University of Auckland Research Repository, ResearchSpace

Copyright Statement

General copyright and disclaimer

P-SPLINE VECTOR GENERALIZED ADDITIVE MODELS

Chanatda Somchit

A thesis submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy in Statistics,

The University of Auckland, 2016

Abstract

Vector generalized additive models (VGAMs) are an extension of the class of generalized additive models (GAMs) to include multivariate regression models in a very natural way by using vector smoothing. The current VGAM class is very large and includes many statistical distributions and models, for example, univariate and multivariate distributions, categorical data analysis, quantile and expectile regression, time series, survival analysis, extreme value analysis, and nonlinear least-squares models. Parameter estimation is performed by a combination of IRLS and modified vector backfitting using vector splines. A major issue, however, is that it is not easy to efficiently integrate smoothness estimation methods with the backfitting approach.

The aim of this research study is to introduce a new efficient method based on penalized regression splines for estimating parameter coefficients to the VGAM class, and to integrate automatic numerical procedures to determine the shape of non-linear terms from the data into the VGAM framework. To achieve these, we develop VGAMs based on penalized regression splines using P-spline smoothers, which we term 'P-spline VGAMs'. P-spline VGAMs are represented in this thesis as penalized vector generalized linear models (VGLMs), where each smooth component of a P-spline VGAM is represented using penalized B-splines or P-spline smoothers and has an associated discrete penalty measuring its wiggliness controlled by the smoothing parameter. P-spline VGAMs can be then fitted by the usual iteratively reweighted least squares (IRLS) scheme for VGLMs, except that a penalized least squares problem, in which the set of smoothing parameters must be estimated alongside the other model parameters, is solved at each iterate. The smoothing parameters are estimated by minimizing the approximate unbiased risk estimator (UBRE) using the computational procedure for the automatic and stable multiple smoothing parameter selection based on the pivoted QR decomposition and singular value decomposition. Importantly, the new fitting procedure is developed for the full range of VGAM models involving infrastructure such as constraints on model terms.

This research study describes the theoretical and practical aspects of the proposed method (P-spline vector generalized additive models). The methods have been implemented as R functions and the practical performance of the proposed method is investigated and compared to the existing approaches (VGAMs based on the classical backfitting) via simulation. As an illustration of the developments, the proposed method is applied to data from a cross-sectional workforce study combined with a health survey from New Zealand during the 1990s, and data from a survey study of the pregnancy and birth process during $1990 - 2004$, using several statistical models, which include the multinomial logit, proportional and non-proportional odds models, bivariate logistic model, and the LMS method for quantile regression.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisors, Professors Chris Wild and Dr.Thomas Yee for all their guidance and insight, for their patience and encouragement, and always providing reliable support. Their guidance helped me during the time of my research and writing of this thesis.

My sincere thanks also goes to Associate Prof. Ross Ihaka. His help, support and suggestions greatly improved this work. Thanks also to the IT team for all the IT assistance they have provided and all the staff from the Department of Statistics for their help throughout my Ph.D. study.

My thanks to the University of Phayao, Thailand, for my Doctoral Scholarship which provided the financial support that enabled me to complete this research.

I would like to acknowledge my colleagues Jennifer Wilcock and Maryann Pirie from the Department of Statistics who were a pleasure to be around. Lastly, I would also like to thank my family, especially my mother and my brother in Thailand, for all their encouragement and faithful support during this Ph.D. Their blessing and their belief resulted in me being focused and completing my Ph.D. successfully.

Table of Contents

Page

List of Tables

Table **Page**

List of Figures

Figure **Page**

List of Abbreviations

AIC	Akaike's Information Criterion
BIC	Bayesian Information Criteria
EDF	Effective Degrees of Freedom
GAMs	Generalized Additive Models
GCV	Generalized Cross Validation
GLMM	Generalized Linear Mixed Model
GLMs	Generalized Linear Models
GML	Generalized Maximum Likelihood
GSS	Generalized Spline Smoothing
IC	Information Criterion
IRLS	Iteratively Reweighted Least Squares
LMM	Linear Mixed Model
ML	Maximum (Marginal) Likelihood

MSE	Mean Squared Error
P-IRLS	Penalized Iteratively Reweighted Least Squares
PMSE	Percentage Mean Squared Error
REML	Restricted Maximum Likelihood
RMSE	Root Mean Squared Error
SS-ANOVA	Smoothing-Spline Analysis-of-Variance
UBRE	Unbiased Risk Estimator
VGAMs	Vector Generalized Additive Models
VGLMs	Vector Generalized Linear Models
w.r.t.	with respect to

CHAPTER 1

Introduction

Generalized additive models (GAMs) (Hastie and Tibshirani, 1986, 1990) are a nonparametric extension of generalized linear models (GLMs) (Nelder and Wedderburn, 1972). GAMs are now widely used and have become a standard statistical technique that allows considerable flexibility. Consider a univariate response y that belongs to the exponential family, where $\mu = \mathrm{E}(y)$ is related to p covariates $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^T$. GLMs model μ parameter in terms of a linear combination of the explanatory variables connected by a link function g

$$g(\mu) = \eta(\boldsymbol{x}) = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p, \tag{1.1}$$

where $x_1 \equiv 1$ if the model contains an intercept and $\beta_1, \beta_2, \cdots, \beta_p$ are a set of unknown parameters. GAMs generalize (1.1) to:

$$g(\mu) = \eta(\boldsymbol{x}) = \beta_1 x_1 + f_2(x_2) + \cdots + f_p(x_p). \tag{1.2}$$

The key feature of (1.2) is that the mean of the response depends on the covariates through a sum of smooth terms, where the f_k's are arbitrary smooth functions. The use of smooth functions adds much flexibility for the modeling of nonlinear relationships between the predictor variables and the dependent variable. The exact parametric form of these smooth functions is unknown, as is the degree of smoothness appropriate for each of them. GAMs as introduced by Hastie and Tibshirani (1990) were based on the classical backfitting approach. In backfitting, the algorithm iterates through the individual smooth components in the model and updates each component using an appropriate smoother. It does this by iteratively smoothing partial residuals from the model with respect to the covariate that the smooth relates to. The cycles continue until the smooth fits stabilize. GAMs within the backfitting framework can be fitted by minimizing a penalized weighted least-squares problem, where the backfitting algorithm is used in conjunction with the *local-scoring algorithm*. Representation of GAMs using linear smoothers and estimation performed by backfitting have the advantage that a wide range of smoothers can be used, but choosing the amount of smoothing to perform model selection is still difficult. The backfitting framework is poorly suited to dealing with the issue of automatically estimating the amount of smoothing for the smooth terms.

The issue of estimating the amount of smoothing for smooth terms of GAMs has been recognized and discussed in Hastie and Tibshirani (1990). Several authors suggested using penalized regression splines for GAM modeling (Hastie and Tibshirani, 1990; Hastie, 1996; Eilers and Marx, 1996; Wand, 2000; Wood, 2000, 2004, 2006b). Eilers and Marx (1996), Marx and Eilers (1998) and Wood (2000) proposed suitable parametric representations for the smooth functions based on the penalized regression spline approach. They represented GAMs as penalized generalized linear models (GLMs), where each smooth term of (1.2) is represented using an appropriate set of basis functions and has an associated penalty measuring the wiggliness, where the smoothing parameters are given to each penalty in the penalized likelihood to control the wiggliness. Models were fitted by penalized iteratively reweighted least squares (P-IRLS). The generalization of GAMs using penalized regression splines allowed the models to be estimated by penalized regression methods. Furthermore, this allowed smoothing parameter selection to be integrated as part of the P-IRLS scheme in a computationally efficient manner using well-founded criteria such as a generalized cross validation (GCV) or unbiased risk estimator (UBRE which can be used as an approximation of AIC for many GAMs) (see Wood, 2006b). Wood (2000, 2004, 2006b) proposed a computational method to control and choose the degrees of smoothness appropriately. His penalized likelihood-based approach did, by contrast, make possible an efficient computational method for automatic multiple smoothing parameter selection used to determine the functional form of any relationship from the data. This approach presented a substantial advantage over the original GAM class.

More recently, Marra and Radice (2011) extended the penalized likelihood approach with regression splines to a bivariate binary response modeling problem. They proposed a fitting procedure based on penalized regression spline approach for a nonstandard semipametric bivariate probit analysis (whose model exhibits the recursive structure). They represented smooth terms in the models using penalized regression splines and fitted by maximization of the penalized likelihood. Once again, representation of such model using penalized regression splines allowed the models to be estimated by penalized regression methods and the appropriate degree of smoothness for the smooth components can be estimated from the data using the UBRE score.

There have been many extensions proposed to the GLM family. One of them called vector generalized linear models (VGLMs) and vector generalized additive models (VGAMs) was developed by Yee and Wild (1996). VGLMs/VGAMs extended the class of GLMs/GAMs to include classes of multivariate regression models. This generalization covers a broad range of models such as multiple logistic regression model for nominal responses, continuation ratio models and proportional and nonproportional odds models for ordinal responses, and bivariate probit and bivariate logistic models for corrected binary responses.

Yee and Wild (1996) described VGLMs as a model of the conditional distribution of $\boldsymbol{Y}$ given $\boldsymbol{x}$ of the form

$$f\left(\boldsymbol{y}|\boldsymbol{x};\mathbf{B}\right) \;=\; h\left(\boldsymbol{y},\eta_1,\ldots,\eta_M\right), \tag{1.3}$$

where, the observed response $\boldsymbol{y}$ is a q-dimensional vector $(q \geq 1)$, $h\left(\cdot\right)$ is some known function, $\mathbf{B} \;=\; \left(\boldsymbol{\beta}_1\,\boldsymbol{\beta}_2\,\cdots\,\boldsymbol{\beta}_M\right)$ is a $p \times M$ matrix of unknown regression coefficients. VGLMs model each of a set of parameters as a linear combination of the covariate variables $\boldsymbol{x}$, specified in terms of a parametric class written as

$$\eta_j \;=\; \eta_j(\boldsymbol{x}) \;=\; \beta_{(j)1}x_1 + \beta_{(j)2}x_2 + \cdots + \beta_{(j)p}x_p, \qquad j \;=\; 1,\ldots,M. \tag{1.4}$$

VGAMs extend (1.4) to

$$\eta_j \;=\; \eta_j(\boldsymbol{x}) \;=\; \beta_{(j)1}x_1 + f_{(j)2}(x_2) + \cdots + f_{(j)p}(x_p), \qquad j \;=\; 1,\ldots,M. \tag{1.5}$$

That is, VGAMs specify the model in terms of smooth functions rather than linear functions. As with GLMs, VGLMs can be fitted using the IRLS algorithm in the same manner as it was done for the GLM class.

For VGAMs, once again, the $f_{(j)k}(x_k)$ are fitted using smoothers. In fact, $\boldsymbol{f}_k(x_k) = \big(f_{(1)k}(x_k),\ldots,f_{(M)k}(x_k)\big)^T$ are fitted simultaneously using *vector smoothers*. Like the original GAMs based on the backfitting approach of Hastie and Tibshirani (1990), VGAMs can be estimated by IRLS combined with modified vector backfitting using vector splines. The most complete reference for VGAMs is Yee (2015b). Only one type of smoother, the vector cubic smoothing spline smoother, is currently implemented by Yee's R package VGAM. The theory and software of VGAMs are based on the backfitting approach proposed by Hastie and Tibshirani (1990) and thus inherit the same difficulties for integrating automatic smoothness-estimation procedures. Current software implementations of the VGAM framework, the R package VGAM, do not deal with this problem satisfactorily. In fact, the backfitting approach obtained from the VGAM library defaults is likely to overfit when the shapes are less complex or results in the model underfitting when the shapes are more complex. Papers such as Marra and Radice (2010) demonstrated the impact of the degree of smoothness choice on the shape of the estimated smooth functions using the backfitting approach. They showed that the estimated curves obtained from GAMs based on the backfitting approach using the default settings are more wiggly than they should be when the shapes are less complex. This indicates that the absence of a numerical procedure for smoothing parameter estimation leads to the model overfitting. On the other hand, with more complicated trend shapes, backfitting resulted in the model underfitting because the degrees of freedom from the default settings did not provide enough flexibility. The development of automatic smoothness-estimation procedure for VGAMs is thus highly desirable.

The more modern GAM formulation based on penalized regression splines (Wood, 2000, 2004, 2006b, 2008) overcomes the issue of estimating the amount of smoothing for smooth terms of GAMs based on the backfitting approach. In this research, we extended these ideas to the vector GAM class. The main challenge for this thesis was to develop a penalized regression spline formulation of VGAMs, together with automated smoothness estimation and then to implement these methods in R. Our approach provided an efficient computational method for automatic smoothing parameter selection for a VGAM class covering a wide range of multivariate response types and models. This way, an advantage of automatic smoothing parameter selection was conveyed to a very large class of models. Currently the VGLM/VGAM classes are very large; potentially, hundreds of models lie within this framework.

1.1 Goals

The main purpose of this research is to introduce an alternative estimation procedure based on penalized regression splines for estimating model coefficients to the VGAM class, and to integrate an efficient computational method for automatic multiple smoothing parameter selection used to determine the functional shape of any relationship from the data into the VGAM framework. To achieve this, more specific goals include the following.

1. Developing a fitting procedure based on the penalized regression spline (P-splines) approach of Eilers and Marx (1996), Marx and Eilers (1998) and Wood (2006b) for the full range of VGAM models including complications such as constraints on model terms (Yee and Wild, 1996).

2. Developing a stable and efficient smoothing parameter selection procedure for the VGAM framework by generalizing the approaches of Wood (2004), Marra and Radice (2011), Marra

et al. (2013b), Marra and Radice (2013), Marra (2013), Marra et al. (2013a), and Radice et al. (2015). The plan involved:

a) developing the general forms of smoothing parameter selection criteria such as a generalized cross validation (GCV) and the unbiased risk estimator (UBRE) for VGAMs, and

b) implementing computational methods for smoothing parameter selection for the VGAM framework by generalizing the approach of Wood (2004).

3. Implementing the above methods in R.

4. Investigating the practical performance of the method proposed (P-spline vector generalized additive models) and comparing it to that of the existing approaches (VGAMs based on the backfitting approach) via simulation.

5. Applying the new approach to data from a cross-sectional workforce study combined with a health survey from New Zealand during the 1990s, and data from a survey study of the pregnancy and birth process during 1990 – 2004, with multivariate response types and models.

1.2 Outline

The outline of the thesis is as follows. Chapter 2 begins by discussing theoretical and practical aspects of GAMs based on the penalized likelihood approach with regression splines (Eilers and Marx, 1996; Marx and Eilers, 1998; Wood, 2006b). It then discusses the basics of smoothing parameter estimation and the effect of smoothing parameter choices on the shape of the estimated smooth functions. Chapter 3 summarizes the development of VGLMs/VGAMs emphasizing the theory required for model construction, constraint matrices and model estimation. Chapter 4 develops the P-spline formulation of the VGAM class and computational algorithms for the case of given smoothing parameters. Chapter 5 develops the theory and computational details for automated smoothing-parameter estimation and compares the new and existing methods using simulation. Chapter 6 applies the new method developed in this study to data from a cross-sectional workforce study combined with a health survey from New Zealand during the 1990s, and data from a survey study of the pregnancy and birth process during $1990-2004$, with several multivariate response types and models, which include the multinomial logit, proportional and non-proportional odds models, bivariate logistic model, and the LMS method for quantile regression. The thesis concludes with a discussion of the new approach and some suggestions for future research in Chapter 7.

Chapter 2

GAMs based on Penalized Regression Splines

Generalized additive models (GAMs) as introduced by Hastie and Tibshirani (1990) were fitted using the classic backfitting approach. More recently, other procedures have been employed for estimating GAMs. Marx and Eilers (1998) and Wood (2000, 2004, 2006b) developed GAMs using the penalized regression spline approach. When using such an approach, GAMs can be estimated by penalized regression methods, and the appropriate degree of smoothness for smooth terms can be automatically estimated from the data using well-founded criteria such as generalized cross-validation (GCV) or the Akaike information criterion (AIC).

In this chapter, the focus is on penalized regression-spline methods. Our objective is to present the basic ideas of the penalized likelihood approach proposed by Marx and Eilers (1998) and Wood (2000, 2004, 2006b) in a way that will lay the ground work for extending these approaches to the vector GAM setting of Yee and Wild (1996).

We begin by reviewing some of the basics of generalized linear models (GLMs) and GAMs based on the classic backfitting of Hastie and Tibshirani (1990). We will then describe some theoretical and practical aspects that underpin the GAM approach based on the penalized likelihood approach with regression splines of Wood (2000, 2004, 2006b) and Marx and Eilers (1998). The basics of smoothing parameter estimation and the effect of smoothing parameter choices on the shape of the estimated smooth functions will be discussed. The methodology will be then illustrated by modeling the kyphosis data and daily air pollution and daily death rate.

2.1 GLMs

Nelder and Wedderburn (1972) introduced the class of generalized linear models. GLMs specify a relationship between the mean of the random variable and a function of a linear combination of predictors. This generalization allows for a response distribution to follow any distribution from the exponential family, and allows a description of the variance as a function of the mean. The family of distributions includes the Gaussian or normal, Poisson, binomial, gamma, and inverse Gaussian. GLMs have the basic structure

$$g\,(\mu_i) \;=\; \eta_i \;=\; \boldsymbol{x}_i^T \boldsymbol{\beta},$$

where $\mu_i = \mathrm{E}(Y_i)$, $g(\cdot)$ is a monotone, differentiable function called the link function, $\boldsymbol{x}_i$ is a $p \times 1$ vector of explanatory variables, and $\boldsymbol{\beta}$ is the $p \times 1$ vector of parameters. In addition, the random responses, Y_i, are independent from a distribution belonging to the exponential family. The probability density functions are of the form

$$f(y;\theta,\phi) \;=\; \exp\left[\frac{y\theta - b(\theta)}{a(\phi,\omega)} + c(y,\phi)\right],$$

where $\theta \in \mathbb{R}$ and $\phi \in \mathbb{R}^+$ are parameters, ω is some known constant, and $a(\cdot), b(\cdot)$ and $c(\cdot)$ are arbitrary functions. The parameter θ is known as the *natural parameter*. The parameter ϕ is the *dispersion parameter*, and the function $a(\phi,\omega)$ can be written in the form ϕ/ω. The mean and variance of such a distribution are $\mathrm{E}(Y) = b'(\theta) = \mu$ and $\mathrm{var}(Y) = a(\phi,\omega)\, b''(\theta)$, respectively. We can define a function $\boldsymbol{V}(\mu) = b''(\theta)/\omega$, such that $\mathrm{var}(Y) = \boldsymbol{V}(\mu)\,\phi$. Then, we wish to estimate parameters $\boldsymbol{\beta}$ which are related to the Y_is through $\mathrm{E}(Y_i) = \mu_i$ and $g(\mu_i) = \boldsymbol{x}_i^T \boldsymbol{\beta}$. Thus the log-likelihood function is

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[\frac{y_i \theta_i - b(\theta_i)}{a(\phi, \omega)} + c(y_i, \phi) \right]. \tag{2.1}$$

The vector $\boldsymbol{\beta}$ can be estimated by maximizing (2.1). Maximization proceeds by partially differentiating ℓ with respect to each element of $\boldsymbol{\beta}$, setting the resulting expressions to zero, and solving for $\boldsymbol{\beta}$. For GLMs, maximum likelihood estimators are obtained by an iteratively reweighted least squares (IRLS) procedure. We will discuss the IRLS algorithm for the exponential-family model in the next section.

2.1.1 IRLS

IRLS is an algorithm for calculating quantities of statistical interest using weighted least squares calculations iteratively. The IRLS fitting algorithm applies naturally for GLMs as follows. We begin by considering the use of Newton's method to maximize the log-likelihood $\ell(\boldsymbol{\beta})$. This provides an iterative procedure

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left(\frac{\partial^2 \ell}{\partial \boldsymbol{\beta}\, \partial \boldsymbol{\beta}^T} \right)^{-1} \frac{\partial \ell}{\partial \boldsymbol{\beta}}, \tag{2.2}$$

where the derivatives are computed at $\boldsymbol{\beta}^{(t)}$ (t is the iteration number). Since we know that $\mathrm{var}(Y_i) = \boldsymbol{V}(\mu_i)\, \phi$, by using the chain rule, we can express the scalar derivatives of vector $\boldsymbol{\beta}$ as

$$\frac{\partial \ell}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i - \mu_i}{\mathrm{var}\,(Y_i)} \left(\frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i} \right) x_{ij}, \qquad \text{for} \quad j = 1, \ldots, p.$$

We can replace the second derivatives in (2.2) with their expectation. This gives a modification of Newton's method known as the Fisher scoring method. The second derivative term can be written as

$$-\mathrm{E} \left(\frac{\partial^2 \ell}{\partial \beta_j\, \partial \beta_k} \right) = \sum_{i=1}^{n} \left(\frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i} \right)^2 \frac{x_{ij}\, x_{ik}}{\mathrm{var}\,(Y_i)}.$$

If $\mathbf{W}$ is a diagonal matrix with diagonal elements

$$w_i = \frac{1}{\mathrm{var}\,(Y_i)} \left(\frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i} \right)^2,$$

and $\boldsymbol{z}$ is a vector of pseudodata (working response) with ith element

$$\boldsymbol{z}_i \;=\; \eta_i + (y_i - \mu_i)\left(\frac{\mathrm{d}\eta_i}{\mathrm{d}\mu_i}\right),$$

then, at iteration t, the Fisher-scoring modification of the algorithm (2.2) can be written in matrix notation as

$$\mathbf{X}^T\,\mathbf{W}^{(t)}\,\mathbf{X}\,\boldsymbol{\beta}^{(t+1)} \;=\; \mathbf{X}^T\,\mathbf{W}^{(t)}\,\boldsymbol{z}^{(t)}. \tag{2.3}$$

The maximum likelihood estimate of $\boldsymbol{\beta}$ is then obtained as follows. An initial approximation $\boldsymbol{\beta}^{(0)}$ is chosen to give an initial estimate of $\boldsymbol{z}^{(0)}$ and $\mathbf{W}^{(0)}$. Then (2.3) is solved in order to obtain $\boldsymbol{\beta}^{(1)}$, which is used to get improved new values for $\boldsymbol{z}^{(t)}$ and $\mathbf{W}^{(t)}$, and so on until adequate convergence is achieved. The maximum likelihood estimate $\boldsymbol{\beta}^{(t+1)}$ is taken when the difference between successive approximations $\boldsymbol{\beta}^{(t)}$ and $\boldsymbol{\beta}^{(t+1)}$ are sufficiently small.

