fahrmeir et al., 2004; Kauermann, 2005; Baladandayuthapani et al., 2005; Kneib & Fahrmeir, 2007; Crainiceanu et al., 2007), because of their good approximation properties and computational efficiency. Most of the research has been developed for spline functions represented using either a truncated power basis (TPB) or a B-spline basis (see, e.g., Brezger & Lang, 2006; Crainiceanu et al., 2007). Whilst the form of TPB functions is easily understood, they may yield unstable estimates with poor numerical properties. B-splines overcome these numerical issues, but their analytical treatment is not straightforward, except in the special case of equally spaced knots as described in Eilers & Marx (1996). Also, apart from the work by Eilers and co-authors (Eilers & Marx, 2003; Eilers & Goeman, 2004), or that of Aldrin (2006), most of the activity has been on models estimated using a single penalty functional.

The current thesis has been mainly motivated by the increased research activity in applied and methodological aspects of the penalized spline regression approach. We aimed to widen the applicability of the approach by exploring a different parametrization for cubic spline functions, and by looking at other forms of penalization. In what follows we will briefly summarize the content of the preceding chapters, drawing together the results, before presenting the conclusions that can be drawn.

Chapter 2 summarized the relevant theoretical background on univariate spline functions. We presented the two most common spline parametrizations, the aforementioned TPF and the B-spline basis, discussing advantages and disadvantages of both representations. We then introduced the class of additive models, a generalization of the multiple linear regression model that retains some of its interpretability. The chapter concluded with a discussion on the motivation behind the use of spline functions to represent the unknown smooth terms in an additive model.

Chapter 3 started with a short literature review of the penalized likelihood methodology in curve fitting problems. We then investigated penalized log-likelihood criteria in detail. The standard approach achieves smoothness by means of a single penalty functional. For cubic spline functions, this is typically based upon the integrated squared second derivative of the curve. However, there exist situations where this does not specify the limit of smoothness appropriate for the regression problem at hand. The nonproportional hazards model studied in Chapter 7 is one such example. We therefore suggested using *double penalty functionals*, measures of roughness that combine two levels of penalization in order to define criteria with the adequate limit of shrinkage.

The chapter concluded with a discussion on the Bayesian equivalent to the penal-

ized log-likelihood criterion. The key idea developed there was that spline estimates optimizing the penalized log-likelihood criterion have an interpretation as posterior mode estimates.

In Chapter 4 we presented a parametrization for cubic spline functions that was intuitive in its elements and provided greater insight into the different levels of penalization imposed by standard measures of roughness. The basic ingredients of the parametrization are the values and first derivatives of the spline function at the knots. Splines defined through such parametrization are known in the numerical analysis literature as cubic Hermite spline. Here we have called it value-first derivative parametrization (VFDP), hence making explicit reference to its components.

We noted that the VFDP resulted in spline functions that were continuous and had continuous first derivatives, but that the second derivatives were allowed to be discontinuous at the knots. We believe that this characteristic of the parametrization induces extra flexibility in the fitting mechanism. We noted further that decomposing the penalty functionals according to the partition defined by the set of knots greatly facilitated their implementation, avoiding integrals or derivatives of basis functions altogether. This is because the resulting local penalties have simple algebraic expressions in terms of the components of the parametrization. Such expressions also provide a better insight into the different levels of penalization imposed over the spline curve. This can be useful for prior elicitation within a Bayesian inference framework.

Chapter 5 used simulation studies to compare the performance of the proposed methodology to that of the B-splines and difference penalties (Eilers & Marx, 1996). We also compared single and double penalty models. The analysis for single penalty models was fully Bayesian, but double penalty estimates were obtained using a

hybrid approach that combined empirical Bayes methods for the smoothing parameters with Monte Carlo estimates for the spline parameters obtained from MCMC outputs.

Estimates of spline functions built using the VFDP proved to be effective in recovering the underlying true function. Double penalty estimates using the VFDP compared favorably to single penalty ones in terms of lower mean squared error in situations of low spatial variability. Time constraints prevented us from conducting simulation studies regarding the coverage probability of spline estimates obtained using the VFDP. This is of great importance, as one would like to ensure that the right information is being captured from the data by the model. In addition, it would also be interesting to study the performance of the VFDP methodology in more challenging settings, for example, in the estimation of functions with changes in the forth derivative, as this determines the approximation error for standard cubic spline functions.

We considered two distinct applications for the proposed methodology. One concerned generalized additive models (GAMs), and the other nonproportional hazard models. Chapter 6 presented the flexible class of GAMs. The smooth terms in the model were represented through the VFDP. Bayesian inference using both the single and the double penalty approach relied upon MCMC sampling techniques using the ideas developed in Chapter 5. However, in GAMs, posterior sampling for the spline parameters used an efficient proposal scheme, based upon the local scoring algorithm.

The data application included in Chapter 6 concerned data from the 1985 current population survey in the U.S. We modeled these data through a semiparametric GAM with a logistic link function. The smooth terms in the GAM were estimated

considering both single and double penalty models, the latter providing a better fit to the data.

In Chapter 7 we focused upon nonproportional hazards models that allow covariate effects to vary smoothly with follow-up time. These coefficient functions were represented through the VFDP. Bayesian inference was based upon the partial likelihood function and followed closely that described in Chapter 6 for GAMs, though the parameters of the proposal in the MCMC algorithm became more complex. We applied the methodology to the well known primary biliary cirrhosis data set described in Fleming & Harrington (1991, App. D). We computed both single and double penalty estimates for the time-varying covariate effects and concluded that double penalty functionals resulted in estimates with better behavior at the right tail. Moreover, when the data clearly supported the proportional hazards assumption, the double penalty model yielded estimates close to a horizontal line. We compared several models using Akaike's information criterion and concluded that the model with all predictors but log (albumin) having time-varying effects provided the best fit to the data. The analysis allowed us to conclude that the effect of log (bilirubin) changed with time as it had been previously suggested in Tian et al. (2005).

We also investigated the penalty functional based upon the first derivative of the spline curve, since it achieves the same limit of smoothness as the double penalty model, and concluded that explicit penalization of the curvature of the cubic spline function yielded better estimates.

Chapter 8 proposed some topics for further research. These concerned the development of penalized spline regression models with adaptive smoothing and interactions terms. Furthermore, we also outlined ideas for the full Bayesian analysis of

double penalty models.

9.2 Final Remarks

This thesis explored an alternative parametrization for the flexible class of cubic spline functions. Such parametrization is defined locally and is not bonded to any particular configuration of knots. The elements of the parametrization relate naturally to the shape of the spline function, and allow extra flexibility when modeling data. It showed how standard penalty functionals and related extensions can be decomposed into interpretable expressions and highlighted the importance of correctly specifying the limit in the penalized likelihood criterion approach.

Appendix A

Some Complementary Results

In Section 4.3 we stated two equivalences between the degree of the polynomial that agrees with the spline g within the knot interval $[k_m,\, k_{m+1})$ and the values of the quantities $d_{m,m+1}$ and $d_{m+1,m}$ defined in (4.6):

$$\text{i)} \qquad d_{m,m+1} = d_{m+1,m} \quad \Leftrightarrow \quad g(x)\,,\ x \in [k_m,\, k_{m+1})\,,\ \text{is quadratic}\,,$$

$$\text{ii)} \qquad d_{m,m+1} = d_{m+1,m} = 0 \quad \Leftrightarrow \quad g(x)\,,\ x \in [k_m,\, k_{m+1})\,,\ \text{is linear}\,.$$

Recall from Section 4.2 that, by definition of a cubic spline, g matches a cubic polynomial within each knot interval. Thus, there exist real coefficients c_{0m}, c_{1m}, c_{2m}, and c_{3m} such that we can write

$$g(x) = c_{0m} + c_{1m}\, x + c_{2m}\, x^2 + c_{3m}\, x^3, \quad x \in [k_m,\, k_{m+1})\,.$$

We shall start by providing a proof to the equivalence in i). From the equality

$d_{m,m+1} = d_{m+1,m}$, it follows that:

$$\begin{aligned}
d_{m,m+1} = d_{m+1,m} \;&\Rightarrow\; a_{m+1} - a_m = \frac{\Delta_m}{2}\left(b_{m+1} + b_m\right) \\
&\Rightarrow\; g(k_{m+1}) - g(k_m) = \frac{\Delta_m}{2}\left(g'(k_{m+1}) + g'(k_m)\right) \\
&\Rightarrow\; c_{3m}\left(k_{m+1}^3 - k_m^3\right) + c_{2m}\left(k_{m+1}^2 - k_m^2\right) = \\
&\qquad\qquad = \frac{\Delta_m}{2}\left[3c_{3m}\left(k_{m+1}^2 + k_m^2\right) + 2c_{2m}\left(k_{m+1} + k_m\right)\right] \\
&\Rightarrow\; c_{3m}\left(k_{m+1} - k_m\right)^3 = 0 \\
&\Rightarrow\; c_{3m} = 0.
\end{aligned}$$

The last implication results from the fact that $k_{m+1} > k_m$. Given that $c_{3m} = 0$, the expression for $g(x)$ within $[k_m,\, k_{m+1})$ will be $g(x) = c_{0m} + c_{1m}\, x + c_{2m}\, x^2$, i.e., a polynomial of degree two.

Now, let $g(x) = c_{0m} + c_{1m}\, x + c_{2m}\, x^2$ be a polynomial of degree two within the knot interval $[k_m,\, k_{m+1})$. Then,

$$\begin{aligned}
d_{m,m+1} &= a_{m+1} - \left(a_m + \Delta_m b_m\right) \\
&= c_{0m} + c_{1m}\, k_{m+1} + c_{2m}\, k_{m+1}^2 - \\
&\qquad\qquad - \left[c_{0m} + c_{1m}\, k_m + c_{2m}\, k_m^2 + \Delta_m\left(c_{1m} + 2c_{2m}\, k_m\right)\right] \\
&= c_{2m}\left(k_{m+1} - k_m\right)^2 .
\end{aligned}$$

Similarly one can show that $d_{m+1,m} = c_{2m}\left(k_m - k_{m+1}\right)^2$, and hence that $d_{m,m+1} = d_{m+1,m}$.

The equivalence in ii) follows easily from that in i). From the condition that $d_{m,m+1} = d_{m+1,m}$ in ii) we can conclude, using i), that g is a quadratic polynomial

within $[k_m,\, k_{m+1})$. Thus, $g(x) = c_{0m} + c_{1m}\, x + c_{2m}\, x^2$, for $k_m \leq x < k_{m+1}$. Now,

$$\left\{ \begin{array}{lcl} d_{m,m+1} & = & 0 \\ \\ d_{m+1,m} & = & 0 \end{array} \right. \Rightarrow \left\{ \begin{array}{lcl} b_m & = & b_{m+1} \\ \\ — & & \end{array} \right. \Rightarrow \left\{ \begin{array}{lcl} c_{2m} & = & 0 \\ \\ — & & \end{array} \right. .$$

Therefore, $g(x) = c_{0m} + c_{1m}\, x$, for $k_m \leq x < k_{m+1}$. The reverse implication follows immediately.

Appendix B

Markov Chain Monte Carlo Algorithms

The application of Bayesian methods to the treatment of complex models was initially limited due to the mathematical intractability of most posterior distributions. During the last two decades, however, Monte Carlo based sampling methods for evaluating high-dimensional posterior integrals have been rapidly developing. Markov chain Monte Carlo (MCMC) is the most popular tool to obtain Monte Carlo estimates of parameters in a model. Two of the best known MCMC algorithms are the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970), and the Gibbs sampler (Geman & Geman, 1984). Chib & Greenberg (1995) and Casella & George (1990) provide excellent tutorials on Metropolis-Hastings algorithm and the Gibbs sampler respectively. For a general description of these and other Monte Carlo methods see, for example, Robert & Casella (2004). The next two sections describe the general form of the Gibbs sampler and the Metropolis-Hastings algorithm.

B.1 The Gibbs Sampler

The Gibbs sampler (Geman & Geman, 1984) is an MCMC algorithm where the transition kernel is formed by the full conditional distributions of the posterior of interest. In order to describe the algorithm let $p(\boldsymbol{\theta} \mid D)$ define the posterior distribution of interest, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)^\intercal$. Assume that the full conditional distribution of θ_i, $p(\theta_i \mid \cdot)$, $i = 1, \ldots, d$, is available. This means that it is completely known and can be sampled from. The Gibbs sampler is as follows:

Gibbs Sampler

(1) `Initialize:` $\boldsymbol{\theta}^{[0]} = (\theta_1^{[0]}, \ldots, \theta_d^{[0]})^\intercal$ `and set the chain counter` $j = 1$.

(2) `Obtain a new value` $\boldsymbol{\theta}^{[c+1]}$ `from` $\boldsymbol{\theta}^{[c]}$ `through the successive draws`

$$\begin{aligned}
\theta_1^{[c+1]} &\sim p(\theta_1 \mid \theta_2^{[c]}, \ldots, \theta_d^{[c]}, D) \\
\theta_2^{[c+1]} &\sim p(\theta_2 \mid \theta_1^{[c+1]}, \theta_3^{[c]}, \ldots, \theta_d^{[c]}, D) \\
&\vdots \\
\theta_d^{[c+1]} &\sim p(\theta_d \mid \theta_1^{[c+1]}, \ldots, \theta_{d-1}^{[c+1]}, D)
\end{aligned}$$

(3) `Change counter` j `to` $j + 1$ `and return to step` (2) `until convergence is reached.`

When convergence is reached, the sampled value $\boldsymbol{\theta}^{[c+1]}$ is a draw from $p(\boldsymbol{\theta} \mid D)$.

B.2 The Metropolis-Hastings Algorithm

The Metropolis algorithm was initially presented in a paper by Metropolis et al. (1953) and later generalized by Hastings (1970), resulting in the MCMC algorithm known as Metropolis-Hastings algorithm. Typically, but not necessarily, Metropolis-Hastings algorithms are implemented when the full conditional distributions of the posterior of interest are not of standard form and hence difficult to sample from. In this case, samples from a proposed distribution are drawn and accepted or not with a certain acceptance probability.

Let D represent the available data and $p(\boldsymbol{\theta} \mid D)$ be the posterior distribution of interest, with $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)^\intercal$. Assume that the full conditional distributions $p(\theta_i \mid \cdot)$, $i = 1, \ldots, d$, are difficult to sample from. Thus, a new value for θ_i is drawn from a proposal distribution conditional on the current value θ_i^c and defined by the transition density $q(\theta_i^c, \theta_i)$, so that the proposed value, θ_i^p, is sampled with probability $q(\theta_i^c, \theta_i^p)$. The Metropolis-Hastings algorithm comprehends the following steps:

Metropolis-Hastings Algorithm

(1) `Initialize:` $\boldsymbol{\theta}^c = (\theta_1^c, \ldots, \theta_d^c)^\intercal$ `and set the chain counter` $j = 1$.

(2) `For` $i = 1, \ldots, d$ `do:`

⋆ `Generate a proposal for` θ_i `from`

$$\theta_i^p \sim q(\theta_i^c, \theta_i)\,.$$

⋆ Accept the proposal with acceptance probability given by

$$\xi(\theta_i^c, \theta_i^p) = \min\left\{1, \frac{p\left(\theta_i^p \mid \cdot\right) q(\theta_i^p, \theta_i^c)}{p\left(\theta_i^c \mid \cdot\right) q(\theta_i^c, \theta_i^p)}\right\}.$$

(3) Change counter j to $j+1$ and return to step (2) until convergence is reached.

Appendix C

MCMC Output

The plots below correspond to selected Markov chain paths resulting from the analysis in the data applications presented in Chapter 6 and Chapter 7.

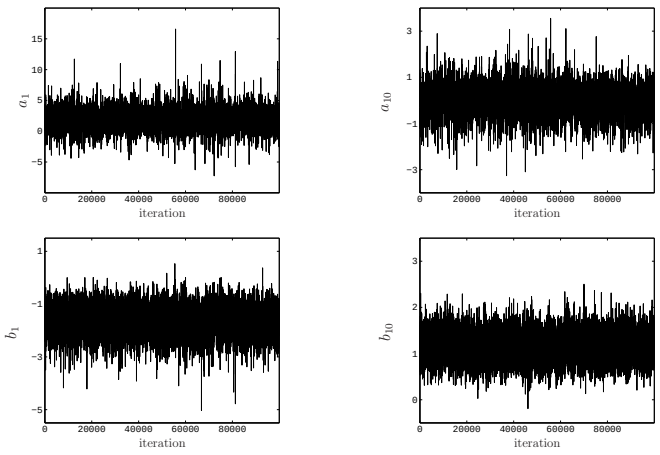

Figure C.1: A selection of chain paths from the Metropolis-Hastings algorithm. The parameters correspond to the nonlinear effect of covariate `wage` *when a single penalty model is used (**union membership data**).*

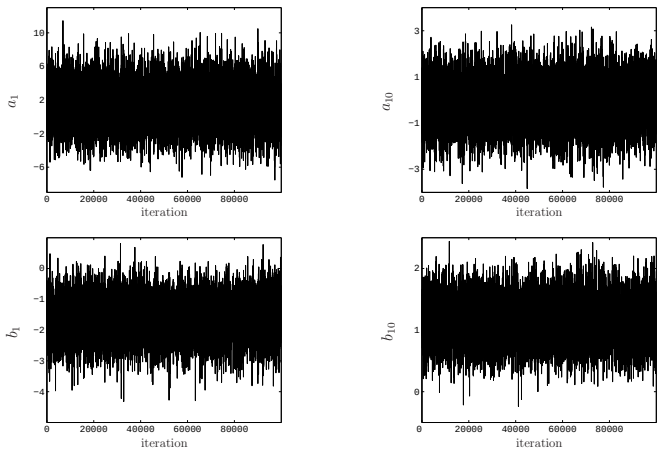

Figure C.2: A selection of chain paths from the Metropolis-Hastings algorithm. The parameters correspond to the nonlinear effect of covariate `wage` *when a double penalty model is used (**union membership data**).*

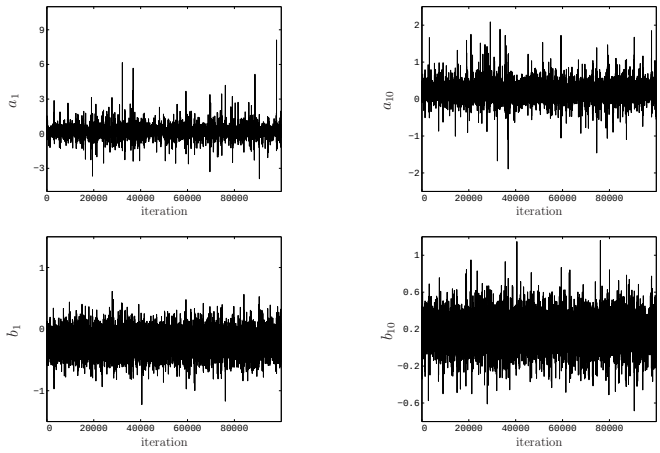

Figure C.3: A selection of chain paths from the Metropolis-Hastings algorithm. The parameters correspond to the nonlinear effect of covariate `age` *when a single penalty model is used (**union membership data**).*