2.2 GAMs

Classical GAMs are GLMs where y has a distribution in the exponential family, and the linear predictor η is specified as a sum of smooth functions of p covariates $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^T$, where $x_1 \equiv 1$ corresponds to the intercept. GAMs have systematic component

$$g\,(\mu) \;=\; \eta\,(\boldsymbol{x}) \;=\; \beta_1 x_1 + f_2\,(x_2) + \cdots + f_p\,(x_p)\,, \tag{2.4}$$

where the f_k's are arbitrary smooth functions. GAMs specify the model in terms of ‘smooth’ functions instead of using only parametric relationships. This approach extends traditional GLMs by allowing the automatic computation of possible nonlinear effects of covariates on a response variable of interest. The term ‘additive’ refers to the multivariate assumption underlying the model which gives the p-predictor function η a low-dimensional additive structure. The additive predictor η can be a simple term, a semiparametric term and a full additive model as in (2.4). In its simple form it gains simplicity by making a no-interaction assumption, though some forms of interaction can be incorporated by adding term like $f_j(x_1 \cdot x_3)$. No interaction models are very attractive as they are much more easier to interpret than a p-dimensional multivariate surface

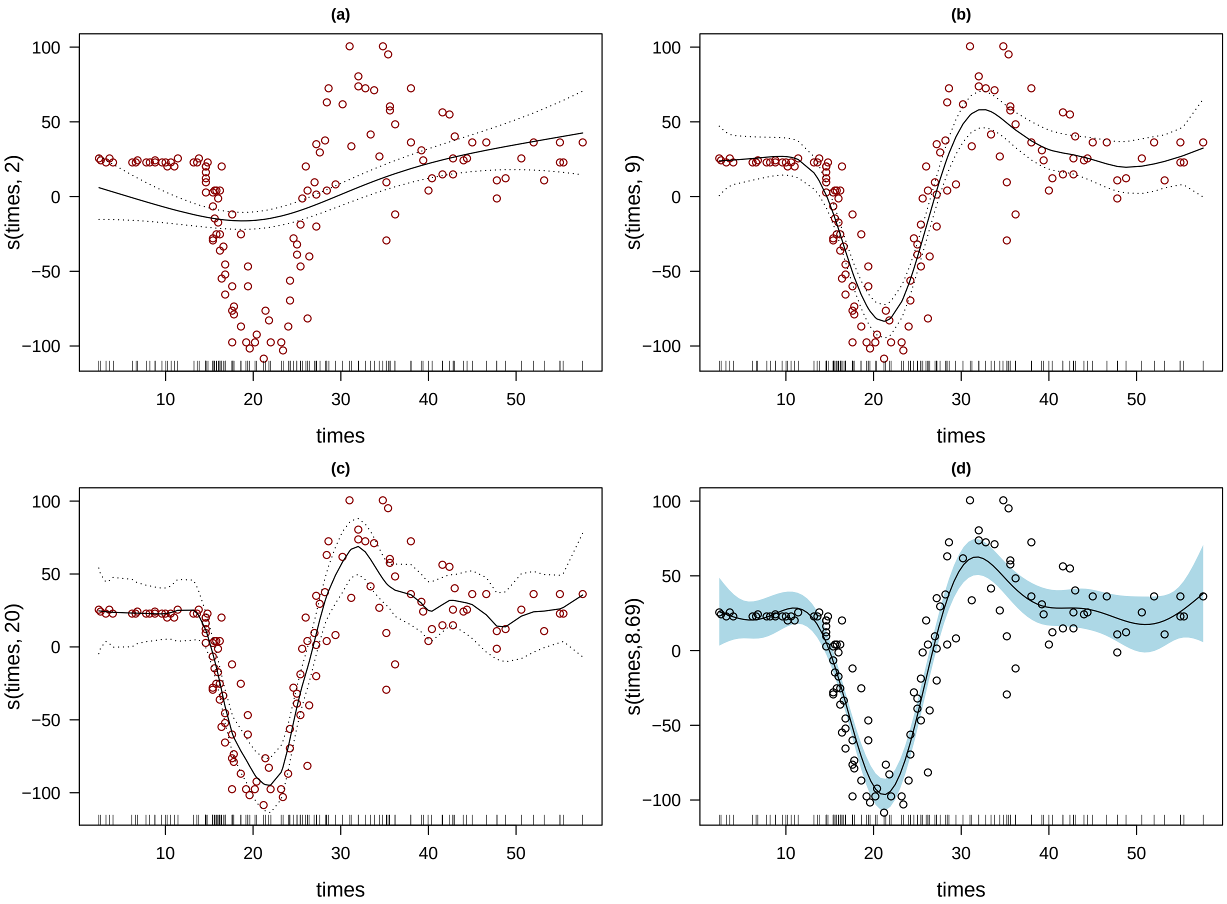


Figure 2.1: (a) – (c): Smoothing spline fits to a simulated motorcycle accident dataset (`mcycle` in MASS).(a) – (c) using three different values of degrees of smoothness. The dash lines are approximate 95% pointwise confidence intervals. (d): with 95% confidence shading, uses penalized regression splines with integrated smoothness estimation.

which allows arbitrary interaction in all dimensions. GAMs as described by Hastie and Tibshirani (1990) are based on the *local scoring algorithm*, which iteratively fits a weighted additive model by backfitting. The backfitting approach has the advantage of allowing the component functions of an additive model to be represented using almost any smoothing or modeling technique. But choosing a 'good' degree of smoothness is difficult with this approach. Wood (2006b) on the other hand, developed an approach to fitting GAMs using a more amenable penalized likelihood-based method. In contrast to classical backfitting, his penalized likelihood framework allows an efficient computational method for automatic multiple smoothing-parameter selection, which can automatically determine the shape of non-linear terms to be used from the data. This approach is a big advance on manually tuning some spline parameters, which is used in a classic backfitting approach. Fig. 2.1 (a) – (c) shows how the shape of the estimated function can be affected by the manual choice of degree of smoothness of a smooth term. By changing the value of the degree of smoothness, a variety of models of different smoothness can be obtained. Wood (2006b)'s penalized likelihood approach has a major advantage over the fixed-degree smoothing approach by Hastie and Tibshirani

Several methods have been proposed to achieve the objective of fitting of smooth function of GAMs. Hastie and Tibshirani (1990) minimized a weighted penalized least squares of GAMs by modified backfitting. They estimated the component functions by using simple linear scatterplot smoothers and standard least squares methods. Smoothing-parameter estimation is difficult to integrate into this approach. Wahba (1990), Wahba et al. (1995) and Gu (2002) introduced smoothing-spline analysis-of-variance (SS-ANOVA) which provides an underlying mathematical theory for function estimation, including GAMs. Gu and Wahba (1991) also developed an algorithm for estimating smoothing parameters for whole generalized spline smoothing (GSS) models and generalized this algorithm to GAMs that employs smoothing splines. However, their methods are extremely computational expensive. Other procedures that have been proposed for fitting GAMs are the marginal integration method by Linton and Nielsen (1995) and the penalized-based boosting procedure by Tutz and Binder (2006).

Due to the high computational cost of the SS-ANOVA approach and the problem of integrating the smoothing parameter estimation in the Hastie and Tibshirani (1990) methods, several authors have proposed using penalized regression splines for GAM modeling (see Hastie and Tibshirani 1990; Hastie 1996; Eilers and Marx 1996; Wand 2000; Wood 2000, 2004, 2006b). For example, Hastie and Tibshirani (1990) studied the use of regression splines, which a smooth function is modeled as the sum of B-splines. Eilers and Marx (1996) and Marx and Eilers (1998) developed P-splines based on a B-spline basis, usually defined on equally-spaced knots, with a penalty based on finite differences of the coefficients on adjacent B-splines applied directly to the parameter, to control function “wiggliness”. B-splines were developed as a very stable basis for large scale spline interpolation (see De Boor, 1978), and can be used as a parametric way to represent a nonparametric function in regression splines. With the P-spline method, GAMs can be then transformed into the GLM framework. Consequently, all smooth components are estimated simultaneously. P-splines are very attractive for implementing the GAM framework: not only are they extremely easy to set up and use, they also allow a great deal of flexibility. For example, users can combine any order of penalty with any order of B-splines basis, as they see fit. More details of P-splines will be provided in section 2.3. In a series of papers, Marra and Radice (2011), Marra et al. (2013b), Marra (2013), Marra et al. (2013a), and Radice et al. (2015) extended the penalized likelihood approach with regression splines to a bivariate binary response modeling problem. They showed that once a basis for the smooth functions has been chosen, together with associated measures of function wiggliness, the penalized likelihood maximization problem can be solved by penalized iteratively re-weighted least squares (P-IRLS), while the smoothing parameters can be estimated using related criteria.

Another interesting approach for fitting semiparametric regression is penalized splines. Penalized splines are very similar to smoothing splines. Actually, they are a generalization of smoothing splines that allow more flexible choices of the spline model, the basis function and the penalty. O'Sullivan (Section 3, 1986) represented penalized splines based on B-spline basis functions. His penalized splines are a direct generalization of smoothing splines and come with attractive properties such as *natural boundary behavior*. Wand and Ormerod (2008) studied the use of O'Sullivan penalized splines and formulated an exact algebraic expression for the corresponding penalty matrix when the basis consists of truncated power splines or B-splines. They reformulated a model formulation of O'Sullivan penalized splines in a convenient manner for implementing in a computing language such as R.

Efficient smoothing-parameter selection methods are critically important for GAM modeling. There are a number of possible methods for automatic selection of the smoothing parameters in GAMs. For example, a generalized cross validation (GCV) (Wahba and Craven, 1978), the Akaike information criterion (AIC), unbiased risk estimator (UBRE) (see Wood, 2006b) and generalized maximum likelihood (GML) (see Gu and Wahba, 1991). Gu and Wahba (1991) optimized GCV and GML scores with multiple smoothing parameters using a modified version of the Newton method. Their approaches provided a well-founded smoothing-parameter selection, as well as good coverage probabilities of confidence intervals. However, this method comes at high computational cost. Wood (2000) achieved developing an efficient smoothing-parameter selection method for GAM models estimated by penalized least squares. He developed a GCV for multiple smoothing parameter selection based on a similar optimization strategy of Gu and Wahba's (1991) approach. Although the methods from Wood (2000) are usually effective, they cannot deal adequately with the numerical rank-deficiency of the GAMs fitting problem. In 2004, he improved the multiple smoothing parameter estimation method to deal with fixed penalties. The method is based on the pivoted QR decomposition and the singular value decomposition. It is capable of detecting and coping with numerical rank-deficiency (see Wood 2004). More recently, Wood (2011) developed the new restricted maximum likelihood (REML) and maximum (marginal) likelihood (ML) methods for the estimation of smoothing parameters. These methods not only cope with numerical rank deficiency in the fitted model, they also provide a slight improvement in numerical robustness.

2.2.1 GAMs based on penalized regression splines

We now move on to consider how GAMs can be reconceived using penalized regression splines. As described in Section 2.2, GAMs model a response variable, Y_i, using a model structure of the form (2.4), where $\mu_i = \mathrm{E}(Y_i)$, Y_i has a distribution in the exponential family and the $f_k(\cdot)$ are a smooth function of covariates x_k. To estimate model (2.4) a basis for each smooth function must be chosen, as well as "wiggliness" measures for the smooth terms. The smooth terms have been represented using regression splines such as cubic splines (Wood, 2006b), P-splines (Eilers and Marx, 1996; Marx and Eilers, 1998) and thin plate regression splines (see Duchon, 1977; Wood, 2003). Specifically, the regression-spline representation for an explanatory variable is made up of a linear combination of known basis functions, $B_{sk}(\cdot)$, and unknown regression parameters, β_{sk},

$$f_k(x_k) \;=\; \sum_{s=1}^{\mathrm{S}} \beta_{sk}\, B_{sk}(x_k). \tag{2.5}$$

Here, k indicates the smooth term for the kth explanatory variable and S is the number of basis functions. Each smooth component is subject to a centering constraint such as $\sum_i f_k(x_{ik}) \;=\; 0$ to ensure that the model is identifiable. Given a basis like (2.5) for each smooth, Wood (2006b) wrote model equation (2.4) as

$$\eta_i \;=\; \mathbf{X}_i\, \boldsymbol{\beta}, \tag{2.6}$$

where $\mathbf{X}_i \;=\; (1 \,|\, \mathbf{X}_{i2} \,|\, \ldots \,|\, \mathbf{X}_{ip})$ is a row of the model matrix $\mathbf{X}$, where $\mathbf{X}$ is constructed from the basis function values given in (2.5) and the vector $\boldsymbol{\beta} \;=\; \left(\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T \ldots, \boldsymbol{\beta}_p^T\right)^T$ contains the β-parameters to be estimated. (2.6) is a GLM, which can be fitted by maximum likelihood.

Wood (2006b) controlled the smoothness for each term by applying a set of penalties to the likelihood $\ell\,(\boldsymbol{\beta})$. A penalized approach was then adopted. Therefore, a penalized likelihood for the model can be written as

$$\ell_p(\boldsymbol{\beta}) \;=\; \ell(\boldsymbol{\beta}) - \frac{1}{2} \sum_k \lambda_k \int \{f_k^{''}(x_k)\}^2 \,\mathrm{d}x_k \tag{2.7}$$

($k \;=\; 1$ relates to the intercept). Here, a wiggliness penalty has been included for each smooth function. The λ_k's are the smoothing parameters that control the trade-off between model fit and model smoothness. The integral of the square of the second derivatives of a fitted function in (2.7) is commonly used as a smoothness penalty (see Reinsch, 1967; Green and Silverman, 1993). To show how this penalty can be calculated for a given $f_k(\cdot)$ that have a basis expansion (2.5), we write

$$f_k^{''}(x_k) \;=\; \sum_{s=1}^{\mathrm{S}} \beta_{sk}\, B_{sk}^{''}(x_k) \;=\; \boldsymbol{b}_k^{''}(x_k)^T \boldsymbol{\beta}_k,$$

where $\boldsymbol{b}_k^{''}(x_k)$ is a vector containing the second derivatives of the basis functions for the kth smooth term with respect to x_k and $\boldsymbol{\beta}_k$ is a parameter vector. We then have

$$\int \{f_k^{''}(x_k)\}^2 \,\mathrm{d}x_k \;=\; \int \boldsymbol{\beta}_k^T\, \boldsymbol{b}_k^{''}(x_k)^T\, \boldsymbol{b}_k^{''}(x_k)\, \boldsymbol{\beta}_k \,\mathrm{d}x_k \;=\; \boldsymbol{\beta}_k^T\, \mathbf{S}_k\, \boldsymbol{\beta}_k,$$

where

$$\mathbf{S}_k \;=\; \int \boldsymbol{b}_k''(x_k)^T \boldsymbol{b}_k''(x_k)\,\mathrm{d}x_k.$$

The penalty term in (2.7) can be written as

$$\sum_k \lambda_k \int \{f_k''(x_k)\}^2\,\mathrm{d}x_k \;=\; \sum_k \lambda_k\, \boldsymbol{\beta}_k^T\, \mathbf{S}_k\, \boldsymbol{\beta}_k.$$

Defining $\mathbf{S} \;=\; \mathrm{blockdiag}\,(0,\, \lambda_2\,\mathbf{S}_2,\, \ldots,\, \lambda_p\,\mathbf{S}_p)$ then

$$\sum_k \lambda_k \int \{f_k''(x_k)\}^2\,\mathrm{d}x_k \;=\; \boldsymbol{\beta}^T\,\mathbf{S}\,\boldsymbol{\beta}.$$

Since smooth functions f_k are linear in parameters $\boldsymbol{\beta}$, the penalty $\sum_k \lambda_k \int \{f_k''(x_k)\}^2\,\mathrm{d}x_k$ can always be written as a quadratic form in $\boldsymbol{\beta}$, with the penalty matrix $\mathbf{S}$. Wand and Ormerod (2008) studied the use of O'Sullivan penalized splines and formulated an exact algebraic expression for the roughness penalties $\int \{f_k''(x_k)\}^2\,\mathrm{d}x_k$ when the basis consists of truncated power splines or B-splines. To formulate this penalty, they wrote

$$\boldsymbol{\Omega}_{kk'}^{(m)} \;=\; \int_a^b B_{2m-1,k}^{(m)}(x) B_{2m-1,k'}^{(m)}(x)\,\mathrm{d}x_k, \tag{2.8}$$

where $B_{2m-1,1}, \ldots, B_{2m-1,K+2m}$ are B-splines of degree $(2m-1)$ and the sequence of knots is given by:

$$a = K_1 = \cdots = K_{2m} < K_{2m+1} < \cdots < K_{2m+K} < K_{2m+K+1} = \cdots = K_{4m+K} = b.$$

They then formulated the penalty term in (2.8) as

$$\boldsymbol{\Omega}^{(m)} \;=\; \left(\widetilde{\mathbf{B}}^{(m)}\right)^T \mathrm{diag}(\boldsymbol{w})\,\widetilde{\mathbf{B}}^{(m)}. \tag{2.9}$$

Here, they defined $\widetilde{\mathbf{B}}^{(m)}$ as the $(2m-1)(K+4m-1)\times(K+2m)$ matrix with (i,j)th element given by $B_{2m-1,j}^{(m)}(\widetilde{x}_i)$, and $\boldsymbol{w}$ is a $(2m-1)(K+4m-1)\times 1$ vector with ith element given by w_i. The values of $\widetilde{x}_i$ and w_i are obtained as follows:

$$\widetilde{x}_{(2m-1)(l-1)+l'+1} = K_l + l'h_{m,l}, \qquad w_{(2m-1)(l-1)+l'+1} = h_{m,l^{\Omega}m,l'}.$$

The matrix expressions for the penalty of O'Sullivan splines that they developed are easy to set up and implement in a matrix-based computing language such as R. Full details about the exact matrix expression for the penalty of O'Sullivan splines are given in Wand and Ormerod (Section 6, 2008). Several penalty functions such as $\int\{f_k'(x_k)\}^2\,\mathrm{d}x_k$, $\int\{f_k'''(x_k)\}^2\,\mathrm{d}x_k$, or the *discrete* approximate wiggliness measurement by Eilers and Marx (1996) have been used in the literature. In practice, the maximization of the penalized likelihood can be performed using penalized iteratively reweighted least squares (P-IRLS) in a way that we will now describe. The penalized likelihood can be written in general matrix-vector form, so that (2.7) becomes

$$\ell_p(\boldsymbol{\beta}) \;=\; \ell(\boldsymbol{\beta}) - \frac{1}{2}\,\boldsymbol{\beta}^T\mathbf{S}\boldsymbol{\beta}. \tag{2.10}$$

Wood (Chapter 4, 2006b) maximized (2.10) by solving the partial derivative of ℓ_p with respect to each element of $\boldsymbol{\beta}$. He then set the resulting system of equations to zero:

$$\frac{\partial\ell_p}{\partial\beta_k} \;=\; \frac{\partial\ell_p}{\partial\beta_k} - [\mathbf{S}\boldsymbol{\beta}]_k \;=\; \frac{1}{\phi}\sum_{i=1}^{n}\frac{y_i-\mu_i}{V(\mu_i)}\left(\frac{\partial\mu_i}{\partial\beta_k}\right) - [\mathbf{S}\boldsymbol{\beta}]_k \;=\; 0, \tag{2.11}$$

where $[\cdot]_k$ denotes the kth row of a vector. The solution to (2.11) solves the penalized non-linear weighted least squares problem

$$\text{minimize} \qquad S_p \;=\; \sum_{i=1}^{n}\frac{(y_i-\mu_i)^2}{\operatorname{var}(Y_i)} + \boldsymbol{\beta}^T\mathbf{S}\boldsymbol{\beta}, \tag{2.12}$$

where μ_i is treated as depending non-linearly on $\boldsymbol{\beta}$, while the $\operatorname{var}(Y_i)$ is treated as known. Wood (2006b) showed that a penalized non-linear least squares problem like (2.12) can be dealt with using a penalized version of iterative linear least squares. Penalized maximum likelihood estimation was achieved by the iterative solution of

$$\text{minimize} \left\|\sqrt{\mathbf{W}^{(t)}}\left(\boldsymbol{z}^{(t)} - \mathbf{X}\boldsymbol{\beta}\right)\right\|^2 + \boldsymbol{\beta}^T\mathbf{S}\boldsymbol{\beta}, \tag{2.13}$$

with respect to $\boldsymbol{\beta}$, where $\boldsymbol{z}^{(t)}$ is a vector of pseudodata defined as

$$z_i \;=\; g'\left(\mu_i^{(t)}\right)\left(y_i - \mu_i^{(t)}\right) + \mathbf{X}_i\widehat{\boldsymbol{\beta}}^{(t)},$$

and $\mathbf{W}^{(t)}$ is a diagonal matrix with diagonal elements

$$w_{ii}^{(t)} \;=\; \frac{1}{V\left(\mu_i^{(t)}\right)\,g'\left(\mu_i^{(t)}\right)^2}.$$

Given the smoothing parameters, the maximum penalized likelihood estimates of $\boldsymbol{\beta}$ were obtained as follows. The current $\widehat{\boldsymbol{\beta}}^{(t)}$ is used in order to calculate $\boldsymbol{z}$ and $\mathbf{W}$ and, next, the least squares objective (2.13) is minimized w.r.t. $\boldsymbol{\beta}$ to find the updated $\boldsymbol{\beta}^{(t+1)}$. Then the linear predictor $\boldsymbol{\eta}^{(t+1)}$ and fitted values $\boldsymbol{\mu}^{(t+1)}$ are evaluated, and a pseudodata $\boldsymbol{z}$ and the working weights $\mathbf{W}$ are updated. All steps are repeated until convergence is achieved. This procedure is known as P-IRLS.

Given the smoothing parameters, the maximum penalized likelihood estimates, $\widehat{\boldsymbol{\beta}}$ can then be obtained by using P-IRLS algorithm. It turns out that an iterative estimation for $\boldsymbol{\beta}$ is given by

$$\widehat{\boldsymbol{\beta}}^{(t+1)} = \left(\mathbf{X}^T\mathbf{W}^{(t)}\mathbf{X} + \mathbf{S}\right)^{-1}\mathbf{X}^T\mathbf{W}^{(t)}\boldsymbol{z}^{(t)}.$$

With the iterative equation above, it is fairly straightforward to show that the formal expression for the influence matrix (or hat matrix) for a GAM model is

$$\mathbf{A} = \mathbf{X}\left(\mathbf{X}^T\mathbf{W}\mathbf{X} + \mathbf{S}\right)^{-1}\mathbf{X}^T\mathbf{W}. \tag{2.14}$$

Therefore, the flexibility of the fitted model is measured by $\mathrm{tr}(\mathbf{A})$ defined as the effective degrees of freedom (EDF) (Wood (Chapter 4, 2006b)). If all smoothing parameters are equal to zero, the degrees of freedom of the model are equal to the dimension of parameter coefficients. The degrees of freedom decreases as the smoothing parameters increase.

2.2.2 Smoothing parameter estimation

The most traditional algorithm for smoothing-parameter selection involves the backfitting algorithm based on a univariate scatterplot-smoother applied iteratively and cubic smoothing splines. The idea of the backfitting is based on univariate smooth components which are applied iteratively. The backfitting algorithm starts by iteratively estimating the updated individual smooth component by smoothing partial residuals from the model. These residuals are formed using the current estimates of smooth components. This algorithm is repeated until the estimates stabilize. A stepwise procedure is integrated into the traditional backfitting in order to obtain the degrees of freedom for each smooth component. Cubic smoothing splines are used as a smoother and the smoothing parameters are selected by stepwise selection of the degrees of freedom for each component. The procedure begins with a model where all terms enter linearly and then improves the AIC by upgrading or downgrading the degrees of freedom for one component by one level. But this method of adapting backfitting to smoothing parameter estimation is very slow when each smooth term has one of a fairly large number of alternative degrees of freedom.