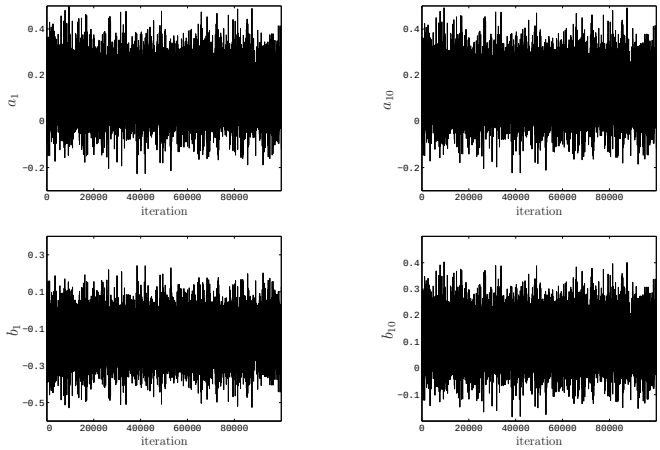

Figure C.4: A selection of chain paths from the Metropolis-Hastings algorithm. The parameters correspond to the nonlinear effect of covariate `age` *when a double penalty model is used (**union membership data**).*

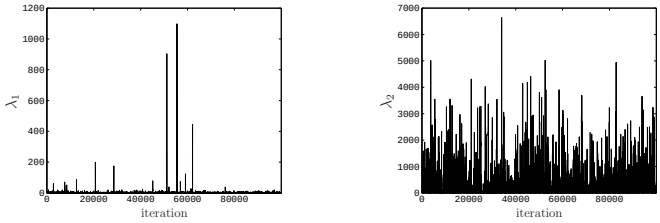

Figure C.5: Chain path from the Gibbs sampler for the smoothing parameters λ_1 and λ_2 associated with the nonlinear effects of `wage` *and* `age`*, respectively, for the single penalty model (**union membership data**).*

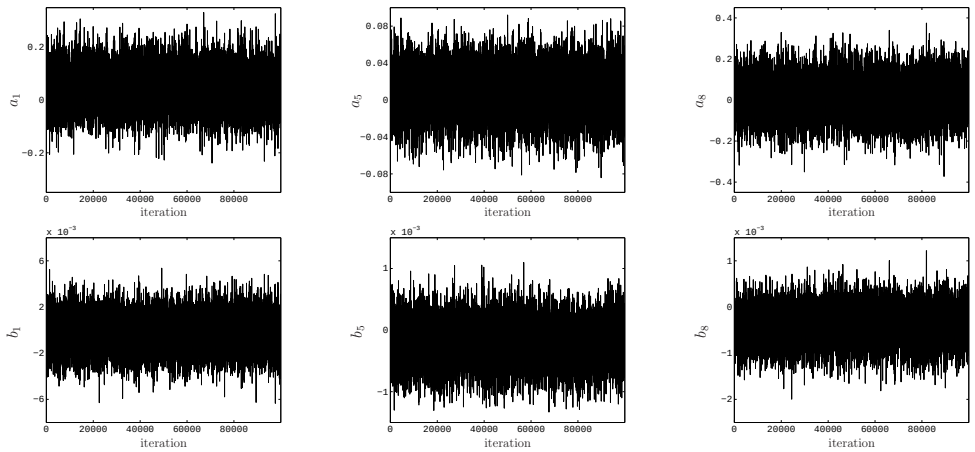

Figure C.6: A selection of chain paths from the Metropolis-Hastings algorithm. The parameters correspond to the time-varying coefficient of covariate **`age`** *when a single penalty model is used (**primary biliary cirrhosis data**).*

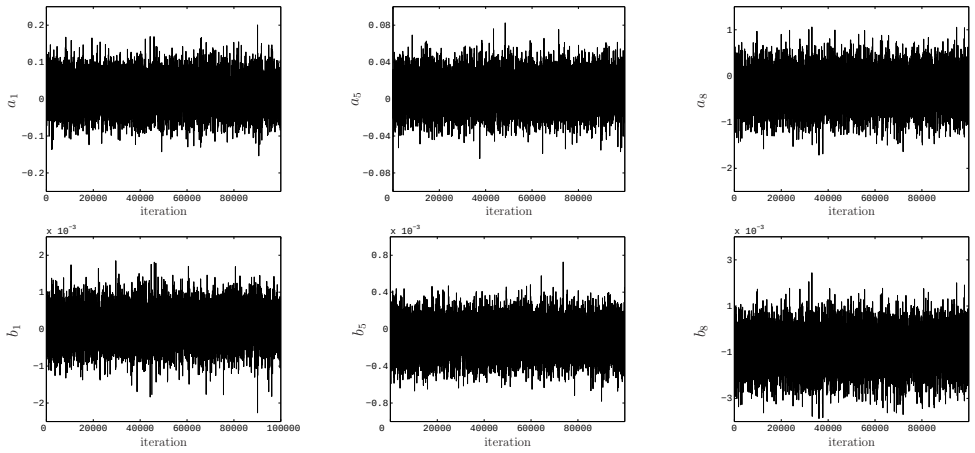

Figure C.7: A selection of chain paths from the Metropolis-Hastings algorithm. The parameters correspond to the time-varying coefficient of covariate **`age`** *when a double penalty model is used (**primary biliary cirrhosis data**).*

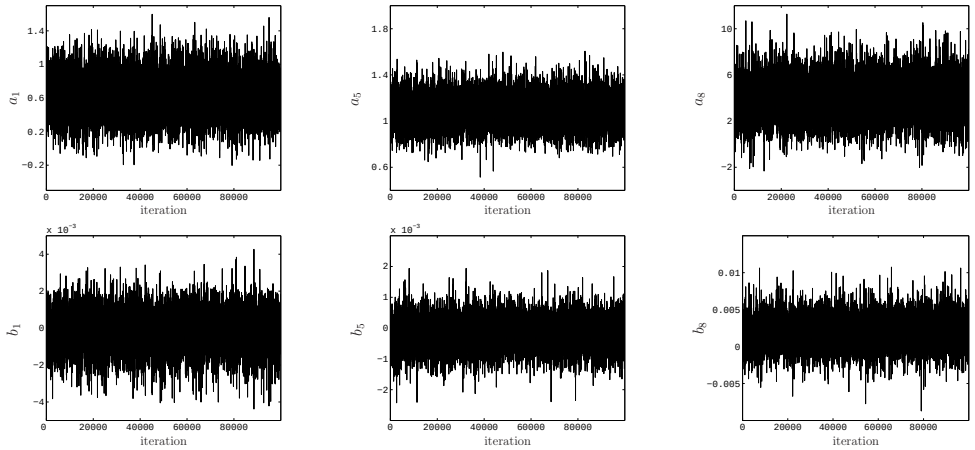

Figure C.8: A selection of chain paths from the Metropolis-Hastings algorithm. The parameters correspond to the time-varying coefficient of covariate **`bilirubin`** *when a single penalty model is used (**primary biliary cirrhosis data**).*

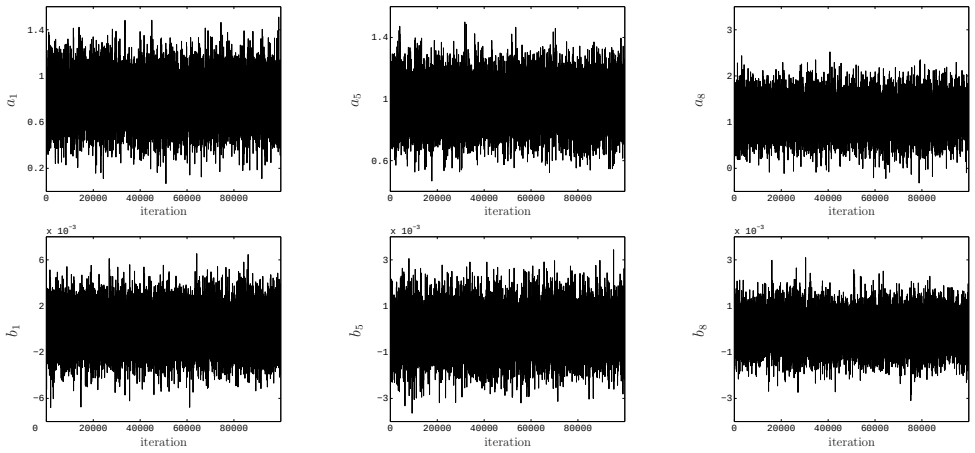

Figure C.9: A selection of chain paths from the Metropolis-Hastings algorithm. The parameters correspond to the time-varying coefficient of covariate **`bilirubin`** *when a double penalty model is used (**primary biliary cirrhosis data**).*

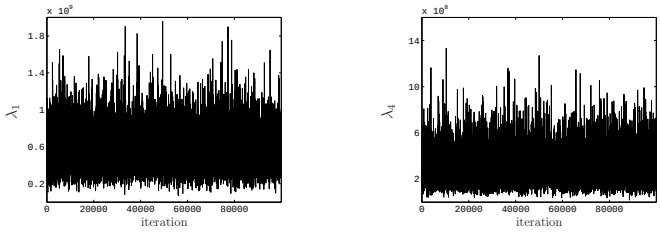

Figure C.10: Chain path from the Gibbs sampler for the smoothing parameters λ_1 and λ_4 associated with the time-varying effects of `age` *and* `bilirubin`*, respectively, for the single penalty model (**primary biliary cirrhosis data**).*

Appendix D

MATLAB Codes

This Appendix presents the MATLAB 7.0.1 (The MathWorks, 2008) codes and functions used to carry out all the analysis presented in Chapter 7. The codes for the analysis presented in Chapter 6 are essentially the same and so we omit them here.