An alternative to smoothing parameter selection methods of GAMs is based on a penalized likelihood framework. Two such approaches have been proposed. The first group develops an efficient GCV or AIC based on smoothness selection methods for the simple additive model case. Those criteria are applied to each working additive model of the P-IRLS scheme which is used to fit the GAMs. Wood (2000, 2004, 2006b) adapted the GCV score and the UBRE criterion for smoothing parameter selection for GAM models estimated by penalized least squares. It can be shown that the estimation of smoothing parameters for GAMs can be achieved by minimizing the GCV score and the UBRE criterion with respect to multiple smoothing parameters (Wood, 2000, 2004, 2006b), where the GCV score and the UBRE criterion are in the form

$$\begin{aligned} \mathcal{V}_g^w &= \frac{n\|\sqrt{\mathbf{W}}\,(\boldsymbol{z}-\mathbf{X}\boldsymbol{\beta})\,\|^2}{[n-\mathrm{tr}\,(\mathbf{A})]^2}, \\ \mathcal{V}_u^w &= \frac{1}{n}\|\sqrt{\mathbf{W}}\,(\boldsymbol{z}-\mathbf{X}\boldsymbol{\beta})\,\|^2-\sigma^2+\frac{2}{n}\,\mathrm{tr}(\mathbf{A})\,\sigma^2, \end{aligned}$$

respectively. Here, $\sqrt{\mathbf{W}} = \left[(w_{ii})^{1/2}\right]$. A globally-applicable GCV score (see Hastie and Tibshirani, 1990; Wood, 2006b) and the UBRE criterion (Wood, 2006b) can be obtained by

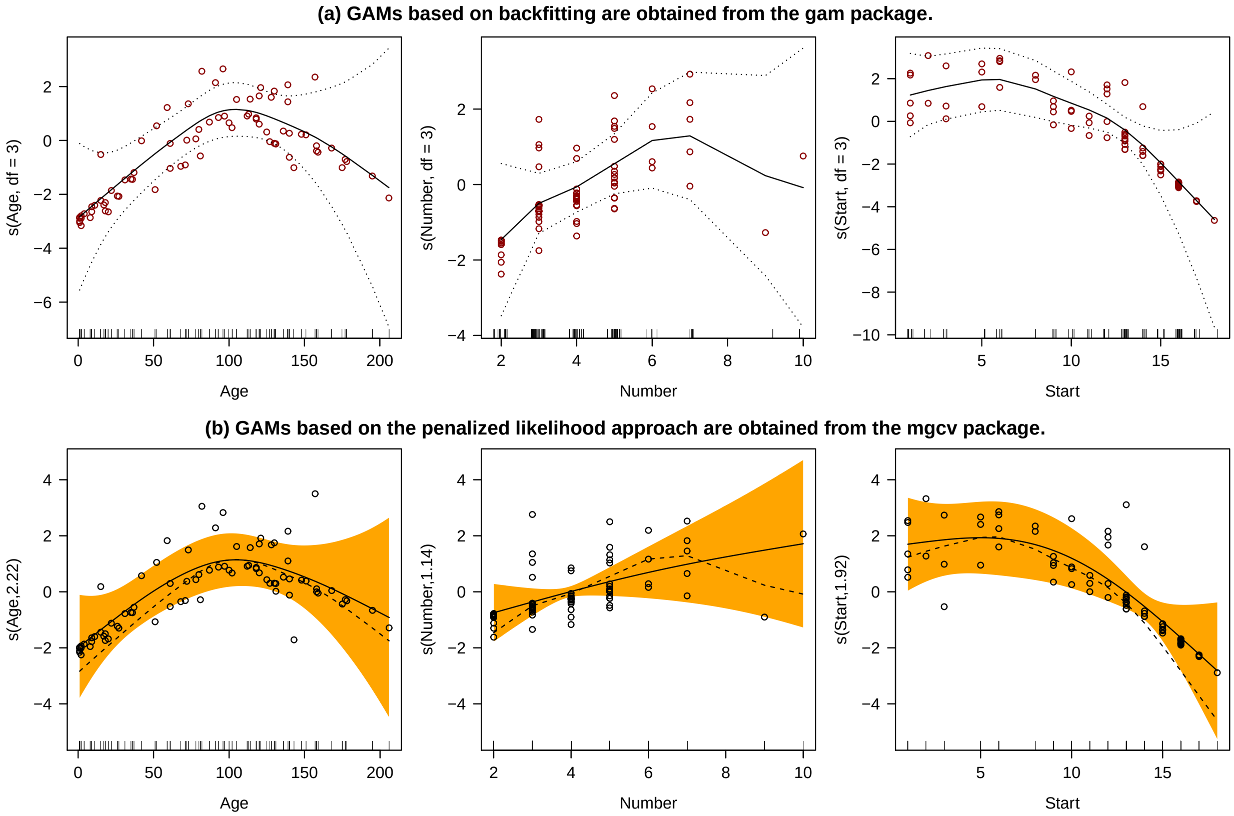
$$\begin{aligned} \mathcal{V}_g &= \frac{n\,Dev\,(\hat{\boldsymbol{\beta}})}{[n-\mathrm{tr}\,(\mathbf{A})]^2}, \\ \mathcal{V}_u &= \frac{1}{n}Dev\,(\hat{\boldsymbol{\beta}})-\sigma^2+\frac{2}{n}\,\mathrm{tr}(\mathbf{A})\,\sigma^2, \end{aligned}$$

where Dev is the deviance of the fitted model, and $\mathbf{A}$ is the influence matrix (cf. equation (2.14)). The model deviance Dev is defined as the saturated log-likelihood minus log-likelihood of the fitted model, all multiplied by 2ϕ. This deviance can be seen as the residual sum of squares for a linear model. According to Wood (2006b), there are two possible numerical strategies for estimating smoothing parameters: using $\mathcal{V}_g$ or $\mathcal{V}_v$ minimization. The first method, originally proposed by Gu (1992), is known as *performance iteration.* This method is achieved by minimizing $\mathcal{V}^w_{g/u}$ with respect to multiple smoothing parameters, and then smoothing parameters are chosen for each working penalized linear model of the P-IRLS iteration. In the second method, is *outer iteration* (Wood, 2006b), the maximization of $\mathcal{V}_{g/u}$ with respect to the model smoothing parameters can be done directly. This means that the P-IRLS process must be iterated to convergence for each trial set of smoothing parameters. For the second group, Wood (2011) treated GAMs as generalized linear mixed models (GLMMs) (see Ruppert et al., 2003), and in which the smoothing parameters are variance components which can be estimated by maximum (marginal) likelihood (ML) (Anderssen and Bloomfield, 1974), or restricted maximum likelihood (REML) in the non-generalized case. In the generalized case, the methods that are based on iterative fitting of working generalized linear mixed models (GLMM's) (see Breslow and Clayton, 1993) are used.

There are now a number of packages available implementing the GAM framework and related models for R. The GAM framework of Hastie and Tibshirani (1990) was implemented in the `gam` package by Hastie. In general, the functions in `gam` share many of the features of `glm()` and `lm()`, with some added flexibility. Gu (2002) introduced the `gss` R package developed for smoothing spline ANOVA models. Wood (2006b) developed the `mgcv` R package (Wood, 2007), which is based on his penalized likelihood approach with penalized regression splines. His software design is based somewhat on Hastie (Chambers and Hastie, 1993, chapter 7). The main difference is that smooth terms `s()` and `te()` are incorporated in the `gam()` model formula and the automatic smoothing parameter selection is provided. Yee (2008) describes the `VGAM` package, which is more similar to `gam` but for a wider class of models.

2.2.3 Illustration of GAMs using the penalized likelihood approach

We will now illustrate these approaches using the Kyphosis data from Hastie and Tibshirani (1990). These data come from Bell et al. (1989). The response outcome of interest is the `presence`


(a) GAMs based on backfitting are obtained from the gam package.

(b) GAMs based on the penalized likelihood approach are obtained from the mgcv package.

Figure 2.2: (a): The estimated smooth components of backfitting with partial residuals and pointwise standard errors. (b): The estimated smooth terms of the penalized likelihood approach with 95% Bayesian intervals. The estimated smooth curves of backfitting from (a) are overlaid in (b) represented by the dash lines in order to show the comparison.

(1) or the absence (0) of kyphosis, defined as postoperative spinal deformity in children. The available regressors are Age in months at the time of the surgery, the starting range of vertebrae levels involved in the surgery (Start), and the number of vertebrae involved (Number). The data are available in the data frame kyphosis from gam. There are 81 observations, resulting in 17 presences and 64 absences. Fig. 2.2 shows GAMs estimated using backfitting (Fig. 2.2 (a)) and penalized likelihood (Fig. 2.2 (b)). Penalized likelihood is performed using the mgcv package and backfitting computation is performed using the gam library. The goal of the analysis is to investigate the relationship between kyphosis and the three predictor variables. In order to investigate this relationship, a logistic additive model is used to describe the conditional probability of kyphosis given the predictor variables. We then fit the additive logistic model to

the three predictors using the function gam() in the mgcv package, where the default settings are thin plate regression splines and penalties based on the second-order derivatives. The automatic smoothing parameter selection is obtained through minimization of the UBRE. The dimension, k, of the basis for the predictors, Number, and Start is adjusted as these variables are heavily tied. The partial contributions of each predictor to the conditional probability of kyphosis with 95% Bayesian intervals are represented in (b) of Fig. 2.2. The EDF estimates of Age, Number, and Start are 2.22, 1.14, and 1.94 respectively. These EDF estimates support the presence of non-linearity as they are higher than 1. The smooth functions for Age and the starting range of vertebrae levels (Start) clearly show non-linear features, while the smooth term for Number of vertebrae involved shows much less curvature. The results from Fig. 2.2 (b) show that children aged about 100 months have higher risk of kyphosis than those of younger or older age. The risk of kyphosis decreases when the starting range of vertebrae levels increases. But conversely, the risk of kyphosis increases with the higher number of vertebrae levels involved.

We investigated the backfitting approach to determine whether it led to different conclusions. We followed Hastie and Tibshirani's 1990 suggestion of fitting each smooth term by using a smoothing spline with 3 degrees of freedom for each smooth term ($\texttt{df}_j$ = 3) and used the gam library. Note that an automatic smoothing parameter selection procedure is not available with the backfitting approach. The plots in Fig. 2.2 (a) are the estimated components with partial residuals and pointwise standard errors obtained from backfitting. Although the estimated components from the backfitting with gam library lie within the Bayesian intervals obtained from the penalized likelihood approach, the estimated points differ quite substantially from the estimated points obtained from the mgcv package when the number of vertebrae is higher than about 7. With backfitting, the relationship between kyphosis and the number of vertebrae from becomes negative suggesting that the risk of kyphosis goes down with higher number of vertebrae levels in contrast to the continued rise in the penalized likelihood curve (of course, there is almost no data in this region).

Marra and Radice (2010) studied the impact of smoothing parameter choice on the shape of the estimated smooth functions by investigating the performance of GAMs based on backfitting and GAMs based on the penalized likelihood approach. Their underlying objectives were to show the importance of implementing an automatic and stable multiple smoothing parameter selection procedure in a GAM framework. They compared the backfitting approach using the gam library to the penalized likelihood approach based on Wood (2006b). In their simulation study, they used the linear predictor of the form

$$\eta_i \;=\; s_1(x_{i1}) + s_2(x_{i2}) + s_3(x_{i3}), \tag{2.15}$$

where the three test functions in (2.15) represented smooth functions ranging from less complex to more complex. For observations, they generated data from four error models (gamma, binomial, normal, and Poisson) at each of three signal to noise ratios, at each of three sample sizes ($n \;=\; 250, 500, 1000$). In each case, they simulated the covariates independently from a uniform distribution on the unit interval. They then generated 500 replicate data sets and fitted GAMs using the two frameworks to each of 500 replicates at each sample size, model structure, distribution and error level combination. All computations for the two frameworks were preformed using the default settings. For each replicate and for each test function, Marra and Radice (2010) measured the performance of GAMs using the percentage mean squared error (PMSE), which they defined as

$$\text{PMSE} \;=\; \left\{\frac{1}{n}\sum_{i=1}^{n}\left(\widehat{s}_k(x_{ik}) - s_k(x_{ik})\right)^2\right\} \times\, 100, \qquad k \;=\; 1, 2, 3.$$

In the simulation study of Marra and Radice (2010), the penalized likelihood smooths with mgcv library defaults performed better than backfitting smooths with gam library defaults. The complexity of the shapes directly affects PMSE. The backfitting approach is likely to overfit when the shapes are less complex. Marra and Radice (2010) indicated that this is because the backfitting does not have any procedures to prevent complex smooth components when the data is not complex. On the other hand, with more complicated trend shapes, backfitting performed poorly because the degrees of freedom from the default settings did not provide enough flexibility. However, the function estimates by using backfitting could be improved by changing the target equivalent degrees of freedom, used as a smoothing parameter, which requires the users to tune these parameters.

2.3 P-splines

Eilers and Marx (1996) originally proposed the penalized likelihood using B-splines with a relatively large number of knots and a difference penalty on coefficients of adjacent B-splines for nonparametric modeling with one predictor variable. They called this method “P-splines”. Marx and Eilers (1998) extended the P-spline approach to include more than one predictor and aimed to fit all smooth GAM components simultaneously. They constructed each smooth component using *penalized* B-spline smoothers, which they called P-spline smoothers (P-spline GAMs were actually penalized GLMs). They were then able to directly fit GAMs using a slightly adjusted version of the Fisher scoring algorithm with all smooth GAM components being estimated at once.

P-splines are low-rank smoothers using B-spline basis functions. Eilers and Marx (1996) and Marx and Eilers (1998) used a relatively large number of equidistant knots, which would purposely overfit each B-spline component. To prevent overfitting, P-splines impose a difference penalty on adjacent B-spline coefficients during model fitting, which is designed to ensure the smoothness of the fitted model. The smoothing parameters of this approach are easily chosen through the minimization of well-founded criteria such as the AIC or the GCV score. P-splines are an efficient implementation of penalized regression splines, and come with many attractive properties:

1. P-splines use a sparse smoothing basis and penalty, enabling efficient and numerically stable computation.

2. Since only a small number of parameters are used to estimate GAMs, the resultant P-spline GAMs have a compact summary. This makes the use of P-splines easier for making predictions.

3. P-splines provide great flexibility. Users can choose: the degree of the B-splines basis, the order of the penalty, and the number of knots.

The next subsection sketches the basic theory of P-splines and discusses the representation and estimation of the smooth functions using penalized regression splines based on the P-spline approach. Then we will explain how GAMs can be constructed using the P-splines, and estimated using penalized regression methods. We also discuss in some details how optimal smoothing parameter values are obtained. An example of P-spline generalized additive modeling is provided using a study of the air pollution in Chicago.

2.3.1 Penalized regression splines based on P-splines

P-spline methods use B-splines as basis functions, together with a large number of equally-spaced knots. (Over-wiggly fitted models are prevented by applying a *discrete* approximate “wiggliness" penalty to the model fitting objective.) B-splines are a sequence of polynomial basis functions possessing special properties related to the continuity of the derivatives at the positions of the knots (De Boor, 1978). They are constructed from a set of polynomial pieces, connected at the certain values of knots, using a recursive algorithm given by De Boor (1978). Fig. 2.3 represents sequences of B-splines of degree 1 (linear B-splines), degree 2 (quadratic B-splines), and degree 3 (cubic B-splines) using equally-spaced knots. A single B-spline of degree q consists of $q+1$ polynomial pieces, cf. the individual curves depicted in Fig. 2.3, connected at q inner knots, where knots are the points on the horizontal axis at which the pieces merge together. For example, a single B-spline of degree 2 consists of 3 quadratic pieces, joined at two inner knots. In general, B-splines can be defined for an arbitrary grid of knots, but in our research we simplify and generalize the approach of Eilers and Marx (1996) and Marx and Eilers (1998). We will only use equidistant knots. More details on B-splines and related algorithms can be found in the books by De Boor (1978) and Dierckx (1995).

Let us suppose that there are n observations (x_i, y_i). A model represented by a single smooth

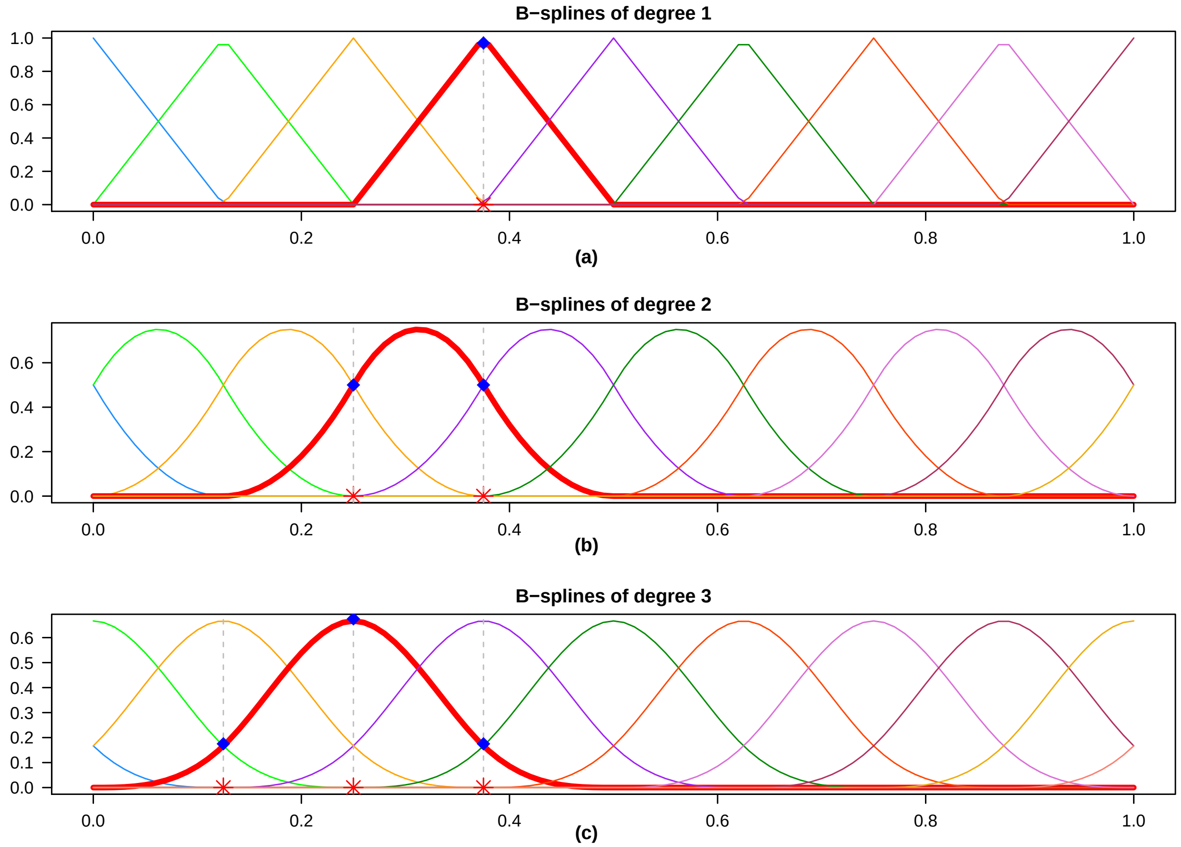


Figure 2.3: The sequence of B-splines of degrees 1 to 3.

function can be written in the form

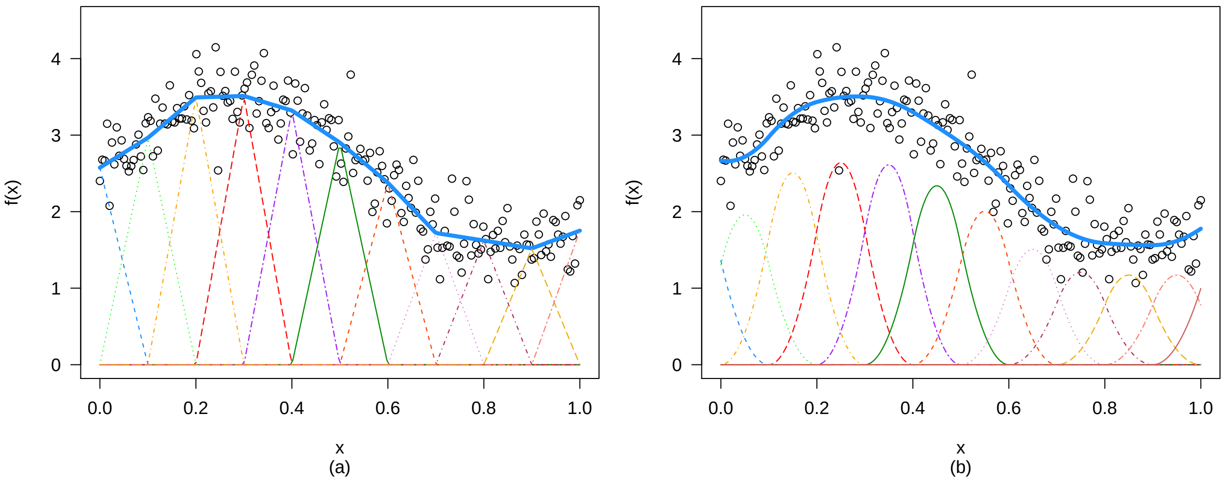
$$y_i \;=\; f\,(x_i) + \varepsilon_i, \qquad \text{for} \qquad i \;=\; 1, \ldots, n, \tag{2.16}$$

where y_i is a response variable, x_i is a covariate, $f\,(\cdot)$ is a smooth function, and $\boldsymbol{\varepsilon_i}$ are independent and identically distributed with mean zero and variance σ^2. Here, we consider a smooth function $f\,(\cdot)$ constructed from B-splines, each of degree q and a grid of knot points in $[L, U]$

$$f(x_i) \;=\; \sum_{s=1}^{\mathrm{S}} a_s\, B_s(x_i; q), \qquad \text{for} \qquad s = 1, \ldots, \mathrm{S}\;(< \;n). \tag{2.17}$$

Here, $B_s(x; q)$ is the value at x for the s-th B-spline, and $\{a_s\}$ are coefficients associated with the B-spline basis functions. These coefficients will be estimated as part of the model fitting. The model (2.16) can be estimated by minimizing the quantities

$$S \;=\; \sum_{i=1}^{n} \left\{ y_i - \sum_{s=1}^{\mathrm{S}} a_s B_s(x_i; q) \right\}^2 . \tag{2.18}$$



(a)

(b)

Figure 2.4: A fitted curve by 10 B-spline bases with (a) degree 1 and (b) degree 2.

When the degree of B-splines has been clearly defined in the context, we then use $B_s(x_i)$ instead of $B_s(x_i; q)$. In Fig. 2.4, we have fitted (2.16) using B-splines to the data simulated from the model

$$y = 2.5 + \sin(5.5x) + \varepsilon, \tag{2.19}$$

where $\varepsilon_i \sim \mathrm{N}(0, 0.3)$, and a covariate x, was generated from uniform $(0, 1)$. Fig. 2.4 (a) shows smooth curves obtained by 10 B-splines of degree 1 and Fig. 2.4 (b) does the same thing using B-splines of degree 2. The thin curves in Fig. 2.4 (a) and Fig. 2.4 (b) display each B-spline multiplied by its associated coefficient. The thick lines in Fig. 2.4 show the resulting smooth curves computed by the sum of estimated coefficients multiplied by the B-spline basis functions.