Code to break ties and select failures.

```
load PBC5var;
t = data(:,1);
delta = data(:,2);
x1 = data(:,3);
x2 = data(:,4);
x3 = log(data(:,5));
x4 = log(data(:,6));
x5 = log(data(:,7));
clear ix2;
for ix2=1:length(x2)
    if (x2(ix2)==0.5)
```

```
        x2(ix2)=1;
    end
end
n = length(x1);
m1 = mean(x1);
m3 = mean(x3);
m4 = mean(x4);
m5 = mean(x5);
x1 = ( x1 - ( mean(x1)*ones(n,1) ) );
x3 = ( x3 - ( mean(x3)*ones(n,1) ) );
x4 = ( x4 - ( mean(x4)*ones(n,1) ) );
x5 = ( x5 - ( mean(x5)*ones(n,1) ) );
******Sort observations according to survival time******
data_ord = sortrows([t,delta,x1,x2,x3,x4,x5],1);
clear t delta x1 x2 x3 x4 x5;
t = data_ord(:,1);
delta = data_ord(:,2);
x1 = data_ord(:,3);
x2 = data_ord(:,4);
x3 = data_ord(:,5);
x4 = data_ord(:,6);
x5 = data_ord(:,7);
******Find tied observations******
ties = diff(t);
m = min(ties(find(ties>0)));
tnew = t;
for idiff=1:length(ties)
    if (ties(idiff)==0)
        while (tnew(idiff+1)<=tnew(idiff))
            a = unifrnd(-(0.5*m),0.5*m,1,1);
```

```
            tnew(idiff+1) = t(idiff+1) + a;
        end
    end
end
data_ord = sortrows([tnew,delta,x1,x2,x3,x4,x5],1);
tnew = data_ord(:,1);
deltanew = data_ord(:,2);
xnew = data_ord(:,3:size(data_all,2));
failures = [];
indexes = [];
clear id;
******Select observed failures******
for id=1:n
    if (deltanew(id)==1)
        indexes = [indexes,id];
        failures = [failures;tnew(id)];
    end
end
nfail = length(failures);
xfailures = xnew(indexes,:);
Xcov = [xnew'];
XcovFailures = [xfailures'];
```

Code to compute design matrix.

```
function X = design(x,knots);
d=diff(knots);
K=length(knots);
C=zeros(K-1,length(x));
for j=1:(K-1)
    C(j,:)=(knots(j)<=x) & (x<knots(j+1));
```

```
end;
C=C';
Tm=(0*x+1)*knots;
Ym=x*(0*knots+1);
Um=Ym-Tm;
X=[];
for j=1:length(x)
    X_local=[];
    for i=1:(K-1)
        X_local(i,(2*i-1):(2*i+2))=
            [((Um(j,i)-d(i))^(2)*((2*Um(j,i))+d(i)))/(d(i)^(3)),
            (Um(j,i)*((Um(j,i)-d(i))^(2)))/(d(i)^(2)),
            ((Um(j,i)^(2))*((3*d(i))-(2*Um(j,i))))/(d(i)^(3)),
            ((Um(j,i)^(2))*(Um(j,i)-d(i)))/(d(i)^(2))];
    end
    X(j,:)=C(j,:)*X_local;
    clear X_local;
end;
```

Code to compute penalty matrix based upon integrated squared second derivative.

```
function [P] = Penalty(knots,K);
d=diff(knots);
P=zeros(2*K,2*K);
P(1:4,1:4)=[12/(d(1)^(3)),6/(d(1)^(2)),-12/(d(1)^(3)),
    6/(d(1)^(2));6/(d(1)^(2)),
    4/d(1),-6/(d(1)^(2)),2/d(1);-12/(d(1)^(3)),
    -6/(d(1)^(2)),12/(d(1)^(3)),
    -6/(d(1)^(2));6/(d(1)^(2)),2/d(1),
    -6/(d(1)^(2)),4/d(1)];
for f=2:(K-1)
```

```
    P_local=[12/(d(f)^(3)),6/(d(f)^(2)),-12/(d(f)^(3))
        ,6/(d(f)^(2));6/(d(f)^(2)),
        4/d(f),-6/(d(f)^(2)),2/d(f);-12/(d(f)^(3)),
        -6/(d(f)^(2)),12/(d(f)^(3)),
        -6/(d(f)^(2));6/(d(f)^(2)),2/d(f),
        -6/(d(f)^(2)),4/d(f)];
    P((2*f-1):(2*f+2),(2*f-1):(2*f+2))=
        P((2*f-1):(2*f+2),(2*f-1):(2*f+2))+P_local;
    clear P_local;
end
```

Code to compute log-partial likelihood.

```
function partial = partiallikelihood(eta,indexes,xfailures,x)
nlik = size(eta,1);
nx = size(x,2);
partial = 0;
clear i;
for i=1:nlik
    partial = partial+((eta(i,:)*xfailures(:,i))-
        log(sum(exp((eta(i,:)*x(:,indexes(i):nx))))));
end
```

Code to compute hessian matrix.

```
function [Walpha,gradalpha] = hessianMDFL(eta,indexes,x,jind);
nhess = size(eta,1);
nx = size(x,2);
Walpha = zeros(nhess);
gradalpha = zeros(nhess,1);
for i=1:nhess
    predictor = exp((eta(i,:)*x(:,indexes(i):nx)));
```

```
    weight1 = sum(predictor);
    weight2 = sum((x(jind,indexes(i):nx).^2 ).*predictor);
    weight3 = (sum(x(jind,indexes(i):nx).*predictor));
    gradalpha(i) = x(jind,indexes(i))-
        (sum(x(jind,indexes(i):nx).*predictor)./weight1);
    Walpha(i,i) = ((weight2/weight1)-
        ((weight3/weight1)^2));
end
```

Code to compute matrix of cross partial derivatives.

```
function [Wuv] = weightuvMDFL(eta,indexes,x,uind,vind);
nhess = size(eta,1);
nx = size(x,2);
Wuv = zeros(nhess);
for i=1:nhess
    predictor = exp((eta(i,:)*x(:,indexes(i):nx)));
    weight1 = sum(predictor);
    weight2 = sum((x(uind,indexes(i):nx).*
        x(vind,indexes(i):nx)).*predictor);
    weight3 = (sum(x(uind,indexes(i):nx).*predictor))*
        (sum(x(vind,indexes(i):nx).*predictor));
    Wuv(i,i) = ((weight2/weight1)-(weight3/(weight1^2)));
end
```

Code to compute posterior mean estimates for the single penalty model.

```
while indx < nIter,
    count = count+1
    ******Update penalty matrices******
    Ppenalty1 = (P./tau1);
    Ppenalty2 = (P./tau2);
```

```
Ppenalty3 = (P./tau3);
Ppenalty4 = (P./tau4);
Ppenalty5 = (P./tau5);
*****************************
******Propose alpha_{1}******
*****************************
eta = X*alpha;
lik0 = partiallikelihood(eta,indexes,XcovFailures,Xcov);
******Replace alpha(:,1) by current posterior mode estimate******
etapmode = X*[pmode(:,1),alpha(:,2:p)];
******Compute weight matrices******
[W1,grad1] = hessianMDFL(etapmode,indexes,Xcov,1);
[W12] = weightuvMDFL(etapmode,indexes,Xcov,1,2);
[W13] = weightuvMDFL(etapmode,indexes,Xcov,1,3);
[W14] = weightuvMDFL(etapmode,indexes,Xcov,1,4);
[W15] = weightuvMDFL(etapmode,indexes,Xcov,1,5);
******Compute adjusted dependent variable******
z1 = (grad1+(W1*X*pmode(:,1))+(W12*X*alpha(:,2))+
    +(W13*X*alpha(:,3))+(W14*X*alpha(:,4))+(W15*X*alpha(:,5)));
******Compute proposal precision matrix******
Prec = ((X'*W1*X) + Ppenalty1);
R = chol(Prec);
vR = R\(normrnd(0,1,2*K,1));
******Compute proposal mean vector******
mnew = R\(R'\(X'*(z1-(W12*X*alpha(:,2))-(W13*X*alpha(:,3))-
    -(W14*X*alpha(:,4))-(W15*X*alpha(:,5)))));
******Compute proposed parameter vector******
alpha1new = vR + mnew;
etap = X*[alpha1new,alpha(:,2:p)];
liknew = partiallikelihood(etap,indexes,XcovFailures,Xcov);
```

```
priornew = -(alpha1new'*Ppenalty1*alpha1new)/2;
prior0 = -(alpha(:,1)'*Ppenalty1*alpha(:,1))/2;
proposalnew =  -((alpha1new-mnew)'*Prec*(alpha1new-mnew))/2;
proposalold = -((alpha(:,1)-mnew)'*Prec*(alpha(:,1)-mnew))/2;
******Compute acceptance probability******
ratio = liknew+priornew+proposalold-lik0-prior0-proposalnew;
u = log(unifrnd(0,1,1,1));
if (u <= ratio)
    alpha(:,1) = alpha1new;
    rate1 = rate1+1;
end;
******Update current posterior mode estimate******
pmode(:,1) = mnew;
****************************
******Propose alpha_{2}******
****************************
eta = X*alpha;
lik0 = partiallikelihood(eta,indexes,XcovFailures,Xcov);
******Replace alpha(:,2) by current posterior mode estimate******
etapmode = X*[alpha(:,1),pmode(:,2),alpha(:,3:p)];
******Compute weight matrices******
[W2,grad2] = hessianMDFL(etapmode,indexes,Xcov,2);
[W12] = weightuvMDFL(etapmode,indexes,Xcov,1,2);
[W23] = weightuvMDFL(etapmode,indexes,Xcov,2,3);
[W24] = weightuvMDFL(etapmode,indexes,Xcov,2,4);
[W25] = weightuvMDFL(etapmode,indexes,Xcov,2,5);
******Compute adjusted dependent variable******
z2 = (grad2+(W12*X*alpha(:,1))+(W2*X*pmode(:,2))+
    +(W23*X*alpha(:,3))+(W24*X*alpha(:,4))+(W25*X*alpha(:,5)));
******Compute proposal precision matrix******
```

```
Prec = ((X'*W2*X)+Ppenalty2);
R = chol(Prec);
vR = R\(normrnd(0,1,2*K,1));
******Compute proposal mean vector******
mnew = R\(R'\(X'*(z2-(W12*X*alpha(:,1))-(W23*X*alpha(:,3))-
    -(W24*X*alpha(:,4))-(W25*X*alpha(:,5)))));
******Compute proposed parameter vector******
alpha2new = vR+mnew;
etap = X*[alpha(:,1),alpha2new,alpha(:,3:p)];
liknew = partiallikelihood(etap,indexes,XcovFailures,Xcov);
priornew = -(alpha2new'*Ppenalty2*alpha2new)/2;
prior0 = -(alpha(:,2)'*Ppenalty2*alpha(:,2))/2;
proposalnew =  -((alpha2new-mnew)'*Prec*(alpha2new-mnew))/2;
proposalold = -((alpha(:,2)-mnew)'*Prec*(alpha(:,2)-mnew))/2;
******Compute acceptance probability******
ratio = liknew+priornew+proposalold-lik0-prior0-proposalnew;
u = log(unifrnd(0,1,1,1));
if (u <= ratio)
    alpha(:,2) = alpha2new;
    rate2 = rate2+1;
end;
******Update current posterior mode estimate******
pmode(:,2) = mnew;
*****************************
******Propose alpha_{3}******
*****************************
eta = X*alpha;
lik0 = partiallikelihood(eta,indexes,XcovFailures,Xcov);
******Replace alpha(:,3) by current posterior mode estimate******
etapmode = X*[alpha(:,1:2),pmode(:,3),alpha(:,4:5)];
```

```
******Compute weight matrices******
[W3,grad3] = hessianMDFL(etapmode,indexes,Xcov,3);
[W13] = weightuvMDFL(etapmode,indexes,Xcov,1,3);
[W23] = weightuvMDFL(etapmode,indexes,Xcov,2,3);
[W34] = weightuvMDFL(etapmode,indexes,Xcov,3,4);
[W35] = weightuvMDFL(etapmode,indexes,Xcov,3,5);
******Compute adjusted dependent variable******
z3 = (grad3+(W13*X*alpha(:,1))+(W23*X*alpha(:,2))+
    +(W3*X*pmode(:,3))+(W34*X*alpha(:,4))+(W35*X*alpha(:,5)));
******Compute proposal precision matrix******
Prec = ((X'*W3*X)+Ppenalty3);
R = chol(Prec);
vR = R\(normrnd(0,1,2*K,1));
******Compute proposal mean vector******
mnew = R\(R'\(X'*(z3-(W13*X*alpha(:,1))-(W23*X*alpha(:,2))-
    -(W34*X*alpha(:,4))-(W35*X*alpha(:,5)))));
******Compute proposed parameter vector******
alpha3new = vR+mnew;
etap = X*[alpha(:,1:2),alpha3new,alpha(:,4:5)];
liknew = partiallikelihood(etap,indexes,XcovFailures,Xcov);
priornew = -(alpha3new'*Ppenalty3*alpha3new)/2;
prior0 = -(alpha(:,3)'*Ppenalty3*alpha(:,3))/2;
proposalnew =  -((alpha3new-mnew)'*Prec*(alpha3new-mnew))/2;
proposalold = -((alpha(:,3)-mnew)'*Prec*(alpha(:,3)-mnew))/2;
******Compute acceptance probability******
ratio = liknew+priornew+proposalold-lik0-prior0-proposalnew;
u = log(unifrnd(0,1,1,1));
if (u <= ratio)
    alpha(:,3) = alpha3new;
    rate3 = rate3+1;
```

```
end;
******Update current posterior mode estimate******
pmode(:,3) = mnew;
*****************************
******Propose alpha_{4}******
*****************************
eta = X*alpha;
lik0 = partiallikelihood(eta,indexes,XcovFailures,Xcov);
******Replace alpha(:,4) by current posterior mode estimate******
etapmode = X*[alpha(:,1:3),pmode(:,4),alpha(:,5)];
******Compute weight matrices******
[W4,grad4] = hessianMDFL(etapmode,indexes,Xcov,4);
[W14] = weightuvMDFL(etapmode,indexes,Xcov,1,4);
[W24] = weightuvMDFL(etapmode,indexes,Xcov,2,4);
[W34] = weightuvMDFL(etapmode,indexes,Xcov,3,4);
[W45] = weightuvMDFL(etapmode,indexes,Xcov,4,5);
******Compute adjusted dependent variable******
z4 = (grad4+(W14*X*alpha(:,1))+(W24*X*alpha(:,2))+
    (W34*X*alpha(:,3))+(W4*X*pmode(:,4))+(W45*X*alpha(:,5)));
******Compute proposal precision matrix******
Prec = ((X'*W4*X)+Ppenalty4);
R = chol(Prec);
vR = R\(normrnd(0,1,2*K,1));
******Compute proposal mean vector******
mnew = R\(R'\(X'*(z4-(W14*X*alpha(:,1))-(W24*X*alpha(:,2))-
    -(W34*X*alpha(:,3))-(W45*X*alpha(:,5)))));
******Compute proposed parameter vector******
alpha4new = vR+mnew;
etap = X*[alpha(:,1:3),alpha4new,alpha(:,5)];
liknew = partiallikelihood(etap,indexes,XcovFailures,Xcov);
```

```
priornew = -(alpha4new'*Ppenalty4*alpha4new)/2;
prior0 = -(alpha(:,4)'*Ppenalty4*alpha(:,4))/2;
proposalnew =  -((alpha4new-mnew)'*Prec*(alpha4new-mnew))/2;
proposalold = -((alpha(:,4)-mnew)'*Prec*(alpha(:,4)-mnew))/2;
******Compute acceptance probability******
ratio = liknew+priornew+proposalold-lik0-prior0-proposalnew;
u = log(unifrnd(0,1,1,1));
if (u <= ratio)
    alpha(:,4) = alpha4new;
    rate4 = rate4 + 1;
end;
******Update current posterior mode estimate******
pmode(:,4) = mnew;
****************************
******Propose alpha_{5}******
****************************
eta = X*alpha;
lik0 = partiallikelihood(eta,indexes,XcovFailures,Xcov);
******Replace alpha(:,5) by current posterior mode estimate******
etapmode = X*[alpha(:,1:4),pmode(:,5)];
******Compute weight matrices******
[W5,grad5] = hessianMDFL(etapmode,indexes,Xcov,5);
[W15] = weightuvMDFL(etapmode,indexes,Xcov,1,5);
[W25] = weightuvMDFL(etapmode,indexes,Xcov,2,5);
[W35] = weightuvMDFL(etapmode,indexes,Xcov,3,5);
[W45] = weightuvMDFL(etapmode,indexes,Xcov,4,5);
******Compute adjusted dependent variable******
z5 = (grad5+(W15*X*alpha(:,1))+(W25*X*alpha(:,2))+
    (W35*X*alpha(:,3))+(W45*X*alpha(:,4))+(W5*X*pmode(:,5)));
******Compute proposal precision matrix******
```

```
Prec = ((X'*W5*X)+Ppenalty5);
R = chol(Prec);
vR = R\(normrnd(0,1,2*K,1));
******Compute proposal mean vector******
mnew = R\(R'\(X'*(z5-(W15*X*alpha(:,1))-(W25*X*alpha(:,2))-
    -(W35*X*alpha(:,3))-(W45*X*alpha(:,4)))));
******Compute proposed parameter vector******
alpha5new = vR+mnew;
etap = X * [alpha(:,1:4),alpha5new];
liknew = partiallikelihood(etap,indexes,XcovFailures,Xcov);
priornew = -(alpha5new'*Ppenalty5*alpha5new)/2;
prior0 = -(alpha(:,5)'*Ppenalty5*alpha(:,5))/2;
proposalnew =  -((alpha5new-mnew)'*Prec*(alpha5new-mnew))/2;
proposalold = -((alpha(:,5)-mnew)'*Prec*(alpha(:,5)-mnew))/2;
******Compute acceptance probability******
ratio = liknew+priornew+proposalold-lik0-prior0-proposalnew;
u = log(unifrnd(0,1,1,1));
if (u <= ratio)
    alpha(:,5) = alpha5new;
    rate5 = rate5+1
end;
******Update current posterior mode estimate******
pmode(:,5) = mnew;
******Update smoothing parameters******
temp1 = (alpha(:,1)'*P*alpha(:,1))/2;
newr1 = 1./(temp1+r);
tau1 = 1./(gamrnd(news,newr1));
temp2 = (alpha(:,2)'*P*alpha(:,2))/2;
newr2 = 1./(temp2+r);
tau2 = 1./(gamrnd(news,newr2));
```

```
    temp3 = (alpha(:,3)'*P*alpha(:,3))/2;
    newr3 = 1./(temp3+r);
    tau3 = 1./(gamrnd(news,newr3));
    temp4 = (alpha(:,4)'*P*alpha(:,4))/2;
    newr4 = 1./(temp4+r);
    tau4 = 1./(gamrnd(news,newr4));
    temp5 = (alpha(:,5)'*P*alpha(:,5))/2;
    newr5 = 1./(temp5+r);
    tau5 = 1./(gamrnd(news,newr5));
    ******Collect post burn-in samples******
    if count>burnin,
        if(rem(count,thining)==0),
            indx1=indx1+1;
            alpha1mat(indx1,:)=alpha(:,1)';
            alpha2mat(indx1,:)=alpha(:,2)';
            alpha3mat(indx1,:)=alpha(:,3)';
            alpha4mat(indx1,:)=alpha(:,4)';
            alpha5mat(indx1,:)=alpha(:,5)';
            tau1mat(indx1,:)=tau1;
            tau2mat(indx1,:)=tau2;
            tau3mat(indx1,:)=tau3;
            tau4mat(indx1,:)=tau4;
            tau5mat(indx1,:)=tau5;
            ****break ties****
            TreatDataAdditive;
            DesignPenalty;
        end;
    end;
    indx = indx+1;
end
```

```
******Computing P.E. and pointwise C.I.******
fhat1=mean(X*(alpha1mat'),2);
fhat2=mean(X*(alpha2mat'),2);
fhat3=mean(X*(alpha3mat'),2);
fhat4=mean(X*(alpha4mat'),2);
fhat5=mean(X*(alpha5mat'),2);
fhattemp1=X*(alpha1mat');
fhatsorted1=sort(fhattemp1,2);
fhattemp2=X*(alpha2mat');
fhatsorted2=sort(fhattemp2,2);
fhattemp3=X*(alpha3mat');
fhatsorted3=sort(fhattemp3,2);
fhattemp4=X*(alpha4mat');
fhatsorted4=sort(fhattemp4,2);
fhattemp5=X*(alpha5mat');
fhatsorted5=sort(fhattemp5,2);
a=2.5;
fhatlower1=(prctile(fhatsorted1',a))';
fhatupper1=(prctile(fhatsorted1',100-a))';
fhatlower2=(prctile(fhatsorted2',a))';
fhatupper2=(prctile(fhatsorted2',100-a))';
fhatlower3=(prctile(fhatsorted3',a))';
fhatupper3=(prctile(fhatsorted3',100-a))';
fhatlower4=(prctile(fhatsorted4',a))';
fhatupper4=(prctile(fhatsorted4',100-a))';
fhatlower5=(prctile(fhatsorted5',a))';
fhatupper5=(prctile(fhatsorted5',100-a))';
```