Although regression spline modeling as (2.18) is easy to understand and solve, in practice, the knot placement problems seriously affect the modeling results. One standard method to prevent these problems are to use penalized regression splines. With the penalized regression approach, a relatively large number of knots is used, but the penalty is applied to the model fitting in order to avoid the danger of over-fitting. Fig. 2.5 (a) shows a fitted curve of the model (2.19) using B-splines with a relatively large number of knots (16 B-splines of degree 3). With such a large number of knots, the fitted model is over-fitted (wiggly) as shown in Fig. 2.5 (a). One can control the wiggliness of the fitted model by incorporating a penalty during the model

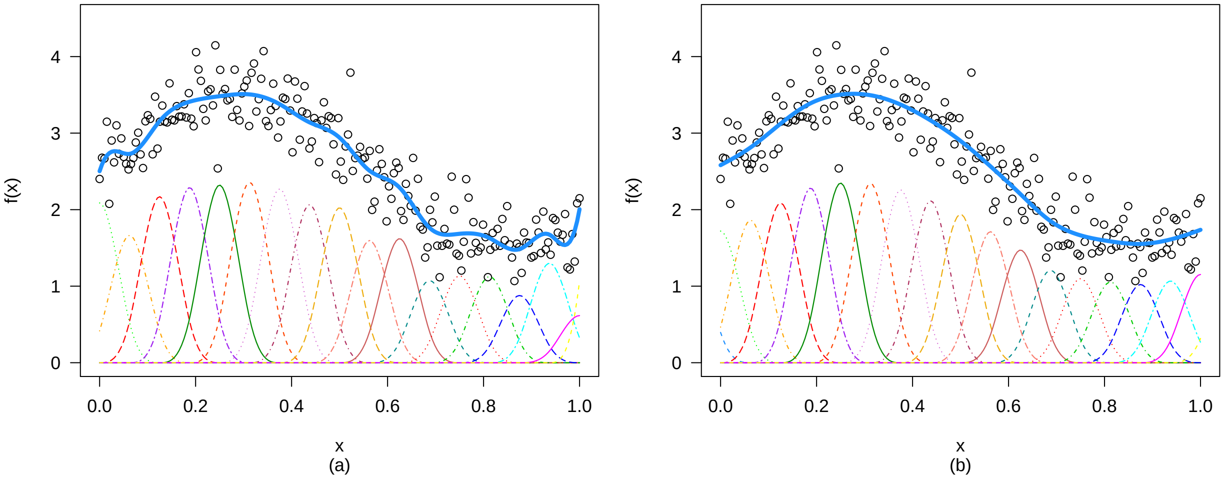


Figure 2.5: (a): Fitting 16 B-splines of degree 3 to the model (2.19). (b): A wiggliness penalty based on the *discrete* approximate wiggliness measurement by Eilers and Marx (1996) is added to the model fitting objective.

fitting. When a wiggliness penalty is added to the model fitting objective, a reasonable smooth curve is obtained as shown in Fig. 2.5 (b).

Eilers and Marx (1996) proposed using a *discrete* approximate wiggliness penalty based on (higher order) finite differences of the coefficients of adjacent B-splines to tune the amount of smoothness instead of using the integral of squared higher derivative of the fitted curve in the penalty. They used the difference penalty on neighbor B-spline coefficients to ensure that neighboring coefficients do not differ too much from each other. This results in a smoother curve fit. In addition they also showed that there is a strong connection between a penalty on a second-order differences of the B-spline coefficients and a penalty on the second derivative of the fitted function given by O'Sullivan (1986). Their approach reduced the dimensionality of the derivative based-penalty problem to the number of B-splines instead of the number of observations as in smoothing splines and offered computational advantages as the penalty is based on difference formulas. But the discrete penalties are somewhat difficult to interpret in terms of function shape than the traditional derivative based spline penalties (cf. Wood, 2016).

Adding the wiggliness measure of Eilers and Marx (1996) to (2.18), the P-spline penalized least squares objective for the model can be expressed as

$$S \;=\; \sum_{i=1}^{n} \left\{ y_i - \sum_{s=1}^{\mathrm{S}} a_s B_s(x_i) \right\}^2 + \lambda \sum_{s=1}^{\mathrm{S}} \left(\Delta^{[d]}\, a_s \right)^2 . \tag{2.20}$$

Here, λ is a non-negative smoothing parameter controlling the trade off between model fit and model smoothness. The function's wiggliness is measured by $\sum_{s=1}^{\mathrm{S}} \left(\Delta^{[d]}\, a_s \right)^2$, where $\Delta^{[d]}$ is the d^{th} order difference of a_s defined as $\Delta^{[d]} a_s \;=\; \sum_{t=0}^{d} (-1)^t \binom{d}{t} a_{s-t}$, where $s \in n$, and $s \;=\; 2, 3, \ldots, \mathrm{S}$. Given the definition of the operator, $\Delta^{[d]} a_s$, the first difference of a_s, $\Delta^{[1]} a_s$ is $a_s - a_{s-1}$. Higher-order differences of a_s can be then obtained by using the same computation, e.g., $\Delta^{[2]} a_s \;=\; \Delta a_s - \Delta a_{s-1} \;=\; a_s - 2a_{s-1} + a_{s-2}$, and $\Delta^{[3]} a_s \;=\; a_s - 3a_{s-1} + 3a_{s-2} - a_{s-3}$.

The wiggliness penalty, $\sum_{s=1}^{\mathrm{S}} \left(\Delta^{[d]}\, a_s \right)^2$, can be represented in a computationally convenient matrix-vector form. Consider first the wiggliness measure using the first order differences of a_s,

$$\sum_{s=1}^{\mathrm{S}} \left(\Delta^{[1]}\, a_s \right)^2 \;=\; \sum_{s=1}^{\mathrm{S}} \left(a_s - a_{s-1} \right)^2 .$$

We can write the operator $\Delta^{[1]}\, a_s$ as

$$\begin{pmatrix} -a_1 + a_2 \\ -a_2 + a_3 \\ \vdots \\ -a_{\mathrm{S}-1} + a_{\mathrm{S}} \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix} \boldsymbol{a} \;=\; \mathbf{D}_{[1]}\, \boldsymbol{a},$$

where $\boldsymbol{a} \;=\; (a_1, \ldots, a_{\mathrm{S}})$. The penalty can be then obtained in the simple form of

$$\sum_{s=1}^{\mathrm{S}} \left(\Delta^{[1]}\, a_s \right)^2 \;=\; \boldsymbol{a}^T\, \mathbf{D}_{[1]}^T\, \mathbf{D}_{[1]}\, \boldsymbol{a}.$$

Equivalent matrices $\mathbf{D}_{[d]}$ for second and third-order differences can be written as

$$\mathbf{D}_{[2]} \;=\; \begin{pmatrix} 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & & & \vdots & & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2 & 1 \end{pmatrix},$$

$$\mathbf{D}_{[3]} \;=\; \begin{pmatrix} -1 & 3 & -3 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & -3 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 3 & -3 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & & & & & \vdots & & \vdots & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 3 & -3 & 1 \end{pmatrix}$$

respectively. More generally, the matrix, $\mathbf{D}_{[d]}$ constructs d^{th} order difference of the coefficients, $\mathbf{D}_{[d]}\,\boldsymbol{a} \;=\; \Delta^{[d]}\,\boldsymbol{a}$. By the definition of the matrix $\mathbf{D}_{[d]}$, the penalty $\sum_{s=1}^{\mathrm{S}}\left(\Delta^{[d]}\,a_s\right)^2$ can be written as a quadratic form in the parameter vector $\boldsymbol{a}$ as

$$\sum_{s=1}^{\mathrm{S}}\left(\Delta^{[d]}\,a_s\right)^2 \;=\; \boldsymbol{a}^T\,\mathbf{D}_{[d]}^T\,\mathbf{D}_{[d]}\,\boldsymbol{a} \tag{2.21}$$

$$\;=\; \boldsymbol{a}^T\,\mathbf{P}_{[d]}\,\boldsymbol{a}, \tag{2.22}$$

where $\mathbf{P}_{[d]} \;=\; \mathbf{D}_{[d]}^T\,\mathbf{D}_{[d]}$. This enables the P-spline penalized-least-squares objective (2.20) to be re-written in a general matrix-vector form as

$$S(\boldsymbol{a}) \;=\; (\boldsymbol{y} - \mathbf{B}\,\boldsymbol{a})^T\,(\boldsymbol{y} - \mathbf{B}\,\boldsymbol{a}) + \lambda\,\boldsymbol{a}^T\,\mathbf{P}_{[d]}\,\boldsymbol{a}, \tag{2.23}$$

where $\boldsymbol{y}^T \;=\; (y_1, \ldots, y_n)$, $\boldsymbol{x}^T = (x_1, \ldots, x_n)$, and $\boldsymbol{a}^T = (a_1, \ldots, a_{\mathrm{S}})$ is the vector of coefficients. We define $\mathbf{B}\,(x_i) \;=\; (B_1(x_i),\, B_2(x_i),\, \ldots,\, B_{\mathrm{S}}(x_i))^T$ as a vector containing each B-spline evaluated at the values of x_i and $\mathbf{B}$ is a B-spline matrix of dimension $n \times \mathrm{S}$ containing the vectors $\mathbf{B}\,(x_i)$ as its columns. Given the smoothing parameter λ, the penalized least squares estimator of $\boldsymbol{a}$ is

$$\widehat{\boldsymbol{a}} \;=\; \left(\mathbf{B}^T\mathbf{B} + \lambda\,\mathbf{P}_{[d]}\right)^{-1}\,\mathbf{B}^T\,\boldsymbol{y}.$$

Then, the influence, or hat matrix, $\mathbf{A}$ for the model can be written

$$\mathbf{A} \;=\; \mathbf{B}\left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{P}_{[d]}\right)^{-1}\mathbf{B}^T.$$

It is not difficult to generalize to more than two smooth components. Consider modeling data $(y_i, x_{i1}, \ldots, x_{ip})$ using

$$y_i \;=\; \beta_1 x_{i1} + f_2\left(x_{i2}\right) + \cdots + f_p\left(x_{ip}\right) + \varepsilon_i, \quad y_i \sim N\left(\mu_i, \sigma^2\right), \tag{2.24}$$

where $x_{i1} \;=\; 1$ denote the intercept, and $\boldsymbol{\varepsilon_i} \;\sim\; \mathrm{N}\left(0, \sigma^2\right)$ independently. This model is known as an additive model (AM). Here, the f_k are smooth functions constructed using B-splines. Given the smoothing parameters λ_k, the model can be estimated by penalized least squares in the same way as described above for the simple univariate model. In Marx and Eilers (1998), the smoothing parameters are selected by cross validation (CV).

The fitting objective for this model is given by

$$\text{minimize} \quad \sum_{i=1}^{n}\left\{y_i - \beta_1 - \sum_{k=2}^{p}\sum_{s=1}^{\mathrm{S}_k} a_{sk}B_{sk}(x_{ik})\right\}^2 + \sum_{k=2}^{p}\sum_{s=1}^{\mathrm{S}} \lambda_k\left(\Delta^{[d]}\, a_{sk}\right)^2. \tag{2.25}$$

Using matrix-vector notation, this AM fitting problem becomes

$$\text{minimize} \quad \left(\boldsymbol{y} - \mathbf{B}\,\boldsymbol{a}\right)^T\left(\boldsymbol{y} - \mathbf{B}\,\boldsymbol{a}\right) + \boldsymbol{a}^T\,\mathbf{P}\,\boldsymbol{a}, \tag{2.26}$$

with respect to $\boldsymbol{a}$. Here, we define $\boldsymbol{a} \;=\; \left(b_1^T,\, \boldsymbol{a}_2^T,\, \ldots,\, \boldsymbol{a}_p^T\right)^T$, $\boldsymbol{y} \;=\; \left(\boldsymbol{y}_1^T, \ldots, \boldsymbol{y}_p^T\right)^T$, and the matrix $\mathbf{B} \;=\; \left(\mathbf{1}\,|\,\mathbf{B}_2\,|\,\ldots\,|\,\mathbf{B}_p\right)$ is the model matrix of dimension $n \times \left(1 + \sum_{k=2}^{p} \mathrm{S}_k\right)$, where $\mathbf{B}_k$ is the B-spline matrix generated from the values of x_{ik}. The matrix $\mathbf{P} \;=\; \text{blockdiag}\left(0,\, \lambda_2\mathbf{P}_{[d]2},\, \ldots,\, \lambda_p\mathbf{P}_{[d]p}\right)$ is a block diagonal matrix of the penalties of the model. The zero appearing on the first diagonal element of the matrix $\mathbf{P}$ is related to the intercept term, and the remaining block diagonal elements represent the values of $\lambda_k\,\mathbf{P}_{[d]k}$, where $\mathbf{P}_{[d]k} \;=\; \mathbf{D}_{[d]k}^T\mathbf{D}_{[d]k}$. Given values for λ_k, the penalized least squares estimator of $\boldsymbol{a}$ is

$$\widehat{\boldsymbol{a}} \;=\; \left(\mathbf{B}^T\mathbf{B} + \mathbf{P}\right)^{-1}\mathbf{B}^T\,\boldsymbol{y}.$$

In the next section, we will show how GAMs can be represented using the P-spline approach, and estimated by penalized regression methods.

2.3.2 P-splines GAM likelihood and estimation

As described in section 2.2, GAMs replace the linear predictor in a GLM by a sum of smooth functions of the covariate variables, assuming that the response follows an exponential family distribution. The general form of GAMs is represented by (2.4). In order to generalize the P-spline method to GAMs, Marx and Eilers (1998) introduced a linear predictor in the form of

$$g\,(\boldsymbol{\mu}) \;=\; \boldsymbol{\eta} \;=\; \mathbf{B}\,\boldsymbol{a},$$

where $y_i \sim$ exponential family and $\boldsymbol{a}^T \;=\; (b_1,\, \boldsymbol{a}_2,\, \ldots,\, \boldsymbol{a}_p)$. The matrix $\mathbf{B}$ is the model matrix consisting of the B-spline matrix constructed in the same way as the model matrix of an AM as explained in the discussion following (2.24). To ensure smooth function estimates, a wiggliness measure based on the differences of coefficients of adjacent B-splines is given for each smooth function as before. Therefore, the penalized log-likelihood version of fitting P-splines for each GAM component is expressed as

$$\ell_p^* \;=\; \ell\,(\boldsymbol{y};\boldsymbol{a}) - \frac{1}{2}\sum_{k=2}^{p}\lambda_k \boldsymbol{a}_k^T P_{[d]k}\boldsymbol{a}_k, \tag{2.27}$$

where the log-likelihood function, $\ell\,(\boldsymbol{y};\boldsymbol{a})$ is given by

$$\ell(\boldsymbol{y};\boldsymbol{a}) \;=\; \sum_{i=1}^{n}\left[\frac{y_i\,\theta_i - b(\theta_i)}{a(\phi,\omega)} + c(y_i,\phi)\right],$$

and the λ_k's are the smoothing parameters. The natural parameter θ_i is determined by μ_i through $\mathrm{E}(Y_i) \;=\; \mu_i \;=\; b'(\theta_i)$, providing $\theta_i \;=\; (b')^{-1}(\mu_i) \;=\; g(\mu_i) \;=\; [\mathbf{B}\boldsymbol{a}]_i$, where $[\cdot]_i$ denotes the ith element of $\mathbf{B}\boldsymbol{a}$. The structure of the block diagonal matrix $\mathbf{P}$ for GAMs is the same as for AMs (cf. equation (2.26)), that is, $\mathbf{P} \;=\; \mathrm{blockdiag}\left(0,\, \lambda_2\mathbf{P}_{[d]2},\, \ldots,\, \lambda_p\mathbf{P}_{[d]p}\right)$, where $\mathbf{P}_{[d]k} \;=\; \mathbf{D}_{[d]k}^T\mathbf{D}_{[d]k}$. Marx and Eilers (1998) suggested that users assign the different order-difference, d, to each $\mathbf{P}_{[d]k}$, and that in practice, any order up to three is considered adequate. The P-spline penalized log-likelihood (2.27) can then be re-written compactly as

$$\ell_p^* \;=\; \ell\,(\boldsymbol{y};\boldsymbol{a}) - \frac{1}{2}\boldsymbol{a}^T\mathbf{P}\boldsymbol{a}. \tag{2.28}$$

Given values for λ_k, when ℓ_p^* is maximized with respect to $\widehat{\boldsymbol{a}}$ using penalized Fisher scoring, the current estimate of $\boldsymbol{a}$ at the tth iteration is updated using

$$\boldsymbol{a}^{(t+1)} = \boldsymbol{a}^{(t)} + \boldsymbol{\mathcal{I}}_E\left(\boldsymbol{a}^{(t)}\right)^{-1}\boldsymbol{U}\left(\boldsymbol{a}^{(t)}\right), \tag{2.29}$$

where

$$\begin{aligned} \boldsymbol{U}(\boldsymbol{a}) &= \frac{\partial \ell_p^*}{\partial \boldsymbol{a}} \\ \text{and} \qquad \boldsymbol{\mathcal{I}}_E(\boldsymbol{a}) &= E\left(-\frac{\partial^2 \ell_p^*}{\partial \boldsymbol{a}\,\partial \boldsymbol{a}^T}\right). \end{aligned}$$

Here, all matrices and vectors are evaluated at the value of $\boldsymbol{a}$ at the current iteration. Using the chain rule of differentiation, $\boldsymbol{U}(\boldsymbol{a})$ and $\boldsymbol{\mathcal{I}}_E(\boldsymbol{a})$ can be written as follows:

$$\begin{aligned} \boldsymbol{U}(\boldsymbol{a}) &= \sum_{i=1}^{n} \frac{(y_i-\mu_i)}{\mathrm{var}(Y_i)}\frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i}\,\mathbf{B}_i - \mathbf{P}\,\boldsymbol{a}, \\ \boldsymbol{\mathcal{I}}_E(\boldsymbol{a}) &= \sum_{i=1}^{n} \frac{1}{\mathrm{var}(Y_i)}\left(\frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i}\right)^2 \mathbf{B}_i\,\mathbf{B}_i^T + \mathbf{P}, \end{aligned} \tag{2.30}$$

and the Fisher weight matrix is defined as

$$\mathbf{W} = \mathrm{diag}\left\{\frac{1}{\mathrm{var}\,(Y_i)}\left(\frac{\mathrm{d}\mu_i}{\mathrm{d}\eta_i}\right)^2\right\},$$

where w_{ii} is the ith diagonal elements of $\mathbf{W}$. Thus, $\boldsymbol{\mathcal{I}}_E(\boldsymbol{a})$ can be written in matrix-vector form as

$$\boldsymbol{\mathcal{I}}_E(\boldsymbol{a}) = \mathbf{B}^T\mathbf{W}\mathbf{B} + \mathbf{P}. \tag{2.31}$$

Similarly, $\boldsymbol{U}(\boldsymbol{a})$ satisfies the equations

$$\begin{aligned} \boldsymbol{U}(\boldsymbol{a}) &= \left\{\sum_{i=1}^{n} w_{ii}\ (y_i-\mu_i)\frac{\mathrm{d}\eta_i}{\mathrm{d}\mu_i}\,\mathbf{B}_i\right\} - \mathbf{P}\,\boldsymbol{a}, \\ &= \mathbf{B}^T\mathbf{W}\left\{(y_i-\mu_i)\,\frac{\mathrm{d}\eta_i}{\mathrm{d}\mu_i}\right\} - \mathbf{P}\,\boldsymbol{a}. \end{aligned} \tag{2.32}$$

Substituting (2.32) and (2.31) into (2.29), the update becomes

$$\boldsymbol{a}^{(t+1)} = \left(\mathbf{B}^T\,\mathbf{W}^{(t)}\,\mathbf{B} + \mathbf{P}\right)^{-1}\mathbf{B}^T\,\mathbf{W}^{(t)}\,\boldsymbol{z}^{(t)}, \tag{2.33}$$

where the “adjusted dependent vector”, $\boldsymbol{z}$, has the ith element

$$\boldsymbol{z}_i \;=\; \eta_i + (y_i - \mu_i)\left(\frac{\mathrm{d}\eta_i}{\mathrm{d}\mu_i}\right).$$

The term η_i is the ith element of $\mathbf{B}\,\boldsymbol{a} \;=\; \boldsymbol{\eta}$. Since P-spline GAMs use a penalized version of the Fisher scoring algorithm, then for given values of λ_k, the maximization of ℓ_p^* in (2.28) with respect to $\boldsymbol{a}$ can be achieved by iteratively using (2.33). In P-IRLS, Marx and Eilers (1998) fitted a generalized weighted linear model using the data augmentation on a response, regressors, and weights until the convergence is achieved, where the augmented response, regressors, and weights are given by

$$\boldsymbol{y}^* \;=\; \begin{pmatrix} \boldsymbol{y} \\ \mathbf{0} \end{pmatrix}, \qquad \mathbf{B}^* \;=\; \begin{pmatrix} \mathbf{B} \\ \mathbf{D}^* \end{pmatrix}, \qquad \mathbf{W}^* \;=\; \begin{pmatrix} \mathbf{W} & 0 \\ 0 & \mathbf{I}^* \end{pmatrix}.$$

Here, $\mathbf{0}$ is a vector of dimension $1 + \sum_{k=2}^{p}(\mathrm{S}_k - d_k)$ and $\mathbf{I}^*$ is the $\{1 + \sum_{k=2}^{p}(\mathrm{S}_k - d_k)\} \times \{1 + \sum_{k=2}^{p}(\mathrm{S}_k - d_k)\}$ identity matrix. The matrix $\mathbf{D}^*$ is defined as follows

$$\mathbf{D}^* \;=\; \mathrm{blockdiag}\left(0, \sqrt{\lambda_1}\,\mathbf{D}_{[d]2}, \ldots, \sqrt{\lambda_p}\,\mathbf{D}_{[d]p}\right).$$

Smoothing parameter selection

In a P-spline approach for GAMs, an optimal value for the smoothing parameters λ_k can be selected through the minimization of an information criterion (IC), the generalized cross-validation (GCV) score, or some other criteria, depending on whether the scale parameter is known. If the scale parameter is known, the minimization of an IC, e.g., Akaike information criterion (AIC), unbiased risk estimator (UBRE), or the Bayesian information criteria (BIC) is preferred. Thus, with non-normal responses, such as Poisson count or binary data, the smoothing parameters $\boldsymbol{\lambda}$ are estimated by minimizing AIC, where the AIC for the P-spline GAMs (Eilers and Marx, 1996; Marx and Eilers, 1998) is equivalent to

$$\mathrm{AIC}\,(\boldsymbol{\lambda}) \;=\; Dev + 2\,\mathrm{tr}(\mathbf{A}).$$

On the other hand, if the scale parameter is unknown, and has to be estimated, such as for normal, gamma, and negative binomial responses, the smoothing parameters $\boldsymbol{\lambda}$ are obtained from the minimization of an approximation to the GCV objective function (Hastie and Tibshirani, 1990),

$$\mathrm{GCV}(\boldsymbol{\lambda}) \;=\; \frac{nDev}{[n - \mathrm{tr}(\mathbf{A})]^2}.$$

Here, $\boldsymbol{\lambda}$ is a vector of smoothing parameters, and Dev is the deviance of the fitted model. The smoother matrix $\mathbf{A}$ is given by $\mathbf{B}\,\left(\mathbf{B}^T\,\mathbf{W}\,\mathbf{B} + \mathbf{P}\right)^{-1}\,\mathbf{B}^T\,\mathbf{W}$, where $\mathbf{W}$ contains the weights, generated at convergence. The term $\mathrm{tr}(\mathbf{A})$ is the estimated degrees of freedom of the model. The regression diagnostics and model checking for the P-spline GAMs will be similar to what is done for the GLMs. The model including the parametric components can be directly estimated by the P-spline GAMs.

2.3.3 An example

The data on daily air pollution and the daily death rate in Chicago, USA, from Peng and Welty (2004) is used to illustrate the P-spline approach. This data was presented by Wood (2006b) as an application of generalized additive modeling. The specific outcome of interest is the daily death rate in Chicago (`death`), and the available predictors are the air quality measured by levels of ozone (`o3median`), levels of particulates such as diesel exhaust (`pm10median`), level of sulphur dioxide (`so2median`), and mean daily temperature in degrees Fahrenheit (`tmpd`). The data are contained in a data frame called `chicago` from `gamair`. For homogeneity, the missing values are removed from the dataset beforehand. All computations were performed using the `mgcv` package, with P-splines being used as smooth terms within the `gam` model formulas.

Since the purpose of this illustration is to provide the basic application of P-splines to GAMs, and show that all the smooth components are estimated simultaneously with the P-spline approach, the details of model development will not be discussed here. We are interested in the relationship between the air pollution and the daily death rate. In order to investigate this relationship, we model the daily death rates as a smooth function of the four predictors. Since the

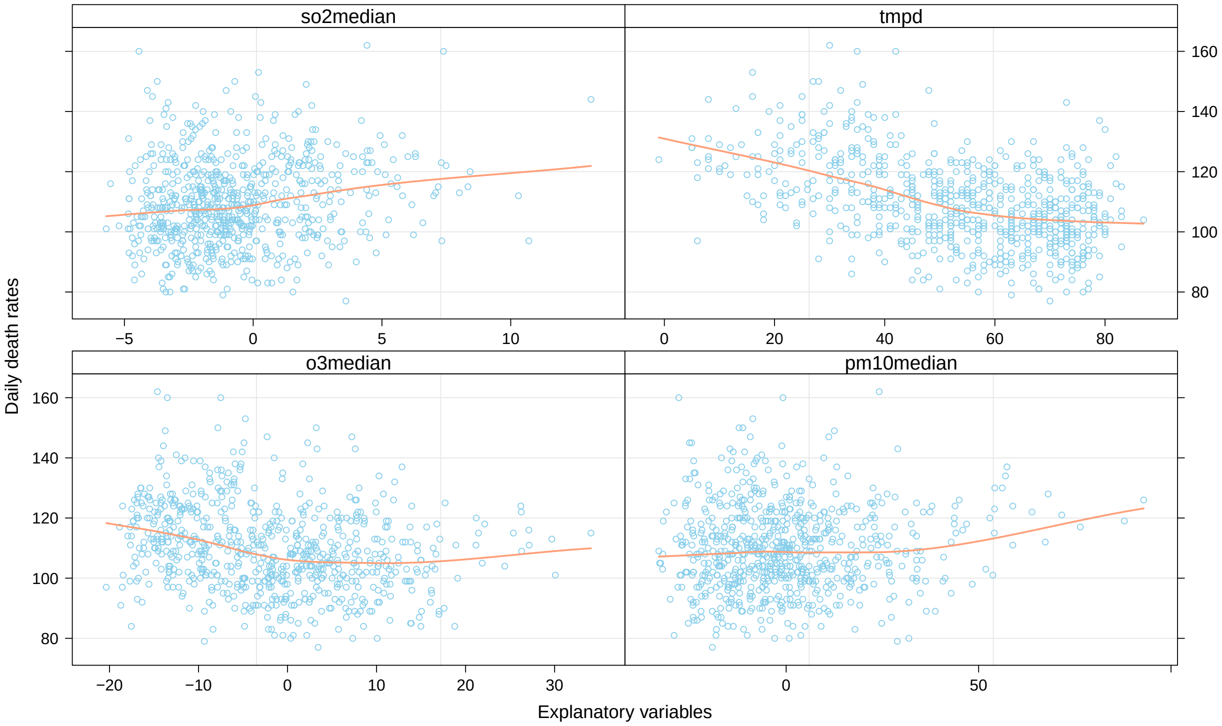


Figure 2.6: Scatterplots with loess smoother relating the outcome variable, `death` to each of the four predictors, `o3median`, `pm10median`, `so2median`, and `tmpd` respectively.

response variable is a count, we will use an additive Poisson model with the log link. Fig. 2.6 shows the scatterplots of pairs death rate with each of the four predictors. We added a loess smoother to the pairwise scatterplots to summarize trend. Fig. 2.6 indicates that there is a linear trend between the sulphur dioxide levels and the daily death rate while the daily temperature, the levels of ozone, and the particulates levels provide a nonlinear trend to the outcome.

Next, we fit an additive Poisson model to the count response, `death`, with smooth terms for the air quality covariates. All the smooth terms are fitted using the P-spline approach with the default settings for cubic B-splines, and the penalty based on the second order differences in the coefficients. The smoothness selection for these models has been performed using the UBRE score. The default model-checking plots for this model shown in Fig. 2.7 indicate that the distribution assumption is reasonable, and the variance is approximately constant.

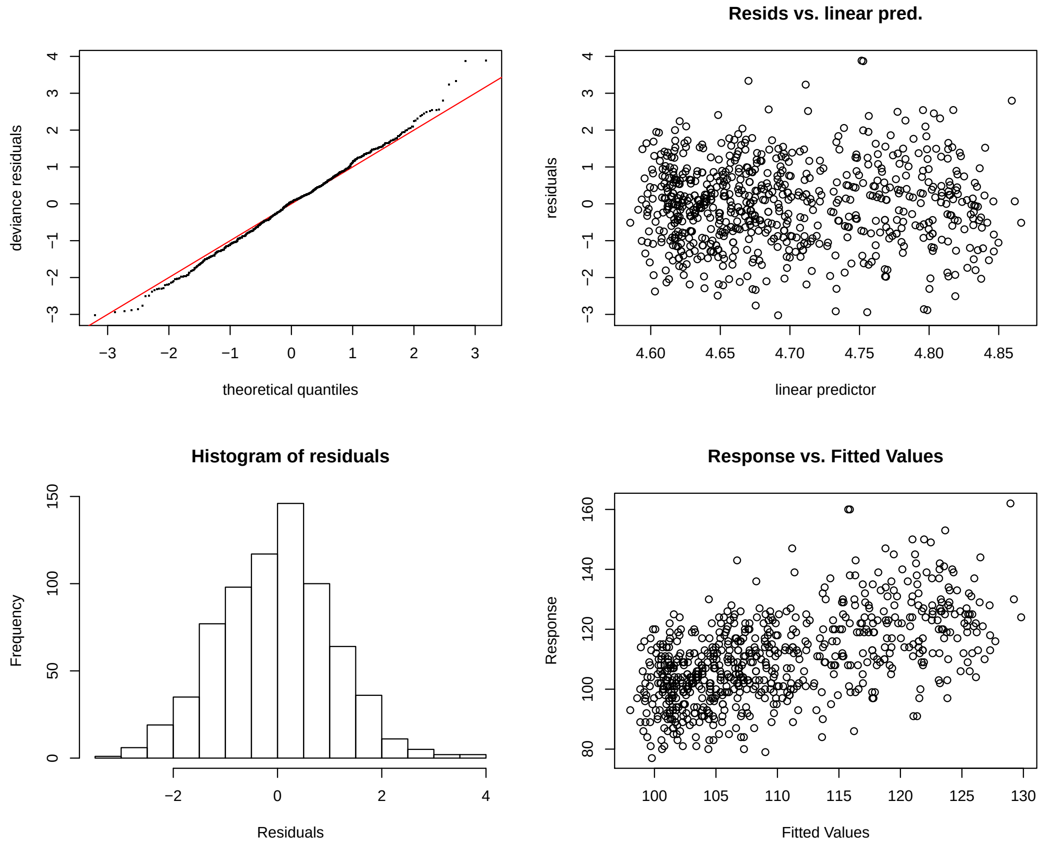


Figure 2.7: Some basic model checking plots for the air pollution mortality models.

The resulting curves for all predictor variables are shown in Fig. 2.8. The fitted curves of the three predictors `o3median`, `pm10median`, and `tmpd` have some interesting non-linear features, while the estimated curve for the effect of sulphur dioxide appears linear. The EDF estimates for the predictors `o3median`, `pm10median`, `so2median`, and `tmpd` are 2.75, 1.87, 1, and 4.64 respectively. The three EDF estimates for the predictors `o3median`, `pm10median`, and `tmpd` suggest the presence of non-linearity, whereas the EDF estimate of `so2median` supports linearity as its EDF estimate is equal to 1. The estimated smooths show that the mean mortality is moderately stable, when the ozone levels were approximately between 0 to 10 part per billion. There is some suggestion that the mean death rates go up when the ozone levels rise above about 10 part per billion. However, there is very little data in this region and the confidence intervals are wide. There is a strong positive non-linear relationship between the pollution levels produced

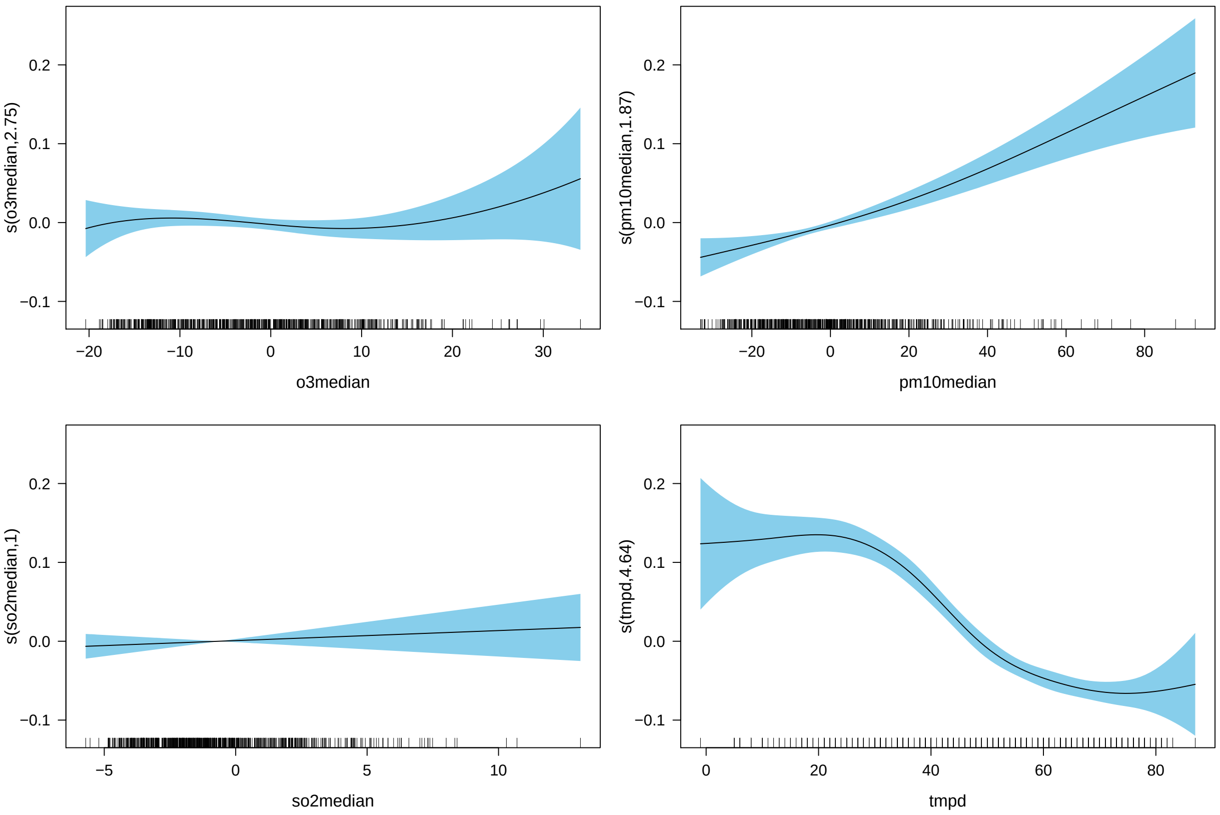


Figure 2.8: Fitted functions for `o3median`, `pm10median`, `so2median`, and `tmpd` using the P-spline approach to the air pollution in Chicago dataset. The shaded region represents twice the pointwise standard errors of the estimated curve.

from the diesel exhaust (`pm10median`) and the number of deaths. Therefore, as the pollution levels increase, there is a tendency for the mean mortality to also increase. For the temperature variable, the estimated smooth curves show that the mean daily death rates are higher with lower temperatures. More details of the model development for the Chicago air pollution mortality data can be found from Wood (2006b, p.247).

2.4 Conclusions

The aim of this chapter is to provide the important ideas of a recent development of GAMs based on the approach of Wood (2000, 2004, 2006b, 2008, 2011). We have illustrated how GAMs can be generalized to penalized regression splines, and estimated by penalized likelihood maximization via penalized iterative least squares. We have discussed the basics of smoothing parameter estimation, and highlighted the effect of smoothing parameter choices on the shape of the estimated smooth functions. The kyphosis data example and the simulation study by Marra and Radice (2010) show that the choice of smoothing parameter plays a crucial role in determining the flexibility of models, and the estimated shape of the smooth functions. Automatic numerical procedures used to determine the shape of non-linear terms from the data is highly desirable for GAMs. Wood (2000, 2004, 2006b, 2008) succeeded in developing penalized likelihood-based methods and implementing an automatic procedure for stable multiple smoothing parameter selection for GAMs. His approaches resolve the issue of determining the shape of non-linear components and lead the GAM method based on the penalized likelihood approach has a major advantage over the backfitting approach.

We have also discussed the representation and estimation of the smooth functions using penalized regression splines based on the P-spline approach of Eilers and Marx (1996) and Marx and Eilers (1998), and illustrated the approach by modeling daily air pollution and daily death rate. P-splines come with many attractive properties. On the computation side, P-splines are very easy to construct and use. Standard errors and regression diagnostics can be easily obtained. All these numerous advantages make P-splines the preferred method method to estimate GAMs.

In subsequent chapters, we will take the ideas of penalized likelihood-based approach developed by Eilers and Marx (1996), Marx and Eilers (1998), and Wood (2000, 2004, 2006b, 2008, 2011) and extend them to vector GAMs.

Chapter 3

VGLMs/VGAMs

Generalized linear models (GLMs) were formulated by Nelder and Wedderburn (1972), and its nonparametric extension known as generalized additive models (GAMs) were introduced by Hastie and Tibshirani (1990). These models were confined to data where the distribution of the response variable comes from the exponential family of distributions. To extend the GLM and GAM classes, Yee and Wild (1996) developed the class of vector generalized linear models (VGLMs) and vector generalized additive models (VGAMs). VGLMs model each of a set of parameters as a linear combination of the covariate variables $\boldsymbol{x}$, specified in terms of a parametric class, while VGAMs are VGLMs with linear predictors replaced by a sum of smooth functions of the covariates. Thus, VGAMs specify the model in terms of 'smooth functions' rather than parametric relationships. The VGLM and VGAM classes extended the model settings of GLMs and GAMs to include (i) responses $\boldsymbol{y}$, with distribution that are not restricted to the exponential family, (ii) multivariate or multiple responses $\boldsymbol{y}$, and multivariate or multiple linear or additive predictors $\boldsymbol{\eta}$, and (iii) linear or additive predictors $\boldsymbol{\eta}$ that are not necessarily functions of a mean. VGLMs/VGAMs (which include GLMs/GAMs) were developed to encompass and unify as many distributions and models as possible. The conditional distribution of the response is intended to be completely general. The currently implemented class of VGLMs and VGAMs is very large and includes many statistical distributions and models.

The theoretical basis of parameter estimation in the VGLMs is the method of maximum likelihood. Its implementation uses IRLS. VGAMs, on the other hand, are fitted by a combination of IRLS and modified vector backfitting using vector splines. An additional underlying idea that allows VGLMs/VGAMs to cover many distributions and models is provision for "constraints on the functions". The summary details of the VGLM/VGAM classes and the ideas of the constraints on the functions will be described in the next sections. The aim of this chapter is to give a brief and partial sketch of the key ideas underpinning the VGLMs/VGAMs framework. For further details about VGLMs and VGAMs can be found in Yee and Hastie (2003) and Yee and Wild (1996) respectively. The most complete reference on the theory and applications of the VGLM/VGAM classes is Yee (2015b).

3.1 VGLMs

Let us suppose that the observed response $\boldsymbol{y}$ is a q-dimensional vector $(q \geq 1)$, and the observed covariate $\boldsymbol{x} = (x_1, \ldots, x_p)^T$ is a p-dimensional vector, where $x_1 = 1$ denotes the intercept. Yee and Wild (1996) described VGLMs as a model of the conditional distribution of $\boldsymbol{Y}$ given $\boldsymbol{x}$ written by

$$f\left(\boldsymbol{y}|\boldsymbol{x};\mathbf{B}\right) = h\left(\boldsymbol{y}, \eta_1, \ldots, \eta_M\right), \tag{3.1}$$

where, $h\left(\cdot\right)$ is some known function, $\mathbf{B} = (\boldsymbol{\beta}_1\, \boldsymbol{\beta}_2\, \cdots\, \boldsymbol{\beta}_M)$ is a $p \times M$ matrix of unknown regression coefficients, and the jth linear predictor is given by

$$\eta_j = \eta_j(\boldsymbol{x}) = \sum_{k=1}^{p} \beta_{(j)k}\, x_k, \qquad j = 1, \ldots, M. \tag{3.2}$$

Let $\boldsymbol{x}_i$ be the vector of explanatory variables for the i^{th} observation, $i = 1, \ldots, n$. Then, the set of linear predictors for the i^{th} individual, $\boldsymbol{\eta}_i$, can be written as follows:

$$\boldsymbol{\eta}_i \;=\; \boldsymbol{\eta}\left(\boldsymbol{x}_i\right) \;=\; \begin{pmatrix} \eta_1\left(\boldsymbol{x}_i\right) \\ \vdots \\ \eta_M\left(\boldsymbol{x}_i\right) \end{pmatrix} \;=\; \mathbf{B}^T \boldsymbol{x}_i \tag{3.3}$$

$$= \begin{pmatrix} \beta_{(1)1} & \cdots & \beta_{(1)p} \\ \vdots & & \vdots \\ \beta_{(M)1} & \cdots & \beta_{(M)p} \end{pmatrix} \boldsymbol{x}_i. \tag{3.4}$$

The full vector of the model coefficients can be then written as

$$\boldsymbol{\beta} \;=\; \left(\boldsymbol{\beta}_{(1)}^T, \ldots, \boldsymbol{\beta}_{(p)}^T\right)^T, \tag{3.5}$$

where $\boldsymbol{\beta}_{(k)} \;=\; \left(\beta_{(1)k}, \ldots, \beta_{(M)k}\right)^T, \;\; k \;=\; 1, \ldots, p.$ Given (3.5), the log-likelihood of the model (3.1) can be written as

$$\ell\left(\boldsymbol{\beta}\right) \;=\; \sum_{i=1}^{n} w_i\, \ell_i\left\{\eta_1\left(\boldsymbol{x}_i\right), \ldots, \eta_M\left(\boldsymbol{x}_i\right)\right\}. \tag{3.6}$$

Here, $\eta_j \;=\; \eta_j\left(\boldsymbol{x}\right) \;=\; \boldsymbol{\beta}_j^T \boldsymbol{x}_i,$ and w_i are known positive prior weights which allow for a minor generalization of (3.1). The Newton-Raphson algorithm for maximizing the log-likelihood (3.6) to estimate $\boldsymbol{\beta}$ is then given by

$$\boldsymbol{\beta}^{(t+1)} \;=\; \boldsymbol{\beta}^{(t)} + \boldsymbol{\mathcal{I}}\left(\boldsymbol{\beta}^{(t)}\right)^{-1} \boldsymbol{U}\left(\boldsymbol{\beta}^{(t)}\right), \tag{3.7}$$

where $\boldsymbol{U}\left(\boldsymbol{\beta}\right)$ is the score vector for the model and $\boldsymbol{\mathcal{I}}\left(\boldsymbol{\beta}\right)$ is the observed information matrix written as

$$\begin{aligned} \boldsymbol{U}\left(\boldsymbol{\beta}\right) \;&=\; \frac{\partial \ell\left(\boldsymbol{\beta}\right)}{\partial \boldsymbol{\beta}}, \\ \boldsymbol{\mathcal{I}}\left(\boldsymbol{\beta}\right) \;&=\; -\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}\, \partial \boldsymbol{\beta}^T}. \end{aligned}$$

The VGAM package uses the Fisher-scoring modification of Newton-Raphson which replaces $\boldsymbol{\mathcal{I}}\left(\boldsymbol{\beta}\right)$ by its expectation which ensures that all the individual working weight matrices are positive-definite. The result is an IRLS algorithm of the form

$$\boldsymbol{\beta}^{(t+1)} \;=\; \left(\sum_{i=1}^{n} \mathbf{X}_i^T\, \mathbf{W}_i^{(t)}\, \mathbf{X}_i\right)^{-1} \left(\sum_{i=1}^{n} \mathbf{X}_i^T\, \mathbf{W}_i^{(t)}\, \boldsymbol{z}_i^{(t)}\right). \tag{3.8}$$

Here, $\mathbf{X}_i$ is a $M \times Mp$ block-diagonal matrix where each main diagonal block is a copy of $\boldsymbol{x}_i^T$, $\mathbf{W}_i$ is a $M \times M$ matrix with (j,k)th element given by

$$(\mathbf{W}_i)_{jk} \;=\; -w_i\, E\left(\frac{\partial^2 \ell_i}{\partial \eta_j\, \partial \eta_k}\right), \tag{3.9}$$

the "adjusted dependent vector" is given by

$$\boldsymbol{z}_i^{(t)} \;=\; \boldsymbol{\eta}_i^{(t)} + \left(\mathbf{W}_i^{(t)}\right)^{-1} \boldsymbol{u}_i^{(t)},$$

the score vector $\boldsymbol{u}_i$ has j element

$$(\boldsymbol{u}_i)_j \;=\; w_i\, \frac{\partial \ell_i}{\partial \eta_j},$$

and $\boldsymbol{\eta}_i^{(t)} = \mathbf{X}_i\, \boldsymbol{\beta}^{(t)}$. The Newton-Raphson algorithm is rarely used as the observed information matrix (OIM) tends only to be positive-definite over a smaller portion of the parameter space. If Newton-Raphson is used, then the working weights matrices are given by

$$(\mathbf{W}_i)_{jk} \;=\; -w_i\, \frac{\partial^2 \ell_i}{\partial \eta_j\, \partial \eta_k}.$$

As explained in Yee and Wild (1996), $\boldsymbol{\beta}^{(t+1)}$ in (3.8) is the solution to the generalized least squares (GLS) problem

$$\boldsymbol{z}^{(t)} \;=\; \mathbf{X}_{\text{VLM}}\, \boldsymbol{\beta}^{(t+1)} + \boldsymbol{\varepsilon}^{(t)}. \tag{3.10}$$

Here, they defined $\boldsymbol{z} = \left(\boldsymbol{z}_1^T, \ldots, \boldsymbol{z}_p^T\right)^T$, $\mathbf{X}_{\text{VLM}} = \mathbf{X}_{\text{LM}} \otimes \mathbf{I}_{\text{M}} = \left(\mathbf{X}_1^T, \ldots, \mathbf{X}_n^T\right)^T$, where $\mathbf{X}_{\text{LM}}$ is the "linear model" model matrix generally used for the small model matrix for one η_j, and $\text{Var}(\boldsymbol{\varepsilon}) = \text{blockdiag}\left(\mathbf{W}_1^{-1}, \ldots, \mathbf{W}_n^{-1}\right)$. In VGLMs, $\mathbf{X}_{\text{VLM}}$ is called the "vector linear model" model matrix. For practical computation, Yee and Wild (1996) converted the GLS system of equations of (3.10) to ordinary least squares (OLS). To achieve this, they first constructed the "square root" of $\mathbf{W}_i$ from a Cholesky decomposition given by

$$\mathbf{W} \;=\; \mathbf{U}^T\, \mathbf{U} \;=\; \text{blockdiag}\left(\mathbf{U}_1^T\, \mathbf{U}_1, \ldots, \mathbf{U}_n^T\, \mathbf{U}_n\right),$$

and then premultiplied (3.10) by $\mathbf{U}^{(t)}$. The system of equations in (3.10) can be therefore written as

$$\boldsymbol{z}^{**(t)} \;=\; \mathbf{X}_{\text{VLM}}^{**}\, \boldsymbol{\beta}^{**(t+1)} + \boldsymbol{\varepsilon}^{**(t)}, \tag{3.11}$$

where $\mathrm{Var}(\boldsymbol{\varepsilon}^{**(t)}) = \sigma^2_{**}\,\mathbf{I}_{nM}$. The inversion of the working weight matrices involves evaluating the Cholesky decomposition $\mathbf{W}_i = \mathbf{U}_i^T\,\mathbf{U}_i$ to each working weight matrix, where $\mathbf{U}_i$ is upper-triangular. Since the working weight matrix is positive-definite, a Cholesky decomposition exits and is unique. Full details about computing the working weights are given in Yee (Section A.3.1, 2015b). In the IRLS scheme, a current $\boldsymbol{\eta}^{(t)}$ is computed, then the adjusted dependent vector $\boldsymbol{z}^{(t)}$ and the working weights $\mathbf{W}^{(t)}$ are calculated. The adjusted dependent vector $\boldsymbol{z}^{(t)}$ is then regressed against $\mathbf{X}_{\text{VLM}}$ with weights $\mathbf{W}^{(t)}$ to obtain $\boldsymbol{\beta}^{(t+1)}$, using (3.11) Then $\boldsymbol{\eta}^{(t+1)} = \mathbf{X}_{\text{VLM}}\,\boldsymbol{\beta}^{(t+1)}$ is evaluated. The process is repeated, till convergence is achieved.