Code to perform iterative AIC minimization.

```
******Initialize estimates******
```

```
alpha = zeros(2*K,p);
tau22best = 10e-5;
tau12best = 10e-5;
tau23best = 10e-5;
tau13best = 10e-5;
tau24best = 10e-5;
tau14best = 10e-5;
tau25best = 10e-5;
tau15best = 10e-5;
TR2new = 1;
TR3new = 1;
TR4new = 1;
TR5new = 1;
iAll = 1;
CritAll = 1;
AICnew = 0;
while ((iAll < 10) && (CritAll > 10^(-8)))
******Estimate smoothing parameters for alpha(:,1)******
    AICold = AICnew;
    alpha1 = [];
    AIC = zeros(ngrid);
    TRtotal = zeros(ngrid);
    TR1 = zeros(ngrid);
    ******Vary tau1 and tau2 over pre-specified grid******
    for i2=1:ngrid,
        tau2test = grid2(i2);
        for i1=1:ngrid,
            tau1test = grid2(i1);
            Matrix = (P1./tau1test)+(P./tau2test);
            Crit1 = 1;
```

```
i = 1;
f_new = X*alpha;
alpha(:,1) = zeros(2*K,1);
f_new(:,1) = zeros(nfail,1);
******Estimate alpha(:,1) by local scoring******
while ( (i < 10) && (Crit1 > 10^(-8)) );
    f_old = f_new;
    eta = f_old;
    [W,grad] = hessianMDFL(eta,indexes,Xcov,1);
    [W12] = weightuvMDFL(eta,indexes,Xcov,1,2);
    [W13] = weightuvMDFL(eta,indexes,Xcov,1,3);
    [W14] = weightuvMDFL(eta,indexes,Xcov,1,4);
    [W15] = weightuvMDFL(eta,indexes,Xcov,1,5);
    R = chol((X'*W*X)+Matrix);
    S = R\(R'\(X'));
    z = (grad+(W*f_old(:,1))+(W12*f_old(:,2))
        +(W13*f_old(:,3))+(W14*f_old(:,4))
        +(W15*f_old(:,5)));
    alpha(:,1) = S*(z-(W12*X*alpha(:,2))-(W13*X*alpha(:,3))
        -(W14*X*alpha(:,4))-(W15*X*alpha(:,5)));
    f_new(:,1) = X*alpha(:,1);
    Crit1 = sqrt(sum(sum((f_new-f_old).^2,1)))
        /sqrt(sum(sum(f_old.^2,1)));
    i = i + 1;
end;
alpha1(i2,i1,:) = alpha(:,1);
******Compute degrees of freedom******
TR1(i2,i1) = sum(diag(X*(((X'*W*X)+Matrix)^(-1))*X'*W))-1;
TRtotal(i2,i1) = TR1(i2,i1)+TR2new+TR3new+TR4new+TR5new;
******Compute AIC******
```

```
            AIC(i2,i1) = -2*partial_likelihoodMD(eta,indexes,
                    XcovFailures,Xcov)+(2*TRtotal(i2,i1));
        end
    end
    Ind = find(AIC==(min(min(AIC))),1);
    ******Find estimate with minimum AIC******
    [r,c] = ind2sub(size(AIC),Ind);
    TR1new = TR1(r,c);
    alpha(:,1) = alpha1(r,c,:);
    tau2best = grid(r);
    tau1best = grid(c);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
******Estimate smoothing parameters for alpha(:,2)******
    alpha2 = [];
    AIC = zeros(ngrid);
    TRtotal = zeros(ngrid);
    TR2 = zeros(ngrid);
    ******Vary tau1 and tau2 over pre-specified grid******
    for i2=1:ngrid,
        tau2test = grid(i2);
        for i1=1:ngrid,
            tau1test = grid(i1);
            Matrix = (P1./tau1test)+(P./tau2test);
            Crit1 = 1;
            i = 1;
            f_new = X*alpha;
            alpha(:,2) = zeros(2*K,1);
            f_new(:,2) = zeros(nfail,1);
            ******Estimate alpha(:,2) by local scoring******
            while ( (i < 10) && (Crit1 > 10^(-8))
```

```
            f_old = f_new;
            eta = f_old;
            [W,grad] = hessianMDFL(eta,indexes,Xcov,2);
            [W21] = weightuvMDFL(eta,indexes,Xcov,2,1);
            [W23] = weightuvMDFL(eta,indexes,Xcov,2,3);
            [W24] = weightuvMDFL(eta,indexes,Xcov,2,4);
            [W25] = weightuvMDFL(eta,indexes,Xcov,2,5);
            R = chol((X'*W*X)+Matrix);
            S = R\(R'\(X'));
            z = (grad+(W21*f_old(:,1))+(W*f_old(:,2))
                +(W23*f_old(:,3))+(W24*f_old(:,4))
                +(W25*f_old(:,5)));
            alpha(:,2) = S*(z-(W21*f_old(:,1))-(W23*f_old(:,3))
                -(W24*f_old(:,4))-(W25*f_old(:,5)));
            f_new(:,2) = X*alpha(:,2);
            Crit1 = sqrt(sum(sum((f_new-f_old).^2,1)))
                /sqrt(sum(sum(f_old.^2,1)));
            i = i + 1;
        end;
        alpha2(i2,i1,:) = alpha(:,2);
        ******Compute degrees of freedom******
        TR2(i2,i1) = sum( diag(X*(((X'*W*X)+Matrix)^(-1))*X'*W))-1;
        TRtotal(i2,i1) = TR2(i2,i1)+TR1new+TR3new+TR4new+TR5new;
        ******Compute AIC******
        AIC(i2,i1) = -2*partial_likelihoodMD(eta,indexes,
                XcovFailures,Xcov) + (2*TRtotal(i2,i1));
    end
end
Ind = find(AIC==(min(min(AIC))),1);
******Find estimate with minimum AIC******
```

```
    [r,c] = ind2sub(size(AIC),Ind);
    TR2new = TR2(r,c);
    alpha(:,2) = alpha2(r,c,:);
    tau22best = grid(r);
    tau12best = grid(c);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
******Estimate smoothing parameters for alpha(:,3)******
    alpha3 = [];
    AIC = zeros(ngrid);
    TRtotal = zeros(ngrid);
    TR3 = zeros(ngrid);
    ******Vary tau1 and tau2 over pre-specified grid******
    for i2=1:ngrid,
        tau2test = grid(i2);
        for i1=1:ngrid,
            tau1test = grid(i1);
            Matrix = (P1./tau1test)+(P./tau2test);
            Crit1 = 1;
            i = 1;
            f_new = X*alpha;
            alpha(:,3) = zeros(2*K,1);
            f_new(:,3) = zeros(nfail,1);
            ******Estimate alpha(:,3) by local scoring******
            while ( (i < 10) && (Crit1 > 10^(-8)) );
                f_old = f_new;
                eta = X * alpha;
                [W,grad] = hessianMDFL(eta,indexes,Xcov,3);
                [W31] = weightuvMDFL(eta,indexes,Xcov,3,1);
                [W32] = weightuvMDFL(eta,indexes,Xcov,3,2);
                [W34] = weightuvMDFL(eta,indexes,Xcov,3,4);
```

```
                [W35] = weightuvMDFL(eta,indexes,Xcov,3,5);
                R = chol((X'*W*X)+Matrix);
                S = R\(R'\(X'));
                z = (grad+(W31*f_old(:,1))+(W32*f_old(:,2))
                    +(W*f_old(:,3))+(W34*f_old(:,4))
                    +(W35*f_old(:,5)));
                alpha(:,3) = S*(z-(W31*f_old(:,1))-(W32*f_old(:,2))
                    -(W34*f_old(:,4))-(W35*f_old(:,5)));
                f_new(:,3) = X*alpha(:,3);
                Crit1 = sqrt(sum(sum((f_new-f_old).^2,1)))
                    /sqrt(sum(sum(f_old.^2,1)));
                i = i + 1;
            end;
            alpha3(i2,i1,:) = alpha(:,3);
            ******Compute degrees of freedom******
            TR3(i2,i1) = sum(diag(X*(((X'*W*X)+Matrix)^(-1))*X'*W))-1;
            TRtotal(i2,i1) = TR3(i2,i1)+TR1new+TR2new+TR4new+TR5new;
            ******Compute AIC******
            AIC(i2,i1) = -2*partial_likelihoodMD(eta,indexes,
                    XcovFailures,Xcov)+(2*TRtotal(i2,i1));
        end
    end
    Ind = find(AIC==(min(min(AIC))),1);
    ******Find estimate with minimum AIC******
    [r,c] = ind2sub(size(AIC),Ind);
    TR3new = TR3(r,c);
    alpha(:,3) = alpha3(r,c,:);
    tau23best = grid(r);
    tau13best = grid(c);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
******Estimate smoothing parameters for alpha(:,4)******
    alpha4 = [];
    AIC = zeros(ngrid);
    TRtotal = zeros(ngrid);
    TR4 = zeros(ngrid);
    ******Vary tau1 and tau2 over pre-specified grid******
    for i2=1:ngrid,
        tau2test = grid(i2);
        for i1=1:ngrid,
            tau1test = grid(i1);
            Matrix = (P1./tau1test)+(P./tau2test);
            Crit1 = 1;
            i = 1;
            f_new = X * alpha;
            alpha(:,4) = zeros(2*K,1);
            f_new(:,4) = zeros(nfail,1);
            ******Estimate alpha(:,4) by local scoring******
            while ( (i < 10) && (Crit1 > 10^(-8)) );
                f_old = f_new;
                eta = f_old;
                [W,grad] = hessianMDFL(eta,indexes,Xcov,4);
                [W41] = weightuvMDFL(eta,indexes,Xcov,4,1);
                [W42] = weightuvMDFL(eta,indexes,Xcov,4,2);
                [W43] = weightuvMDFL(eta,indexes,Xcov,4,3);
                [W45] = weightuvMDFL(eta,indexes,Xcov,2,5);
                R = chol((X'*W*X)+Matrix);
                S = R\(R'\(X'));
                z = (grad+(W41*f_old(:,1))+(W42*f_old(:,2))
                    +(W43*f_old(:,3))+(W*f_old(:,4))
                    +(W45*f_old(:,5)));
```

```
                alpha(:,4) = S*(z-(W41*f_old(:,1))-(W42*f_old(:,2))
                    -(W43*f_old(:,3))-(W45*f_old(:,5)));
                f_new(:,4) = X*alpha(:,4);
                Crit1 = sqrt(sum(sum((f_new-f_old).^2,1)))
                    /sqrt(sum(sum(f_old.^2,1)));
                i = i + 1;
            end;
            alpha4(i2,i1,:) = alpha(:,4);
            ******Compute degrees of freedom******
            TR4(i2,i1) = sum(diag(X*(((X'*W*X)+Matrix)^(-1))*X'*W) )-1;
            TRtotal(i2,i1) = TR4(i2,i1)+TR1new+TR2new+TR3new+TR5new;
            ******Compute AIC******
            AIC(i2,i1) = -2*partial_likelihoodMD(eta,indexes,
                    XcovFailures,Xcov)+(2*TRtotal(i2,i1));
        end
    end
    Ind = find(AIC==(min(min(AIC))),1);
    ******Find estimate with minimum AIC******
    [r,c] = ind2sub(size(AIC),Ind);
    TR4new = TR4(r,c);
    alpha(:,4) = alpha4(r,c,:);
    tau24best = grid(r);
    tau14best = grid(c);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
******Estimate smoothing parameters for alpha(:,5)******
    alpha5 = [];
    AIC = zeros(ngrid);
    TRtotal = zeros(ngrid);
    TR5 = zeros(ngrid);
    ******Vary tau1 and tau2 over pre-specified grid******
```

```
for i2=1:ngrid,
    tau2test = grid(i2);
    for i1=1:ngrid,
        tau1test = grid(i1);
        Matrix = (P1./tau1test)+(P./tau2test);
        Crit1 = 1;
        i = 1;
        f_new = X*alpha;
        alpha(:,5) = zeros(2*K,1);
        f_new(:,5) = zeros(nfail,1);
        ******Estimate alpha(:,5) by local scoring******
        while ( (i < 10) && (Crit1 > 10^(-8)) );
            f_old = f_new;
            eta = X * alpha;
            [W,grad] = hessianMDFL(eta,indexes,Xcov,5);
            [W51] = weightuvMDFL(eta,indexes,Xcov,5,1);
            [W52] = weightuvMDFL(eta,indexes,Xcov,5,2);
            [W53] = weightuvMDFL(eta,indexes,Xcov,5,3);
            [W54] = weightuvMDFL(eta,indexes,Xcov,5,4);
            R = chol((X'*W*X)+Matrix);
            S = R\(R'\(X'));
            z = (grad+(W51*f_old(:,1))+(W52*f_old(:,2))
                +(W53*f_old(:,3))+(W54*f_old(:,4))
                +(W*f_old(:,5)));
            alpha(:,5) = S*(z-(W51*f_old(:,1))-(W52*f_old(:,2))
                -(W53*f_old(:,3))-(W54*f_old(:,4)));
            f_new(:,5) = X*alpha(:,5);
            Crit1 = sqrt(sum(sum((f_new-f_old).^2,1)))
                /sqrt(sum(sum(f_old.^2,1)));
            i = i + 1;
```

```
            end;
            alpha5(i2,i1,:) = alpha(:,5);
            ******Compute degrees of freedom******
            TR5(i2,i1) = sum(diag(X*(((X'*W*X)+Matrix)^(-1))*X'*W) )-1;
            TRtotal(i2,i1) = TR5(i2,i1)+TR1new+TR2new+TR3new+TR4new;
            ******Compute AIC******
            AIC(i2,i1) = -2*partial_likelihoodMD(eta,indexes,
                    XcovFailures,Xcov)+(2*TRtotal(i2,i1));
        end
    end
    Ind = find(AIC==(min(min(AIC))),1);
    ******Find estimate with minimum AIC******
    [r,c] = ind2sub(size(AIC),Ind);
    TR5new = TR5(r,c);
    alpha(:,5) = alpha5(r,c,:);
    tau25best = grid(r);
    tau15best = grid(c);
    AICnew = AIC(r,c);
    ******Compute convergence criterion******
    CritAll = sqrt(sum((AICnew-AICold)^2))/sqrt(sum(AICold^2))
    iAll = iAll + 1
end;
```