In VGAMs, the expected information matrix (Fisher scoring) is much preferred over the observed information matrix (Newton-Raphson algorithm). This is because *all* working weight matrices $\mathbf{W}_i$ are required to be positive-definite over a larger region of the parameter space. As explained by Seber and Wild (Chapter 13, 2003), one reason for failure with the Newton-Raphson method is that although the Hessian is positive-definite at a local maximum in most applications, it may not be positive-definite at each iteration. This can cause the Newton method to fail. If the negative definite Hessian is encountered, the method may converge to the local maximum. Indeed, the Hessian at the tth iteration may be indefinite or even singular. The Fisher scoring method therefore uses the expected negative Hessian which is always positive-definite, so that the step taken at the tth iteration, $\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^{(t)}$, leads uphill. The Newton method however is usually faster compared to the Fisher scoring since the former will converge at a quadratic rate whereas the latter has a superlinear rate of convergence.

3.2 VGAMs

Yee and Wild (1996) introduced an extension of the class of GAMs to include multivariate regression models using vector smoothers. VGAMs are defined as an additive model extension of VGLMs. They did this by replacing the linear function of the covariates in (3.2) by a sum of smooth functions of the individual covariates, written as

$$\begin{aligned} \eta_j(\boldsymbol{x}) &= \sum_{k=1}^{p} f_{(j)k}(x_k), \qquad \text{or equivalently,} \\ \boldsymbol{\eta}(\boldsymbol{x}) &= \sum_{k=1}^{p} \boldsymbol{f}_k(x_k), \qquad j = 1, \ldots, M. \end{aligned} \tag{3.12}$$

Here, $\boldsymbol{x} = (x_1, \ldots, x_p)^T$, with $x_1 = 1$ if there is an intercept. In (3.12), there is no requirement for η_j to be a function of a mean. The term $\boldsymbol{f}_k$ is defined as

$$\boldsymbol{f}_k(x_k) = \left(f_{(1)k}(x_k), \ldots, f_{(M)k}(x_k)\right)^T. \tag{3.13}$$

Before describing some theoretical aspects of VGAM estimation, we need to discuss the basics of vector splines and the vector "measurement model" that are used as the building block for estimating VGAMs. Vector smoothing splines given in Yee and Wild (1996) were generalized from the idea of the classical smoothing splines as described by Hastie and Tibshirani (1990, chapter 2, 3). As explained in Yee and Wild (1996), implementation of VGAMs involved computational procedures for the vector smoothing problem. This problem involves a "measurement model" for a vector response $\boldsymbol{y}_i$, of dimension M at each value of $\boldsymbol{x}_i$ given by

$$\begin{gathered} \boldsymbol{y}_i = \boldsymbol{f}(x_i) + \boldsymbol{\varepsilon}_i, \qquad i = 1, \ldots, n, \\ \boldsymbol{\varepsilon}_i \sim (\boldsymbol{0}, \boldsymbol{\Sigma}_i) \ \text{ independently,} \end{gathered} \tag{3.14}$$

where $\boldsymbol{y}_i \in \mathbb{R}^M$, and $\boldsymbol{\Sigma}_i$ are known symmetric and positive-definite error covariances. Yee and Wild (1996) estimated the smooth vector function $\boldsymbol{f}(x)$, written $(f_1(x), \ldots, f_M(x))^T$ by minimizing the generalized least squares objective of (3.14), and the roughness penalty approach $\sum_j^M \lambda_j \int \{f_j''(x)\}^2 \, \mathrm{d}x$ of Green and Silverman (1993):

$$\sum_{i=1}^{n} \{\boldsymbol{y}_i - \boldsymbol{f}(x_i)\}^T \boldsymbol{\Sigma}_i^{-1} \{\boldsymbol{y}_i - \boldsymbol{f}(x_i)\} + \sum_{j=1}^{M} \lambda_j \int \{f_j''(x)\}^2 \, \mathrm{d}x. \tag{3.15}$$

The first term of (3.15) measures the lack of fit, while the second term is called the roughness penalty, which penalizes wiggliness. Each of the smoothing parameters $\lambda_1, \ldots, \lambda_M$ are assumed non-negative and fixed constants. In (3.15), vector splines are used to smooth vector values of

$\boldsymbol{y}_i$ against $\boldsymbol{x}_i$ using a weights matrix $\mathbf{W}_i$. The matrix $\mathbf{W}_i$ is defined as the inverse of the covariance matrix for $\boldsymbol{y}_i$. Fessler et al. (1991) gave an $O(nM^3)$ algorithm based on Reinsch (1967) for fitting (3.14) called VSPLINE. Yee and Wild (1996) then gave a modification of VSPLINE, called YEE-SPLINE, for fitting VGAMs more effectively.

Recall from Section 3.1 that models of the form (3.1) have the log-likelihood in the form of (3.6). This log-likelihood can be estimated using the IRLS algorithm and $\boldsymbol{\beta}^{(t+1)}$ results in the solution to the generalized least squares problem

$$\text{minimize} \quad \sum_{i=1}^{n} (\boldsymbol{z}_i - \mathbf{X}_i\boldsymbol{\beta})^T \, \mathbf{W}_i \, (\boldsymbol{z}_i - \mathbf{X}_i\boldsymbol{\beta}) \;=\; \sum_{i=1}^{n} \{\boldsymbol{z}_i - \boldsymbol{\eta}(\boldsymbol{x}_i)\}^T \mathbf{W}_i \{\boldsymbol{z}_i - \boldsymbol{\eta}(\boldsymbol{x}_i)\} \qquad (3.16)$$

with respect to $\boldsymbol{\beta}$, where $\boldsymbol{\eta}(\boldsymbol{x}_i)$ is an M-vector with j^{th} element $\eta_j(\boldsymbol{x}_i) = \boldsymbol{\beta}_j^T \boldsymbol{x}_i$. To estimate VGAMs, Yee and Wild (1996) updated $\boldsymbol{\eta}$ by minimizing the objective

$$\sum_{i=1}^{n} \{\boldsymbol{z}_i - \boldsymbol{\eta}\,(\boldsymbol{x}_i)\}^T \, \mathbf{W}_i \, \{\boldsymbol{z}_i - \boldsymbol{\eta}\,(\boldsymbol{x}_i)\} + \sum_{k=2}^{p} \sum_{j=1}^{M} \lambda_{(j)k} \int f''_{(j)k}(t)^2 \, \mathrm{d}t, \qquad (3.17)$$

where the integrated square of second derivative penalizes models that are too "wiggly" and $\eta_j\,(\boldsymbol{x}) = \sum_{k=1}^{p} f_{(j)k}(x_k), \quad j = 1, \ldots, M$.

For a single covariate, the minimization of (3.17) involves solving vector splines (3.15) identifying terms in (3.17) with $\mathbf{W}_i = \boldsymbol{\Sigma}_i^{-1}$. For multiple covariates, they applied a backfitting algorithm with vector smoothing instead of the usual y-scalar smoothing. This procedure, called the "vector backfitting algorithm", is used for fitting the vector additive model

$$E(\boldsymbol{y}_i) = \boldsymbol{\beta}_{(1)} + \sum_{j=1}^{p} \boldsymbol{f}_j(x_{ij})$$

to response vectors $\boldsymbol{y}_i$. They solved the problem of (3.17) for when there is more than one covariate by applying vector backfitting to the adjusted dependent vector $\boldsymbol{z}_i$. To avoid identifiability problems, they centered the intercept $\boldsymbol{\beta}_{(1)}$ and $\boldsymbol{f}_k$'s using a suitable constraint.

3.3 Constraints on the functions

In this section, we are going to discuss the imposition of constraints primarily as they apply to VGLMs. Recall from Section 3.1 that VGLMs have the form of

$$\eta_j \;=\; \eta_j(\boldsymbol{x}) \;=\; \sum_{k=1}^{p} \beta_{(j)k}\, x_k, \qquad j \;=\; 1, \ldots, M. \tag{3.18}$$

In (3.18) a covariate acts differently for each linear predictor. In practice, there are many occasions on which we may wish to constrain the effects of a single covariate to be the same for different η_j, or else constrain the covariate to have no effect for a given η_j. For example, we may wish to constrain the effects of a covariate x_2 on η_1 and η_2 to be the same as in

$$\begin{aligned} \eta_1 &= \beta_{(1)1} + \beta_{(1)2}\, x_2 + \beta_{(1)3}\, x_3 \\ \eta_2 &= \beta_{(2)1} + \beta_{(1)2}\, x_2 + \beta_{(2)3}\, x_3. \end{aligned} \tag{3.19}$$

We may also wish to specify one parameter to be independent of covariate x_2 as in

$$\begin{aligned} \eta_1 &= \beta_{(1)1} + \beta_{(1)2}\, x_2 \\ \eta_2 &= \beta_{(2)1}. \end{aligned} \tag{3.20}$$

Yee and Wild (1996) proposed a unified way of accommodating any constraints of the types described above by imposing constraints on the functions. In the “constraints on the functions” framework, “constraint matrices” are applied directly to the linear/additive predictors to control how the covariates act. These constraints allow VGLMs/VGAMs to cope with the special structures, e.g., parallelism, exchangeability between functions and functions that use only a subset of the covariates. They are very useful especially when we wish to model categorical data.

In the next subsection, two specific models from applied statistics: (i) negative binomial regression and (ii) bivariate logistic model that use the constraint ideas will be used to illustrate how the constraints on the functions can be accommodated within the VGLM framework.

3.3.1 Poisson and negative binomial regression

Modeling count variables is common in a wide range of settings, e.g., ecology, economics and the social sciences. The usual model for count data is Poisson regression with the probability distribution function given by

$$P\,(Y\,=\,y;\mu)\;=\;\frac{e^{-\mu}\,\mu^{y}}{y!},\qquad y\,=\,0,1,2,\ldots,\quad \mu>0,$$

and where the mean and variance functions of the Poisson distribution are identical: that is,

$$E(Y)\;=\;\mathrm{Var}(Y)\;=\;\mu(\boldsymbol{x}).$$

However, with real data it is common for the variance of the dependent variable to exceed the mean. This is called overdispersion. This problem may be modeled using a quasi-Poisson model or negative binomial regression. The negative binomial distribution has probability function

$$P(Y\,=\,y;\mu,k)\;=\;\binom{y+k-1}{y}\left(\frac{\mu}{\mu+k}\right)^{y}\left(\frac{k}{k+\mu}\right)^{k},$$

where $y\;=\;0,1,2,\ldots,$ and $\mu>0,$ and $k>0.$ The overdispersion parameter is $1/k,$ and $\mathrm{Var}(Y)\;=\;\mu+\dfrac{\mu^2}{k}\;>\;\mu.$ The Poisson and negative binomial models described in the GLM framework are implemented in R using the `glm()` function (Chambers and Hastie, 1993) in the stats package and the `glm.nb()` function in the MASS package (Venables and Ripley, 2002). But in the implementations of these functions, the parameter k is a scalar or 'intercept-only', that is, the parameter k cannot be modeled as a function of covariates $\boldsymbol{x}$, as for example in:

$$\log k\;=\;\beta_{(2)1}+\beta_{(2)2}\,x_2. \tag{3.21}$$

In contrast, in the VGLM framework, equation (3.21) can be fitted as follows:

$$\begin{aligned}\log\mu\;&=\;\eta_1\;=\;\boldsymbol{\beta}_1^T\boldsymbol{x},\\ \log k\;&=\;\eta_2\;=\;\boldsymbol{\beta}_2^T\boldsymbol{x}.\end{aligned} \tag{3.22}$$

Suppose that there are three covariate terms $(p\;=\;3)$ of interest, where $x_{i1}\;=\;1$ corresponding to the intercept, and x_{ik} is the value of the variable x_k for individual i. Then, model (3.22) can be written as

$$\begin{aligned}
\eta_1(\boldsymbol{x}_i) &= \beta_{(1)1} + \beta_{(1)2}\, x_{i2} + \beta_{(1)3}\, x_{i3}, \\
\eta_2(\boldsymbol{x}_i) &= \beta_{(2)1} + \beta_{(2)2}\, x_{i2} + \beta_{(2)3}\, x_{i3}.
\end{aligned}$$

The usual negative binomial is a special case in which the covariates x_2 and x_3 are constrained to have no effect on η_2, leaving an "intercept-only model"

$$\begin{aligned}
\log \mu &= \eta_1 = \boldsymbol{\beta}_1^T \boldsymbol{x}, \\
\log k &= \eta_2 = \beta^*_{(2)1}.
\end{aligned}$$

Therefore, we would wish to fit

$$\begin{aligned}
\eta_1(\boldsymbol{x}_i) &= \beta^*_{(1)1} + \beta^*_{(1)2}\, x_{i2} + \beta^*_{(1)3}\, x_{i3}, \\
\eta_2(\boldsymbol{x}_i) &= \beta^*_{(2)1}, \qquad (3.23)
\end{aligned}$$

where only the starred quantities in (3.23) need to be estimated. The model can be re-written as

$$\begin{aligned}
\boldsymbol{\eta}(\boldsymbol{x}_i) &= \begin{pmatrix} \eta_1(\boldsymbol{x}_i) \\ \eta_2(\boldsymbol{x}_i) \end{pmatrix} \\
&= \begin{pmatrix} \beta^*_{(1)1} & \beta^*_{(1)2} & \beta^*_{(1)3} \\ \beta^*_{(2)1} & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta^*_{(1)1} \\ \beta^*_{(2)1} \end{pmatrix} x_{i1} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \beta^*_{(1)2}\, x_{i2} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \beta^*_{(1)3}\, x_{i3}, \qquad (3.24)
\end{aligned}$$

which is of the form $\sum_{k=1}^{3} \mathbf{H}_k\, \boldsymbol{\beta}^*_k\, x_{ik}$ with the set of constraint matrices $\mathbf{H}_1$, $\mathbf{H}_2$, and, $\mathbf{H}_3$ as follows:

$$\mathbf{H}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ \mathbf{H}_2 = \mathbf{H}_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

3.3.2 Bivariate logistic model

Let us consider modeling two binary responses for the presence or absence of cataracts in elderly patients's eyes. Here, y_1 and y_2 are the presence or absence of cataracts of the left and right eye respectively. The bivariate logistic model is a natural choice for modeling this type of data. The responses of this model are generally dependent and the association between the two binary responses is modeled in terms of the odds ratio. Let $\boldsymbol{y} = (y_1, y_2)^T$ be the bivariate binary response variables with the distribution expressed as the four joint probabilities of $p_{rs} = \Pr(y_1 = r, y_2 = s)$, $r, s = 0, 1$. McCullagh and Nelder (1989, Section 6.5.6) and Palmgren (1989) specified the bivariate logistic (or logit) model as the logistic transforms of the marginal distributions of each y_i, together with an equation for the odds ratio, ψ, as

$$\begin{aligned} p_j &= \Pr(y_i = 1) = \frac{\exp\{\eta_j(\boldsymbol{x})\}}{1+\exp\{\eta_j(\boldsymbol{x})\}}, \qquad j = 1,2, \\ \log\psi(\boldsymbol{x}) &= \eta_3(\boldsymbol{x}). \end{aligned} \tag{3.25}$$

In (3.25), the odds ratio is $\psi = p_{00}\,p_{11}/(p_{01}\,p_{10})$. A probability, p_{11} can be obtained in terms of the marginal probabilities $p_1 = p_{11} + p_{10}$, $p_2 = p_{11} + p_{01}$, and ψ as

$$p_{11} = \begin{cases} \frac{1}{2}(\psi-1)^{-1}\{a - \sqrt{a^2+b}\}, & \psi \neq 1, \\ p_1 p_2, & \psi = 1, \end{cases}$$

where $a = 1 + (p_1 + p_2)(\psi - 1)$ and $b = -4\psi(\psi-1)p_1 p_2$. The remaining joint probabilities can also be obtained from the marginal probabilities in a similar way.

For example, with a bivariate logistic (odds ratio) model applied to two binary responses for the presence or absence of cataracts in elderly patients's eyes, one should test whether the covariates have the same effect on the responses. Therefore, we would need to constrain $\eta_1 = \eta_2$. This error structure is called "exchangeable" (cf. Yee and Wild, 1996):

$$\operatorname{logit} p_j(\boldsymbol{x}) = \eta_1(\boldsymbol{x}), \qquad j = 1,2, \tag{3.26}$$

$$\log\psi(\boldsymbol{x}) = \eta_3(\boldsymbol{x}). \tag{3.27}$$

In McCullagh and Nelder (1989, Section 6.5.6) and Palmgren (1989), the odds-ratio is modeled as a single parameter. If we combine this with constraining intercepts to be equal and the covariates (x_2 and x_3) to act in the same way for both eyes (η_1 and η_2), then we have

$$\begin{aligned}
\eta_1(\boldsymbol{x}_i) &= \beta^*_{(1)1} + \beta^*_{(1)2}\, x_{i2} + \beta^*_{(1)3}\, x_{i3}, \\
\eta_2(\boldsymbol{x}_i) &= \beta^*_{(1)1} + \beta^*_{(1)2}\, x_{i2} + \beta^*_{(1)3}\, x_{i3}, \\
\eta_3(\boldsymbol{x}_i) &= \beta^*_{(2)1}.
\end{aligned} \tag{3.28}$$

Here, the odds ratio has an intercept-only model, with three covariate terms $(p = 3)$, and we assume that $x_{i1} = 1$ is an intercept. The model (3.28) can be re-written as

$$\begin{aligned}
\boldsymbol{\eta}(\boldsymbol{x}_i) &= \begin{pmatrix} \eta_1(\boldsymbol{x}_i) \\ \eta_2(\boldsymbol{x}_i) \\ \eta_3(\boldsymbol{x}_i) \end{pmatrix} \\
&= \begin{pmatrix} \beta^*_{(1)1} & \beta^*_{(1)2} & \beta^*_{(1)3} \\ \beta^*_{(1)1} & \beta^*_{(1)2} & \beta^*_{(1)3} \\ \beta^*_{(2)1} & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta^*_{(1)1} \\ \beta^*_{(2)1} \end{pmatrix} x_{i1} + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \beta^*_{(1)2}\, x_{i2} + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \beta^*_{(1)3}\, x_{i3},
\end{aligned} \tag{3.29}$$

which is again of the form $\sum_{k=1}^{3} \mathbf{H}_k \, \boldsymbol{\beta}^*_k \, x_{ik}$ with slightly different constraint matrices $\mathbf{H}_k$.

The next subsection describes a maximum likelihood estimation based on iteratively reweighted least squares for the constrained VGLMs and the method used to estimate the constrained VGAMs.

3.3.3 VGLMs and constraint matrices

Equations (3.24) and (3.29) illustrate how constraint matrices can be accommodated within the VGLM framework with the general form of

$$\boldsymbol{\eta}(\boldsymbol{x}_i) \;=\; \mathbf{B}^T \boldsymbol{x}_i \;=\; \sum_{k=1}^{p} \mathbf{H}_k \, \boldsymbol{\beta}^*_{(k)} \, x_{ik},$$

where $\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_p$ are known full-column-rank matrices known as constraint matrices, $\boldsymbol{\beta}^*_{(k)} = \left(\beta^*_{(1)k}, \ldots, \beta^*_{(\mathrm{R}_k)k}\right)^T$, is a vector containing a possibly reduced set of regression coefficients, and

$$\mathbf{B}^T = \left(\mathbf{H}_1 \boldsymbol{\beta}^*_{(1)} \quad \mathbf{H}_2 \boldsymbol{\beta}^*_{(2)} \quad \cdots \quad \mathbf{H}_p \boldsymbol{\beta}^*_{(p)}\right).$$

Where no constraints are imposed on a set of $\boldsymbol{\beta}$-coefficients, then the relevant constraint matrix is an $M \times M$ identity matrix. Unconstrained coefficients such as this are said to be subject to "trivial constraints" in VGAM. The parameter vector to be estimated is given by

$$\boldsymbol{\beta}^* = \left(\boldsymbol{\beta}^{*T}_{(1)}, \ldots, \boldsymbol{\beta}^{*T}_{(p)}\right)^T.$$

Yee and Wild (1996) maximized the log-likelihood of the constrained problem using Fisher scoring or Newton Raphson, which can be implemented by iteratively minimizing the objective function

$$\text{minimize} \qquad \sum_{i=1}^{n} \left(\boldsymbol{z}_i - \sum_{k=1}^{p} \mathbf{H}_k \boldsymbol{\beta}^*_k x_{ik}\right)^T \mathbf{W}_i \left(\boldsymbol{z}_i - \sum_{k=1}^{p} \mathbf{H}_k \boldsymbol{\beta}^*_k x_{ik}\right)$$

with respect to $\{\boldsymbol{\beta}^*_k\}$. The relevant minimization problem is again the GLS problem (as described in the unconstrained problem, cf. equation (3.10)), but with a transformed "vector linear model" model matrix given by

$$\mathbf{X}_{\text{VLM}} = \left((\mathbf{X}_{\text{LM}}\, \boldsymbol{e}_1) \otimes \mathbf{H}_1 \,\Big|\, (\mathbf{X}_{\text{LM}}\, \boldsymbol{e}_2) \otimes \mathbf{H}_2 \,\Big|\, \cdots \,\Big|\, (\mathbf{X}_{\text{LM}}\, \boldsymbol{e}_p) \otimes \mathbf{H}_p\right), \qquad (3.30)$$

cf. $\mathbf{X}_{\text{VLM}}$ for the unconstrained problem in equation (3.10), where $\otimes$ is the Kronecker product and $\boldsymbol{e}_k$ is a zero vector with a one in the kth position. In IRLS, Yee and Wild (1996) regressed adjusted dependent vectors $\boldsymbol{z}_i = \boldsymbol{\eta}_i + \mathbf{W}_i^{-1}\boldsymbol{u}_i$ on $\mathbf{X}_{\text{VLM}}$ given by (3.30), with $\boldsymbol{u}_i = w_i\, \partial\ell_i / \partial\eta_i$ and the working weights $\mathbf{W}_i = -w_i\, E\left(\partial^2\ell_i / \left(\partial\eta_i\, \partial\eta_i^T\right)\right)$, until convergence is met (cf. the IRLS procedure for the unconstrained problem in Section 3.1).

3.3.4 VGAMs and constraint matrices

Yee and Wild (1996) also used "constraints on the functions" to enforce relationship between the $f_{(j)k}$ of VGAMs. Analogous to the constrained VGLM case in Sections 3.3.1 and 3.3.2, the smooth-function equivalent of (3.23) and (3.28) can be written as

$$\begin{aligned} \eta_1(\boldsymbol{x}_i) &= \beta^*_{(1)1} + f^*_{(1)2}(x_{i2}) + f^*_{(1)3}(x_{i3}), \\ \eta_2(\boldsymbol{x}_i) &= \beta^*_{(2)1}, \end{aligned} \tag{3.31}$$

and

$$\begin{aligned} \eta_1(\boldsymbol{x}_i) &= \beta^*_{(1)1} + f^*_{(1)2}(x_{i2}) + f^*_{(1)3}(x_{i3}), \\ \eta_2(\boldsymbol{x}_i) &= \beta^*_{(1)1} + f^*_{(1)2}(x_{i2}) + f^*_{(1)3}(x_{i3}), \\ \eta_3(\boldsymbol{x}_i) &= \beta^*_{(2)1}. \end{aligned} \tag{3.32}$$

respectively. For VGAMs, Yee and Wild (1996) represented these models using

$$\boldsymbol{\eta}(\boldsymbol{x}) = \sum_{k=1}^{p} \mathbf{H}_k \, \boldsymbol{f}^*_k \, (x_k) \tag{3.33}$$

where $\boldsymbol{f}^*_k$ is a vector containing a possibly reduced set of component functions written in the forms of $\boldsymbol{f}^*_k = \left(f^*_{(1)k}\,(x_k)\,, \ldots, f^*_{(r_k)k}\,(x_k)\right)^T$, and $\mathbf{H}_k$ are the constraint matrices. Like $\boldsymbol{f}_k$, each smooth component in $\boldsymbol{f}^*_k$ is centered for uniqueness. If there are no constraints, the relevant constraint matrix is an $M \times M$ identity matrix and $\boldsymbol{f}^*_k = \boldsymbol{f}_k = \left(f_{(1)k}\,(x_k), \ldots, f_{(M)k}\,(x_k)\right)^T$.

To estimate the constrained VGAMs, Yee and Wild (1996) extended linear constraints on the component functions $\mathbf{H}_k \, \boldsymbol{f}^*_k \, (x_{ik})$ to restrictions of the form $\mathbf{H}_k \, \boldsymbol{f}^*_k \, (x_{ik}) + \boldsymbol{c}_k$, for some vector $\boldsymbol{c}_k$. They fitted the kth variable within the backfitting algorithm by minimizing the problem

$$\begin{aligned} &\sum_{i=1}^{n} \left\{\boldsymbol{z}_i^{[k]} - \mathbf{H}_k \boldsymbol{f}^*_k(x_{ik}) - \boldsymbol{c}_k\right\}^T \mathbf{W}_i \, \left\{\boldsymbol{z}_i^{[k]} - \mathbf{H}_k \boldsymbol{f}^*_k(x_{ik}) - \boldsymbol{c}_k\right\} + \sum_j \lambda_{(j)k} \int f^{*''}_{(j)k}(t)^2 \, \mathrm{d}t, \qquad (3.34) \\ &= \text{constant} + \sum_{i=1}^{n} \left\{\boldsymbol{z}_i^{*[k]} - \boldsymbol{f}^*_k(x_{ik})\right\}^T \mathbf{W}^*_{i,k} \, \left\{\boldsymbol{z}_i^{*[k]} - \boldsymbol{f}^*_k(x_{ik})\right\} + \sum_j \lambda_{(j)k} \int f^{*''}_{(j)k}(t)^2 \, \mathrm{d}t, \end{aligned}$$

cf. equations (3.15) and (3.17), where

$$\boldsymbol{x}_i^{*[k]} = \left(\mathbf{H}_k^T \mathbf{W}_i \mathbf{H}_k\right)^{-1} \mathbf{H}_k^T \, \mathbf{W}_i \left(\boldsymbol{z}_i^{[k]} - \boldsymbol{c}_k\right) \tag{3.35}$$

$$\text{and} \quad \mathbf{W}^*_{i,k} = \mathbf{H}_k^T \, \mathbf{W}_i \, \mathbf{H}_k. \tag{3.36}$$

The relevant minimization problem is again a vector spline problem, but with a dependent vector and weight matrix transformed respectively into (3.35) and (3.36). A detailed explanation of the constrained VGAMs is given in Yee and Wild (1994).

3.4 Conclusions

VGLMs/VGAMs are extensions of GLMs/GAMs made to include a wide class of multivariate regression models. The underlying algorithm of VGLMs is IRLS, while VGAMs are fitted via IRLS and modified vector backfitting using vector splines. The constraints on the functions are a very important component of the VGLM/VGAM class for applied work as they add a great deal of modeling flexibility. In this chapter, we have provided an introduction to the theory required for model construction, constraint matrices, and estimation for VGLMs/VGAMs, emphasizing elements that will be extended or modified later in this thesis.

Chapter 4

P-spline VGAMs

VGAMs (Yee and Wild, 1996) are VGLMs with linear predictors replaced by a sum of smooth functions of covariates. The model has the form of $\eta_j(\boldsymbol{x}) = \sum_{k=1}^{p} f_{(j)k}(x_k)$ (cf. equation (3.12)). Parameter estimation was achieved by a combination of IRLS and modified vector backfitting using vector splines. In general, the backfitting approach requires the users to manually investigate the possible values for the target equivalent degrees of freedom, used as a smoothing parameter. In backfitting, the algorithm used for estimating all the smooth terms in the models is suitable only for estimating single smooth terms individually. Its iterative procedure does not provide straightforward expressions for the estimation of smooth components. This is why a computational method for automatic smoothing parameter estimation cannot be implemented easily within the backfitting approach. This leads to difficulties with VGAMs for smoothness estimation and inference.

To show how the manual choice of degrees of smoothness affects the shape of the estimated functions of VGAMs based on the backfitting approach, we consider the fitted curves from the LMS method for quantile regression on the age-centile curves of European women using the VGAM approach. The LMS method is a popular technique for quantile regression. It was originally proposed by Cole (1988) and Cole and Green (1992). Yee (2004) called this the classical LMS-normal method. The basic idea is as follows. At a fixed value of x, a Box-Cox power transformation is applied to a response in order to transform it to normality. Smooths of quantiles are then obtained on the normal scale, and these are then back-transformed to the original scale. LMS was named from the starting letter 'L-M-S' of the three parameters λ, μ, σ which have to be estimated. Cole and Green (1992) estimated the three parameters by penalized likelihood using an iterative smoothing spline. Yee (2004) fitted all three functions simultaneously using a vector smoothing spline. Further information is given in Green and Silverman (1993) and Wright and Royston (1997).

For illustrative purposes, we will use a cross-sectional data set from the VGAMdata package of a company's workforce data combined with health-survey data, from New Zealand during the 1990s. The variables of interest are: age ranging between 16 and 88 years (`age`), and body mass index ($\text{BMI} = \text{weight}/\text{height}^2, \text{kg m}^{-2}$), used for measuring obesity. We confine our analysis to a subset of 2600 European women. Missing values were removed. The LMS-normal method was fitted to this data set with $Y = \text{BMI}$ and $X_2 = \text{age}$ using the VGAM package (Yee, 2008) with the family function `lms.bcn`. Three different sets of manual choices of degrees of freedom will be used: `c(1.5, 1.5, 1.5)`, `c(4, 15, 4)`, and `c(1, 4, 1)` coded as follows:

```
fit1 <- vgam(BMI ~ s(age, df = c(1.5, 1.5, 1.5)), lms.bcn(zero = NULL),
             data = women.eth0s)
fit2 <- vgam(BMI ~ s(age, df = c(4, 15, 4)), lms.bcn(zero = NULL),
             data = women.eth0s)
fit3 <- vgam(BMI ~ s(age, df = c(1, 4, 1)), lms.bcn(zero = NULL),
             data = women.eth0s)
```

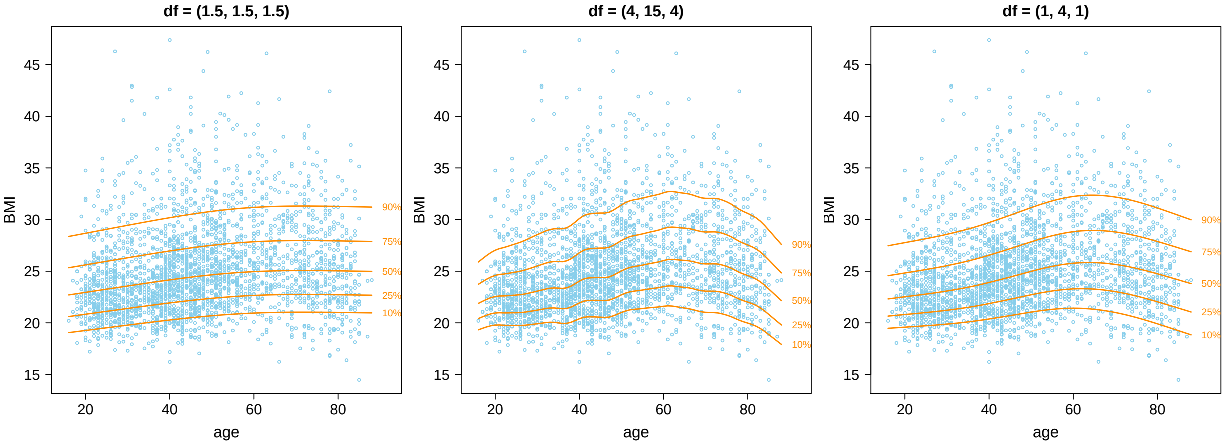


Figure 4.1: Quantile regression fits to dataset xs.nz in VGAMdata using the LMS method. The numbers in brackets at the top of each panel are the manual choice of the degrees of freedom. The solid lines represent the estimated smooth quantiles. The plots represent underfitting, overfitting, and a good fit respectively.

Fig. 4.1 shows the smooth quantile estimates that the modified vector backfitting yields. By changing the value of the degrees of smoothness, we obtain a variety of fitted models. The left panel in Fig. 4.1 shows that a small value for the degrees of freedom leads to nearly linear estimates or the model underfitting, while a large value of the degrees of freedom results in an excessive wiggliness of the estimated smooth function from model overfitting as shown in the middle panel of Fig. 4.1. Fig. 4.1 shows that the choice of degree of smoothness of smooth terms plays a crucial role in determining the flexibility of models, and the estimated shape of the smooth functions. Being able to automatically determine the shape of nonlinear terms from the data for VGAMs is highly desirable. While a reasonable fitted model can be obtained by manually turning the degrees of freedom as shown in Fig. 4.1 (right panel), a well-experienced researcher is required to tune these spline parameters.

We will develop an alternative estimation procedure for estimating model coefficients for the VGAM class by adapting automatic numerical procedures previously described to determine the shape of nonlinear terms from the data to the VGAM framework. We do this by generalizing the ideas of penalized regression splines for GAM modeling to the VGAM class.

To construct VGAMs using penalized regression splines, we need a basis for the smooth function, which is large enough to approximate the function well, but small enough for efficient computation. And to avoid overfitting with smooths, we need a measure of function "wiggliness" that can be used to penalize overly complex models during fitting. P-splines are an attractive approach for representing VGAMs as penalized regression splines. They are low-rank smoothers represented by B-splines, usually defined on equally-spaced knots, together with a difference penalty applied directly to the parameters to control function wiggliness. They also have the attractive properties described in Section 2.3.

We will develop VGAMs based on penalized regression splines using P-spline smoothers, which we term "P-spline VGAMs". By using P-spline smoothers for every smooth component, we will transform VGAMs into the VGLM framework and fit P-spline VGAMs by penalized likelihood maximization.

4.1 Setting up P-spline VGAMs as penalized VGLMs

The purpose of this section is to document exactly how P-spline VGAMs can be constructed using penalized regression splines in a way that allows the smoothness selection to be integrated, and to describe penalized likelihood estimation for the approach proposed.

4.1.1 Modeling P-spline VGAMs with basis functions

We will now reformulate VGAMs using penalized regression splines based on P-spline smoothers. Each smooth term in (3.12) is rewritten using a set of B-splines and has an associated 'discrete' penalty measuring its wiggliness. The model is considered in the form of

$$\eta_j\left(\boldsymbol{x}\right) \;=\; \sum_{k=1}^{p} f_{(j)k}\left(x_k\right), \qquad j \;=\; 1, \ldots, M,$$

or in vector form

$$\boldsymbol{\eta}\left(\boldsymbol{x}\right) \;=\; \sum_{k=1}^{p} \boldsymbol{f}_k\left(x_k\right), \tag{4.1}$$

where $\boldsymbol{x} = (x_1, \ldots, x_p)^T$. Each smooth term $f_{(j)k}\,(x_k)$ in (4.1) is represented as

$$f_{(j)k}(x_k) \;=\; \sum_{s=1}^{\mathrm{S}_k} a_{(j)k:s}\, B_{k:s}(x_k), \tag{4.2}$$

where S_k is the number of B-spline basis functions used for $f_{(j)k}(x_k)$, the $B_{k:s}(x_k)$ are B-spline basis function and the $\{a_{(j)k:s}\}$ are coefficients associated with the B-spline basis functions that are estimated as part of model fitting. We define $\boldsymbol{f}_k(x_k)$ in (4.1) in the same ways as it was defined in VGAMs (cf. equation (3.13)),

$$\boldsymbol{f}_k(x_k) \;=\; \left(f_{(1)k}(x_k), \ldots, f_{(M)k}(x_k)\right)^T.$$

Recall if there is an intercept, $x_1 = 1$. The $f_{(j)k}\,(x_k)$ of equation (4.1) is fitted using a set of B-splines as basis functions, usually defined on evenly spaced knots (cf. Marx and Eilers, 1998). Let

$$\boldsymbol{f}_{(j)} \;=\; \left(f_{(j)1}(x_1), \ldots, f_{(j)p}(x_p)\right)^T$$

be a vector containing all smooth components modeled at the jth smooth predictor. Each smooth component function in $\boldsymbol{f}_{(j)}$ is subject to a centering constraint in order to ensure that the model is identifiable. Following Gill et al. (1981) and Wood (2006b, Sections 1.8.1 and 4.2), we use the constraint such as

$$\sum_i f_{(j)k}(x_{ik}) \;=\; 0$$

for each smooth component in $\boldsymbol{f}_{(j)}$. In this thesis, we model each smooth component function in $\boldsymbol{f}_k(x_k)$ with the same knots. Given a set of B-splines as a basis, we re-write (4.2) in the matrix notation as

$$\boldsymbol{f}_{(j)k} \;=\; \mathrm{X}_k^* \boldsymbol{\beta}_{(j)k}, \tag{4.3}$$

where $\boldsymbol{\beta}_{(j)k} = \left(a_{(j)k:1}, \ldots, a_{(j)k:\mathrm{S}_k}\right)^T$ is a $\mathrm{S}_k \times 1$ B-spline coefficient vector at the jth smooth predictor and kth predictor, and X_k^* is an $n \times \mathrm{S}_k$ matrix containing B-splines generated from the values of x_k. Here, S_k is the number of knots for x_k. We then define the parameter vector for one η_j, $\boldsymbol{\beta}_{(j)} = \left(\boldsymbol{\beta}_{(j)1}^T, \ldots, \boldsymbol{\beta}_{(j)p}^T\right)^T$. Given centered model matrices for each smooth term, we re-write (4.1) as

$$\begin{aligned}\eta_j(\boldsymbol{x}) &= \sum_{k=1}^{p} f_{(j)k}\left(x_k\right)\\ &= \sum_{k=1}^{p} \mathrm{X}_k^* \,\boldsymbol{\beta}_{(j)k}\\ &= \mathrm{X}^* \,\boldsymbol{\beta}_{(j)},\end{aligned} \tag{4.4}$$

where $\mathrm{X}^* = \left(\mathrm{X}_1^* \,|\, \ldots \,|\, \mathrm{X}_p^*\right)$ is an $n \times \sum_{k=1}^{p} \mathrm{S}_k$ block matrix containing the matrices X_k^*, as its column blocks, $k = 1, \ldots, p$. Let $\boldsymbol{x}_{ik}^* = \left(B_{k:1}(x_{ik}), \ldots, B_{k:\mathrm{S}_k}(x_{ik})\right)^T$ be a $\mathrm{S}_k \times 1$ vector of B-splines generated from the values of x_{ik}, and $\boldsymbol{x}_i^* = \left(\boldsymbol{x}_{i1}^{*T}, \ldots, \boldsymbol{x}_{ip}^{*T}\right)^T$ be a $(\sum_{k=1}^{p} \mathrm{S}_k) \times 1$ vector stacking the $\boldsymbol{x}_{ik}^*$. We now write the smooth predictor vector $\boldsymbol{\eta}$ in terms of the observations as follows:

$$\begin{aligned}\boldsymbol{\eta}_i &= \boldsymbol{\eta}(\boldsymbol{x}_i) = \begin{pmatrix}\eta_1\left(\boldsymbol{x}_i\right)\\ \vdots\\ \eta_M\left(\boldsymbol{x}_i\right)\end{pmatrix}\\ &= \boldsymbol{f}_1(x_{i1}) + \cdots + \boldsymbol{f}_p(x_{ip})\\ &= \begin{pmatrix} a_{(1)1:1} & \cdots & a_{(1)1:\mathrm{S}_1}\\ \vdots & & \vdots\\ a_{(M)1:1} & \cdots & a_{(M)1:\mathrm{S}_1}\end{pmatrix} \boldsymbol{x}_{i1}^* + \cdots + \begin{pmatrix} a_{(1)p:1} & \cdots & a_{(1)p:\mathrm{S}_p}\\ \vdots & & \vdots\\ a_{(M)p:1} & \cdots & a_{(M)p:\mathrm{S}_p}\end{pmatrix} \boldsymbol{x}_{ip}^*\\ &= \left(\boldsymbol{\beta}_{1:1} \cdots \boldsymbol{\beta}_{1:\mathrm{S}_1}\right) \boldsymbol{x}_{i1}^* + \cdots + \left(\boldsymbol{\beta}_{p:1} \cdots \boldsymbol{\beta}_{p:\mathrm{S}_p}\right) \boldsymbol{x}_{ip}^*,\end{aligned} \tag{4.5}$$

where $\boldsymbol{\beta}_{k:s} = \left(a_{(1)k:s}, \ldots, a_{(M)k:s}\right)^T$, $k = 1, \ldots, p$ and $s = 1, \ldots, \mathrm{S}_k$. Before proceeding, we need to define the 'vectorization' of a matrix. The 'vectorization' of a matrix is a linear transformation which converts the matrix into a column vector. For example, the vectorization of an $n \times m$ matrix $\mathbf{A}$, denoted by $\mathrm{vec}(\mathbf{A})$, is the $nm \times 1$ column vector obtained by stacking the columns of the matrix $\mathbf{A}$ below one another:

$$\mathrm{vec}(\mathbf{A}) = \left(a_{11}, \ldots, a_{n1}, a_{12}, \ldots, a_{n2}, \ldots, a_{1m}, \ldots, a_{nm}\right)^T.$$

By applying the 'vectorization' to the matrix $\left(\boldsymbol{\beta}_{k:1} \ldots \boldsymbol{\beta}_{k:\mathrm{S}_k}\right)$, we obtain a $(\mathrm{S}_k \cdot M) \times 1$ B-spline coefficient vector $\boldsymbol{\beta}_k$ as follows:

$$\begin{aligned}
\boldsymbol{\beta}_k &= \operatorname{vec}\left(\boldsymbol{\beta}_{k:1} \ldots \boldsymbol{\beta}_{k:\mathrm{S}_k}\right) \\
&= \operatorname{vec}\begin{pmatrix} a_{(1)k:1} & \cdots & a_{(1)k:\mathrm{S}_k} \\ \vdots & & \vdots \\ a_{(M)k:1} & \cdots & a_{(M)k:\mathrm{S}_k} \end{pmatrix} \\
&= \operatorname{vec}\begin{pmatrix} \boldsymbol{\beta}_{(1)k}^T \\ \vdots \\ \boldsymbol{\beta}_{(M)k}^T \end{pmatrix}, \qquad (4.6)
\end{aligned}$$

where $\boldsymbol{\beta}_{(j)k} = \left(a_{(j)k:1}, \ldots, a_{(j)k:\mathrm{S}_k}\right)^T$. We re-write (4.5) as follows:

$$\begin{aligned}
\boldsymbol{\eta}_i &= \boldsymbol{\eta}(\boldsymbol{x}_i) \;=\; (\eta_1(\boldsymbol{x}_i), \ldots, \eta_M(\boldsymbol{x}_i))^T \\
&= \boldsymbol{f}_1(x_{i1}) + \cdots + \boldsymbol{f}_p(x_{ip}) \\
&= (\boldsymbol{x}_{i1}^{*T} \otimes \mathbf{I}_M)\, \boldsymbol{\beta}_1 + \cdots + (\boldsymbol{x}_{ip}^{*T} \otimes \mathbf{I}_M)\, \boldsymbol{\beta}_p \\
&= \mathbf{X}_{i1}\, \boldsymbol{\beta}_1 + \cdots + \mathbf{X}_{ip}\, \boldsymbol{\beta}_p \\
&= \sum_{k=1}^{p} \mathbf{X}_{ik}\, \boldsymbol{\beta}_k, \qquad (4.7)
\end{aligned}$$

where $\mathbf{X}_{ik} = \boldsymbol{x}_{ik}^{*T} \otimes \mathbf{I}_{\mathrm{M}}$ is an $M \times (\mathrm{S}_k \cdot M)$ matrix. In other words, the matrix $\mathbf{X}_{ik}$ here is a block diagonal matrix made up of M copies of $\boldsymbol{x}_{ik}^{*T}$. Let

$$\mathbf{X}_i \;=\; (\mathbf{X}_{i1} \,|\, \ldots \,|\, \mathbf{X}_{ip}) \qquad (4.8)$$

be an $M \times \left(\sum_{k=1}^{p} \mathrm{S}_k \cdot M\right)$ block matrix containing the matrices $\mathbf{X}_{ik}$ as its column blocks. Let $\boldsymbol{\beta} = \left(\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_p^T\right)^T$ is a $\sum_{k=1}^{p}(\mathrm{S}_k \cdot M) \times 1$ parameter vector. If the intercept is in a model, then the matrix $\mathbf{X}_i$ becomes an $M \times \left(M + \sum_{k=2}^{p} \mathrm{S}_k \cdot M\right)$ matrix given by $\mathbf{X}_i = (\mathbf{I}_{\mathrm{M}} \,|\, \mathbf{X}_{i2} \,|\, \ldots \,|\, \mathbf{X}_{ip})$, and $\boldsymbol{\beta}_1 = \left(\beta_{(1)1}, \ldots, \beta_{(M)1}\right)^T$. We then re-write equation (4.7) in a general matrix-vector form as

$$\boldsymbol{\eta} \;=\; \mathbf{X}_{\mathrm{VAM}}\, \boldsymbol{\beta}, \qquad (4.9)$$

where $\boldsymbol{\eta} = \left(\boldsymbol{\eta}_1^T, \ldots, \boldsymbol{\eta}_n^T\right)^T$, and

$$\mathbf{X}_{\text{VAM}} = \mathbf{X}_{\text{AM}} \otimes \mathbf{I}_{\text{M}} = \left(\mathbf{X}_1^T, \ldots, \mathbf{X}_n^T\right)^T \tag{4.10}$$

is the "vector additive model" model matrix with dimensions $(n \cdot M) \times \left(\sum_{k=1}^{p} \mathrm{S}_k \cdot M\right)$. Here, $\mathbf{X}_{\text{AM}} = \left(\mathrm{X}_1^* \,|\, \ldots \,|\, \mathrm{X}_p^*\right)$ is the "additive model" model matrix constructed in the manner described in Section 2.3, and $\mathbf{I}_{\text{M}}$ is the $M \times M$ identity matrix. By using B-splines for each smooth component function $f_{(j)k}(x_k)$, and imposing the identifiability constraints on the model before fitting, we have transformed equation (4.1) into the form of VGLMs to that shown in (4.9), where the ith row of the model matrix is now equivalent to $\mathbf{X}_i$, and the parameter vector is $\boldsymbol{\beta} = \left(\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_p^T\right)^T$. We therefore write down its log-likelihood as (cf. equation (3.6))

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} w_i \, \ell\{\eta_1(\boldsymbol{x}_i), \ldots, \eta_M(\boldsymbol{x}_i)\}. \tag{4.11}$$

Here, $\eta_j(\boldsymbol{x}_i) = \sum_{k=1}^{p} f_{(j)k}(x_k) = \boldsymbol{\beta}_{(j)}^T \boldsymbol{x}_i^*$.