Bibliography

Abrahamowicz, M., MacKenzie, T. & Esdaile, J. M. (1996). Time-dependent hazard ratio: modelling and hypothesis testing with application in lupus nephritis. *Journal of the American Statistical Association* 91 1432–1439.

Aldrin, M. (2006). Improved predictions penalizing both slope and curvature in additive models. *Computational Statistics and Data Analysis* 50 267–284.

Andersen, P. K. & Gill, R. D. (1982). Cox's regression model for counting processes: A large sample study. *The Annals of Statistics* 10 1100–1120.

Baladandayuthapani, V., Mallick, B. K. & Carroll, R. J. (2005). Spatially adaptive Bayesian penalized regression splines (P-splines). *Journal of Computational and Graphical Statistics* 14 378–394.

Berndt, E. R. (1991). *The Practice of Econometrics: Classical and Comtemporary.* Addison-Wesley.

Breslow, N. E. (1972). Discussion of the paper by D. R. Cox. *Journal of the Royal Statistical Society, Series B* 34 216–217.

Brezger, A. & Lang, S. (2006). Generalized structured additive regression based on Bayesian P-splines. *Computational Statistics and Data Analysis* 50 957–991.

Burden, R. L. & Faires, J. D. (2004). *Numerical Analysis*. Brooks-Cole Publishing, eighth ed.

Carlin, B. P. & Louis, T. A. (2000). *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall.

Casella, G. & George, E. I. (1990). Explaining the Gibbs sampler. *The American Statistician* 46 167–174.

Chib, S. & Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician* 49 327–335.

Collett, D. (2003). *Modelling Survival Data in Medical Research*. Chapman & Hall.

Costa, M. J. & Shaw, J. E. H. (2008). Parametrization and penalties in spline models with an application to survival analysis. To appear in *Computational Statistics and Data Analysis* (http://dx.doi.org/10.1016/j.csda.2008.07.026).

Cox, D. D. (1996). Discussion of the paper by P. Eilers and Brian Marx. *Statistical Science* 11 112–114.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society, Series B* 34 187–220.

Cox, D. R. (1975). Partial likelihood. *Biometrika* 62 269–276.

Crainiceanu, C. M., Ruppert, D., Carroll, R. J., Joshi, A. & Goodner, B. (2007). Spatially adaptive Bayesian penalized splines with heteroscedastic errors. *Journal of Computational and Graphical Statistics* 16 265–288.

Dannegger, F., Klinger, A., & Ulm, K. (1995). Identification of prognostic factors with censored data. Discussion Paper 11, Ludwig-Maximilians Universität München.

De Boor, C. (1978). *A Practical Guide to Splines*. Springer.

Denison, D. G. T., Mallick, B. K. & Smith, A. F. M. (1998a). Automatic Bayesian curve fitting. *Journal of the Royal Statistical Society, Series B* 60 333–350.

Denison, D. G. T., Mallick, B. K. & Smith, A. F. M. (1998b). Bayesian MARS. *Statistics and Computing* 8 337–346.

Dikshit, H. P. & Powar, P. (1981). On deficient cubic spline interpolation. *Journal of Approximation Theory* 31 99–106.

Dubeau, F. & Savoie, J. (1999). On best error bounds for deficient splines. In *CRM Proceedings and Lecture Notes*, vol. 18. Centre de Recherches Mathématiques, 33–39.

Eilers, P. H. C. & Goeman, J. J. (2004). Enhancing scatterplots with smoothed densities. *Bioinformatics* 20 623–628.

Eilers, P. H. C. & Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science* 11 89–121.

Eilers, P. H. C. & Marx, B. D. (2003). Multivariate calibration with temperature interaction using two-dimensional penalized signal regression. *Chemometrics and Intelligent Laboratory Systems* 66 159–174.

Eubank, R. L. (1999). *Nonparametric Regression and Spline Smoothing*. Marcel Dekker, Inc.

Fahrmeir, L., Kneib, T. & Lang, S. (2004). Penalized structured additive regression for space-time data: a Bayesian perspective. *Statistica Sinica* 14 731–761.

Fleming, T. R. & Harrington, D. P. (1991). *Counting Processes and Survival Analysis*. Wiley-Interscience, New York.

Friedman, J. & Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American Statistical Association* 76 817–823.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics* 19 1–67.

Gamerman, D. (1997). Sampling from the posterior distribution in generalized linear mixed models. *Statistics and Computing* 7 57–68.

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 721–741.

Grambsch, P. M. & Therneau, T. M. (1994). Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika* 81 515–526.

Gray, R. J. (1992). Flexible methods for analyzing survival data using splines, with applications to breast cancer prognosis. *Journal of the American Statistical Association* 87 942–951.

Gray, R. J. (1994). Spline based tests in survival analysis. *Biometrics* 50 640–652.

Green, P. J. (1987). Penalized likelihood for general semi-parametric regression models. *International Statistical Review* 55 245–259.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82 711–732.

Green, P. J. & Silverman, B. W. (1994). *Nonparametric Regression and Generalized Linear Models*. Chapman & Hall.

Hastie, T. & Tibshirani, R. (1986). Generalized additive models (with discussion). *Statistical Science* 1 297–318.

Hastie, T. & Tibshirani, R. (1990a). Exploring the nature of covariate effects in the proportional hazards model. *Biometrics* 46 1005–1016.

Hastie, T. & Tibshirani, R. (1990b). *Generalized Additive Models*. Chapman & Hall.

Hastie, T. & Tibshirani, R. (1993). Varying-coefficients models. *Journal of the Royal Statistical Society, Series B* 55 757–796.

Hastings, W. K. (1970). Monte carlo sampling methods using Markov chains and their applications. *Biometrika* 57 97–109.

Hennerfeind, A., Brezger, A. & Fahrmeir, L. (2006). Geoadditive survival models. *Journal of the American Statistical Association* 101 1065–1075.

Kalbfleisch, J. D. (1978). Nonparametric Bayesian analysis of survival time data. *Journal of the Royal Statistical Society, Series B* 40 214–221.

Kauermann, G. (2005). Penalized spline smoothing in multivariable survival models with varying coefficients. *Computational Statistics and Data Analysis* 49 169–186.

Kneib, T. & Fahrmeir, L. (2007). A mixed model approach for geoadditive hazard regression. *Scandinavian Journal of Statistics* 34 207–228.

Lambert, P. & Eilers, P. H. C. (2005). Bayesian proportional hazards model with time-varying regression coefficients: A penalized Poisson regression approach. *Statistics in Medicine* 24 3977–3989.

Lancaster, P. & Šalkauskas, K. (1986). *Curve and Surface Fitting: An Introduction*. London: Academic Press.

Lang, S. & Brezger, A. (2004). Bayesian P-splines. *Journal of Computational and Graphical Statistics* 13 183–212.

Loader, C. (2004). Smoothing: Local regression techniques. To appear in Handbook of Computational Statistics. Editors: James Gentle, Wolfgang Härdle, Yoichi Mori. Springer-Verlag.

Martinussen, T. & Scheike, T. H. (2006). *Dynamic Regression Models for Survival Data*. Springer.

Marx, B. D. & Eilers, P. H. C. (1998). Direct generalized additive modeling with penalized likelihood. *Computational Statistics and Data Analysis* 28 193–209.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953). Equation of state calculations by fast computing machine. *Journal of Chemical Physics* 21 1087–1091.

Nelder, J. A. & Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A* 135 370–384.

O'Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems (with discussion). *Statistical Science* 1 505–527.

Pintore, A., Speckman, P. L. & Holmes, C. C. (2006). Spatially adaptive smoothing splines. *Biometrika* 93 113–125.

Qiou, Z., Ravishanker, N. & Dey, D. K. (1999). Multivariate survival analysis with positive stable frailties. *Biometrics* 55 637–644.

Rana, S. S. & Purohit, M. (1988). Deficient cubic spline interpolation. In *Proceedings of the Japan Academy*, vol. 64. Serie A.

Robert, C. P. & Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer.

Ruppert, D. & Carroll, R. J. (1997). Penalized regression splines. `"http://www.orie.cornell.edu/$\sim$davidr/papers`.

Ruppert, D. & Carroll, R. J. (2000). Spatially-adaptive penalties for spline fitting. *Australian and New Zealand Journal of Statistics* 42 205–223.

Ruppert, D., Wand, M. P. & Carroll, R. J. (2003). *Semiparametric Regression*. Cambridge University Press.

Sargent, D. J. (1997). A flexible approach to time-varying coefficients in the Cox regression setting. *Lifetime Data Analysis* 3 13–25.

Schoenberg, I. (1964a). On interpolation by spline functions and its minimum properties. *International Series of Numerical Analysis* 5 109–129.

Schoenberg, I. (1964b). Spline functions and the problem of graduation. vol. 52. Natural Academy of Science U.S.A., 947–950.

Schumaker, L. L. (1981). *Spline Functions: Basic Theory*. John Wiley & Sons, Inc.

Shaw, J. E. H. (1987). Numerical Bayesian analysis of some flexible regression models. *The Statistician* 36 147–153.

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

Sinha, D., Ibrahim, J. G. & Chen, M.-Hui. (2003). A Bayesian justification of Cox's partial likelihood. *Biometrika* 90 629–641.

Tian, L., Zucker, D. & Wei, L. J. (2005). On the Cox model with time-varying regression coefficients. *Journal of the American Statistical Association* 100 172–183.

Wahba, G. (1978). Improper priors, spline smoothing and the problem of guarding against model errors in regression. *Journal of the Royal Statistical Society, Series B* 40 364–372.

Wahba, G. (1980). Spline bases, regularization, and generalized cross validation for solving approximation problems with large quantities of noisy data. In W. Cheney, ed., *Approximation Theory III*. London: Academic Press, 905–912.

Wahba, G. (1983). Bayesian confidence intervals for the cross-validated smoothing spline. *Journal of the Royal Statistical Society, Series B* 45 133–150.

Wahba, G. (1990). *Spline Models for Observational Data*. CBMS-NSF 59, Regional Conference Series in Applied Mathematics.

Wakefield, J. C., Gelfand, A. E. & Smith, A. F. M. (1991). Efficient generation of random variates via the ratio-of-uniforms method. *Statistics and Computing* 1 129–133.

Whittaker, E. T. (1923). On a new method of graduation. *Proceedings of the Edinburgh Mathematical Society* 41 63–75.

Wood, S. N. (2000). Modelling and smoothing parameter estimation with multiple quadratic penalties. *Journal of the Royal Statistical Society, Series B* 62 413–428.

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society, Series B* 65 95–114.

Wood, S. N. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association* 99 673–686.

Wood, S. N. (2006a). *Generalized Additive Models - An Introduction with R*. Chapman & Hall.

Wood, S. N. (2006b). Low-rank scale-invariant tensor product smooths for generalized additive mixed models. *Biometrics* 62 1025–1036.

Zucker, D. M. & Karr, A. F. (1990). Nonparametric survival analysis with time-dependent covariate effects: A penalized partial likelihood approach. *The Annals of Statistics* 18 329–353.