If the number of B-spline basis functions is large enough, then the estimated smooth components are able to approach the unknown underlying true function, and the $\boldsymbol{\beta}$ can be estimated by ordinary likelihood maximization. But with a relatively large number of basis functions, this can cause substantially overfitting. So, as in Chapter 2, we have to impose penalties during model fitting to control model's smoothness. In this thesis, we use the penalty based on finite differences of adjacent B-splines coefficients (cf. Section 2.3) to control overfitting.

4.1.2 The penalty for P-spline VGAMs

Now we consider how measures of function wiggliness based on finite differences of adjacent B-spline coefficients can be constructed for VGAMs. Recall that the function's wiggliness for univariate GAMs is measured by (for variable x_k)

$$\sum_{s=1}^{\mathrm{S}_k} \left(\Delta^{[d]} \, a_{k:s}\right)^2 = \boldsymbol{\beta}_k^T \, \mathbf{D}_{[d]k}^T \, \mathbf{D}_{[d]k} \, \boldsymbol{\beta}_k = \boldsymbol{\beta}_k^T \, \mathbf{P}_{[d]k} \, \boldsymbol{\beta}_k, \tag{4.12}$$

where $\mathbf{P}_{[d]k} = \mathbf{D}_{[d]k}^T \mathbf{D}_{[d]k}$ is a $\mathrm{S}_k \times \mathrm{S}_k$ penalty matrix, and $\boldsymbol{\beta}_k = (a_{k:1}, \ldots, a_{k:\mathrm{S}_k})^T$ denote the vector of the B-spline coefficients for the kth predictor. The matrix $\mathbf{D}_{[d]k}$ is a $(\mathrm{S}_k - d_k) \times \mathrm{S}_k$

penalty building-block matrix, where each row consists of the contrasts of dth order polynomial, and S_k is the number of knots for x_k. To extend the penalty in (4.12) to the VGAM class, we first define the penalty at the jth smooth predictor as follows:

$$\begin{aligned} J_j(\boldsymbol{\lambda}) &= \sum_{k=1}^{p} \sum_{s=1}^{\mathrm{S}_k} \lambda_{(j)k} \left(\Delta^{[d]}\, a_{k:s} \right)^2 \\ &= \sum_{k=1}^{p} \lambda_{(j)k}\, \boldsymbol{\beta}_{(j)k}^T\, \mathbf{D}_{[d]k}^T\, \mathbf{D}_{[d]k}\, \boldsymbol{\beta}_{(j)k} \\ &= \sum_{k=1}^{p} \boldsymbol{\beta}_{(j)k}^T \left\{ \lambda_{(j)k} \left(\mathbf{D}_{[d]k}^T\, \mathbf{D}_{[d]k} \right) \right\} \boldsymbol{\beta}_{(j)k}. \end{aligned} \tag{4.13}$$

Here, $\lambda_{(j)k} \geq 0$ for all $j > 0$ and $k > 0$ are the smoothing parameters that control the tradeoff between fit and smoothness, and $\boldsymbol{\beta}_{(j)k} = \left(a_{(j)k:1}, \ldots, a_{(j)k:\mathrm{S}_k} \right)^T$ (cf. equation (4.3)). Then, we define $\mathbf{P}_{\lambda(j)}$ as a $\sum_{k=1}^{p} \mathrm{S}_k \times \sum_{k=1}^{p} \mathrm{S}_k$ diagonal block matrix, where the main diagonal blocks are the $\mathrm{S}_k \times \mathrm{S}_k$ matrix of $\lambda_{(j)k} \left(\mathbf{D}_{[d]k}^T\, \mathbf{D}_{[d]k} \right)$. Thus, $\mathbf{P}_{\lambda(j)}$ is in the form of

$$\mathbf{P}_{\lambda(j)} = \begin{pmatrix} \lambda_{(j)1} \left(\mathbf{D}_{[d]1}^T\, \mathbf{D}_{[d]1} \right) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \lambda_{(j)p} \left(\mathbf{D}_{[d]p}^T\, \mathbf{D}_{[d]p} \right) \end{pmatrix}.$$

If there is an intercept, $\mathbf{P}_{\lambda(j)} = \text{blockdiag}\left(0, \lambda_{(j)2} \left(\mathbf{D}_{[d]2}^T\, \mathbf{D}_{[d]2} \right), \ldots, \lambda_{(j)p} \left(\mathbf{D}_{[d]p}^T\, \mathbf{D}_{[d]p} \right) \right)$ is a $\left(1 + \sum_{k=2}^{p} \mathrm{S}_k \right) \times \left(1 + \sum_{k=2}^{p} \mathrm{S}_k \right)$ matrix. We now write the penalty in (4.13) as a quadratic form in the parameter vector $\boldsymbol{\beta}_{(j)}$ as

$$J_j(\boldsymbol{\lambda}) = \boldsymbol{\beta}_{(j)}^T\, \mathbf{P}_{\lambda(j)}\, \boldsymbol{\beta}_{(j)}.$$

Next, we will show how the discrete wiggliness penalty $J_j(\boldsymbol{\lambda})$ can be extended to more than one smooth predictor. Since the smoothing parameters are given to the objective in order to smooth each component function, each smooth component function in $\boldsymbol{f}_k(x_k)$ will obtain the different values of the smoothing parameters. Let $\boldsymbol{\lambda}_k = \left(\lambda_{(1)k}, \ldots, \lambda_{(M)k} \right)^T$ be a vector containing a set of the smoothing parameters for each smooth component function in $\boldsymbol{f}_k(x_k)$. Let $\mathbf{P}_{\lambda k} = \left(\mathbf{D}_{[d]k}^T\, \mathbf{D}_{[d]k} \right) \otimes \text{diag}\left(\lambda_{(1)k}, \ldots, \lambda_{(M)k} \right)$ be a $(\mathrm{S}_k \cdot M) \times (\mathrm{S}_k \cdot M)$ penalty matrix for each smooth vector $\boldsymbol{f}_k(x_k)$. We then write the penalty term for the P-spline VGAMs as follows:

$$\begin{aligned} J(\boldsymbol{\lambda}) &= \sum_{k=1}^{p}\sum_{j=1}^{M} \lambda_{(j)k}\, \boldsymbol{\beta}_{(j)k}^T\, \mathbf{D}_{[d]k}^T\, \mathbf{D}_{[d]k}\, \boldsymbol{\beta}_{(j)k} \\ &= \sum_{k=1}^{p} \boldsymbol{\beta}_k \left\{ \left(\mathbf{D}_{[d]k}^T\, \mathbf{D}_{[d]k} \right) \otimes \operatorname{diag}\left(\lambda_{(1)k}, \ldots, \lambda_{(M)k} \right) \right\} \boldsymbol{\beta}_k \\ &= \sum_{k=1}^{p} \boldsymbol{\beta}_k^T\, \mathbf{P}_{\lambda k}\, \boldsymbol{\beta}_k. \end{aligned} \tag{4.14}$$

In this thesis, we model each smooth component function in $\boldsymbol{f}_k(x_k)$ with the same penalty building-block matrix $\mathbf{D}_{[d]k}$. Let

$$\mathbf{P}_\lambda \;=\; \text{blockdiag}\left(\mathbf{P}_{\lambda 1}, \cdots, \mathbf{P}_{\lambda p} \right) \tag{4.15}$$

be the $\left(\sum_{k=1}^{p} \mathrm{S}_k \cdot M\right) \times \left(\sum_{k=1}^{p} \mathrm{S}_k \cdot M\right)$ penalty matrix of the model. Here, $\mathbf{P}_{\lambda k}$ corresponds to the smooth vector $\boldsymbol{f}_k(x_k)$. The matrix $\mathbf{P}_{\lambda 1}$ is replaced by $\mathbf{0}$ if there is an intercept. We therefore re-write the penalty $J(\boldsymbol{\lambda})$ into the forms of the quadratic penalty on the parameter vector $\boldsymbol{\beta}$ as

$$J(\boldsymbol{\lambda}) \;=\; \boldsymbol{\beta}^T\, \mathbf{P}_\lambda\, \boldsymbol{\beta}.$$

As usual, $\boldsymbol{\beta}$ here is a parameter vector with $\boldsymbol{\beta}_k$ in its kth element. Clearly, given B-splines as a basis, we can always generate the matrix $\mathbf{P}_\lambda$ which allows the penalty $J(\boldsymbol{\lambda})$ to be written as a quadratic form in the parameter vector $\boldsymbol{\beta}$ (cf. Marx and Eilers (1998)). Given a wiggliness measure for each smooth function for all smooth predictors, a penalized log-likelihood for the P-spline VGAM model can be expressed as

$$\ell^*(\boldsymbol{\beta}) \;=\; \ell(\boldsymbol{\beta}) - \frac{1}{2} \sum_{k=1}^{p}\sum_{j=1}^{M} \lambda_{(j)k}\, \boldsymbol{\beta}_{(j)k}^T\, \mathbf{D}_{[d]k}^T\, \mathbf{D}_{[d]k}\, \boldsymbol{\beta}_{(j)k}. \tag{4.16}$$

For notational compactness, we re-write the log-likelihood objective (4.16) as

$$\ell^*(\boldsymbol{\beta}) \;=\; \ell(\boldsymbol{\beta}) - \frac{1}{2} \boldsymbol{\beta}^T\, \mathbf{P}_\lambda\, \boldsymbol{\beta}. \tag{4.17}$$

Given values for the $\lambda_{(j)k}$, then we would fit the P-spline VGAM by maximizing the penalized log-likelihood objective (4.17) in order to obtain $\widehat{\boldsymbol{\beta}}$.

4.1.3 P-spline VGAMs with parametric terms

The model structure presented so far has considered only models consisting of smooth terms. There is no difficulty, however, in combining parametric components and smooth functions of covariates (cf. Wood (Chapter 4, 2006b)). If a parametric term, such as, dummy and categorical variables is included as the kth term in the model, then $\mathbf{X}_{ik}$ in (4.8) corresponds to an $M \times M$ vector with corresponding parameter vector $\boldsymbol{\beta}_k$ of M dimensions, and in (4.15), the $\mathbf{P}_{\lambda k}$ term for a strictly parametric model component is set to a zero matrix as such terms are not penalized.

4.2 P-IRLS formulation via Fisher scoring

Given a B-spline as a basis and discrete wiggliness penalty for each smooth in the model, then, for given smoothing parameters $\lambda_{(j)k}$, the P-spline VGAMs can be estimated using penalized likelihood maximization. The penalized log-likelihood of the P-spline VGAM is estimated in the same manner as the log-likelihood objective of VGLMs is solved and this can be done using penalized iteratively reweighted least squares (P-IRLS). In this section, we will describe how P-spline VGAMs can be estimated using P-IRLS.

4.2.1 The Fisher scoring algorithm

We now spell out some of the details of maximizing the penalized log-likelihood using the Fisher scoring algorithm. The Newton algorithm for maximizing $\ell^*(\boldsymbol{\beta})$ (4.17) is given by

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \boldsymbol{\mathcal{I}}\left(\boldsymbol{\beta}^{(t)}\right)^{-1} \boldsymbol{U}\left(\boldsymbol{\beta}^{(t)}\right), \tag{4.18}$$

where (suppressing the superscript (t) for simplicity)

$$\begin{aligned} \boldsymbol{U}(\boldsymbol{\beta}) &= \frac{\partial \ell^*(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \\ &= \frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} - \frac{1}{2} \frac{\partial\left(\boldsymbol{\beta}^T \mathbf{P}_\lambda \boldsymbol{\beta}\right)}{\partial \boldsymbol{\beta}}, \end{aligned} \tag{4.19}$$

and

$$\begin{aligned}\boldsymbol{\mathcal{I}}\,(\boldsymbol{\beta}) &= -\frac{\partial^2 \ell^*(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}\,\partial\boldsymbol{\beta}^T} \\ &= -\left\{\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}\,\partial\boldsymbol{\beta}^T} - \frac{1}{2}\,\frac{\partial^2\left(\boldsymbol{\beta}^T\,\mathbf{P}_\lambda\,\boldsymbol{\beta}\right)}{\partial\boldsymbol{\beta}\,\partial\boldsymbol{\beta}^T}\right\}. \qquad (4.20)\end{aligned}$$

Here, $\boldsymbol{U}\,(\boldsymbol{\beta})$ is the score vector for the model and $\boldsymbol{\mathcal{I}}\,(\boldsymbol{\beta})$ is the observed information matrix. Recall from equation (4.17) that the log-likelihood $\ell(\boldsymbol{\beta})$ is of the form

$$\ell(\boldsymbol{\beta}) \;=\; \sum_{i=1}^{n} w_i\, \ell\{\eta_1(\boldsymbol{x}_i), \ldots, \eta_M(\boldsymbol{x}_i)\},$$

where $\eta_j(\boldsymbol{x}_i) \;=\; \sum_{k=1}^{p} f_{(j)k}(x_k) \;=\; \boldsymbol{\beta}_{(j)}^T \boldsymbol{x}_i^*$ and w_i are known positive prior weights (cf. equation (3.6)). Using the chain rule, the first derivatives of the log-likelihood $\ell(\boldsymbol{\beta})$ in (4.19) can be written as

$$\begin{aligned}\frac{\partial \ell}{\partial \boldsymbol{\beta}_j} &= \sum_{i=1}^{n} w_i\, \frac{\partial \ell_i}{\partial \eta_j} \cdot \frac{\partial \eta_j}{\partial \boldsymbol{\beta}_j} \\ &= \sum_{i=1}^{n} w_i\, \frac{\partial \ell_i}{\partial \eta_j}\, \boldsymbol{x}_i^*. \qquad (4.21)\end{aligned}$$

The second derivatives of the log-likelihood $\ell(\boldsymbol{\beta})$ in (4.20) satisfies the equations

$$\begin{aligned}-\frac{\partial^2 \ell}{\partial \boldsymbol{\beta}_j\,\partial \boldsymbol{\beta}_k^T} &= \sum_{i=1}^{n} -w_i \left\{\frac{\partial^2 \ell_i}{\partial \eta_j\,\partial \eta_k}\right\} \frac{\partial \eta_j}{\partial \boldsymbol{\beta}_j}\,\frac{\partial \eta_k}{\partial \boldsymbol{\beta}_k}, \\ &= \sum_{i=1}^{n} -w_i\, \frac{\partial^2 \ell_i}{\partial \eta_j\,\partial \eta_k}\, \boldsymbol{x}_i^*\, \boldsymbol{x}_i^{*T}. \qquad (4.22)\end{aligned}$$

Taking (4.19) and (4.20), then substituting in (4.21) and (4.22) as well as derivative of the penalty terms we have

$$\frac{\partial \ell^*}{\partial \boldsymbol{\beta}_j} \;=\; \sum_{i=1}^{n} w_i\, \frac{\partial \ell_i}{\partial \eta_j}\, \boldsymbol{x}_i^* - [\mathbf{P}_\lambda]_{(j)}\, \boldsymbol{\beta}_j, \qquad (4.23)$$

and

$$-\frac{\partial^2 \ell^*}{\partial \boldsymbol{\beta}_j\,\partial \boldsymbol{\beta}_k^T} \;=\; -\sum_{i=1}^{n} w_i\, \frac{\partial^2 \ell_i}{\partial \eta_j\,\partial \eta_k}\, \boldsymbol{x}_i^*\, \boldsymbol{x}_i^{*T} + [\mathbf{P}_\lambda]_{(j)} \cdot \delta_{jk}, \qquad (4.24)$$

where $[\mathbf{P}_\lambda]_{(j)}$ the jth diagonal block of $\mathbf{P}_\lambda$, and δ_{jk} is defined as

$$\delta_{jk} \;=\; \begin{cases} 1 & \text{if } \; j \;=\; k \\ 0 & \text{if } \; j \;\neq\; k \end{cases}$$

Here, $\dfrac{\partial \ell^*}{\partial \boldsymbol{\beta}_j}$ and $-\dfrac{\partial^2 \ell^*}{\partial \boldsymbol{\beta}_j \, \partial \boldsymbol{\beta}_k^T}$ are a $(\sum_{k=1}^{p} \mathrm{S}_k)$-vector and a $(\sum_{k=1}^{p} \mathrm{S}_k \;\times\; \sum_{k=1}^{p} \mathrm{S}_k)$ matrix respectively. If there is an intercept, the equivalent dimension of (4.23) and (4.24) is respectively given by $(1 + \sum_{k=2}^{p} \mathrm{S}_k) \times 1$ and $(1 + \sum_{k=2}^{p} \mathrm{S}_k) \times (1 + \sum_{k=2}^{p} \mathrm{S}_k)$.

In the context of simultaneous equation estimation methods, Fisher scoring is preferable to Newton-Raphson since it results in more stable computations (Yee and Wild, 1996). The Fisher scoring algorithm uses the expected rather than observed information matrix in (4.18). Equation (4.24) is replaced by

$$E\left(-\frac{\partial^2 \ell^*}{\partial \boldsymbol{\beta}_j \, \partial \boldsymbol{\beta}_k^T}\right) \;=\; \sum_{i=1}^{n} w_i \, E\left(-\frac{\partial^2 \ell_i}{\partial \eta_j \, \partial \eta_k}\right) \boldsymbol{x}_i^* \, \boldsymbol{x}_i^{*T} + [\mathbf{P}_\lambda]_{(j)} \cdot \delta_{jk}. \tag{4.25}$$

Let $\mathbf{W}$ be the "working weights matrix" made up of the diagonal blocks of $\mathbf{W}_1, \ldots, \mathbf{W}_n$, where $\mathbf{W}_i$ is an $M \times M$ matrix with (j,k)th element

$$(\mathbf{W}_i)_{jk} \;=\; -w_i \, E\left(\frac{\partial^2 \ell_i}{\partial \eta_j \, \partial \eta_k}\right).$$

Thus, the expected second derivatives of $\ell(\boldsymbol{\beta})$ become

$$\begin{aligned} E\left(-\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \, \partial \boldsymbol{\beta}^T}\right) &= \sum_{i=1}^{n} \mathbf{X}_i^T \, \mathbf{W}_i \, \mathbf{X}_i \\ &= \mathbf{X}_{\text{VAM}}^T \, \mathbf{W} \, \mathbf{X}_{\text{VAM}}, \end{aligned}$$

where $\mathbf{X}_i$ and $\mathbf{X}_{\text{VAM}}$ are respectively given by (4.8) and (4.10). The expected information matrix $\boldsymbol{\mathcal{I}}_E(\boldsymbol{\beta})$ is therefore

$$\boldsymbol{\mathcal{I}}_E(\boldsymbol{\beta}) \;=\; E\left(-\frac{\partial^2 \ell^*(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \, \partial \boldsymbol{\beta}^T}\right) \;=\; \mathbf{X}_{\text{VAM}}^T \, \mathbf{W} \, \mathbf{X}_{\text{VAM}} + \mathbf{P}_\lambda. \tag{4.26}$$

Let $\boldsymbol{u}_i$ be the vector with jth element

$$(\boldsymbol{u}_i)_j \;=\; w_i \, \frac{\partial \ell_i}{\partial \eta_j},$$

and $\boldsymbol{u} \;=\; \left(\boldsymbol{u}_1^T, \ldots, \boldsymbol{u}_n^T\right)^T$. We then obtain $\dfrac{\partial \ell}{\partial \boldsymbol{\beta}}$ in the form of

$$\begin{aligned} \frac{\partial \ell}{\partial \boldsymbol{\beta}} &= \sum_{i=1}^{n} \mathbf{X}_i^T \, \boldsymbol{u}_i \\ &= \mathbf{X}_{\text{VAM}}^T \, \boldsymbol{u}. \end{aligned}$$

Now $\boldsymbol{U}(\boldsymbol{\beta})$ from (4.19) can be written as

$$\boldsymbol{U}(\boldsymbol{\beta}) \;=\; \frac{\partial \ell^{*}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \;=\; \mathbf{X}_{\text{VAM}}^{T}\,\boldsymbol{u} - \mathbf{P}_{\lambda}\,\boldsymbol{\beta}. \tag{4.27}$$

Then the maximum log-likelihood estimates for the P-spline VGAM at iteration t can be derived as following:

$$\begin{aligned}
\boldsymbol{\beta}^{(t+1)} &= \left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1}\left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)\boldsymbol{\beta}^{(t)} \\
&\quad + \left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1}\left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\left(\mathbf{W}^{(t)}\right)^{-1}\boldsymbol{u} - \mathbf{P}_{\lambda}\,\boldsymbol{\beta}^{(t)}\right) \\
&= \left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1}\left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\,\mathbf{X}_{\text{VAM}}\,\boldsymbol{\beta}^{(t)} + \mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\left(\mathbf{W}^{(t)}\right)^{-1}\boldsymbol{u}\right) \\
&= \left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1}\left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\left(\mathbf{X}_{\text{VAM}}\,\boldsymbol{\beta}^{(t)} + \left(\mathbf{W}^{(t)}\right)^{-1}\boldsymbol{u}\right)\right) \\
&= \left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1}\left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\,\boldsymbol{z}^{(t)}\right).
\end{aligned}$$

From the expression above, our iterative procedure for maximizing $\ell^{*}(\boldsymbol{\beta})$ is given by

$$\boldsymbol{\beta}^{(t+1)} \;=\; \left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1}\left(\mathbf{X}_{\text{VAM}}^{T}\,\mathbf{W}^{(t)}\,\boldsymbol{z}^{(t)}\right). \tag{4.28}$$

Here, $\boldsymbol{z}^{(t)} \;=\; \mathbf{X}_{\text{VAM}}\,\boldsymbol{\beta}^{(t)} + \left(\mathbf{W}^{(t)}\right)^{-1}\boldsymbol{u}^{(t)}$ is an nM-vector called the adjusted dependent variable and can be partitioned as $\boldsymbol{z} \;=\; \left(\boldsymbol{z}_1^{T}, \ldots, \boldsymbol{z}_n^{T}\right)^{T}$. Given values for the smoothing parameters, $\boldsymbol{\beta}^{(t+1)}$ is the solution to the penalized iteratively reweighted least squares problem

$$\text{minimize} \qquad \left(\boldsymbol{z}^{(t)} - \mathbf{X}_{\text{VAM}}\,\boldsymbol{\beta}\right)^{T}\mathbf{W}^{(t)}\left(\boldsymbol{z}^{(t)} - \mathbf{X}_{\text{VAM}}\,\boldsymbol{\beta}\right) + \boldsymbol{\beta}^{T}\,\mathbf{P}_{\lambda}\,\boldsymbol{\beta} \tag{4.29}$$

with respect to $\boldsymbol{\beta}$. Consequently, $\boldsymbol{\eta}^{(t+1)} = \mathbf{X}_{\text{VAM}}\,\boldsymbol{\beta}^{(t+1)}$. Thus, for given values of the smoothing parameters, our penalized likelihood approach using P-spline smoothers proposed here directly fits VGAMs through a slightly modified method of scoring algorithm. All the smooth components are estimated simultaneously.

4.2.2 The geometry of penalized regression for P-spline VGAMs

In practice, we do not use (4.28) and (4.29) for the computation. We prefer the greater numerical stability offered by orthogonal-matrix methods. This can be achieved by facilitating models fitted by least squares by taking a geometric view of the fitting process. In general, the geometry of linear and generalized linear model fitting becomes more complicated when the quadratic penalties are applied to the model. This is because when penalized estimation is used, the geometry in terms of projections requires a larger space than the data space that was considered in linear and generalized linear models (cf. Wood, 2006b, Sections 1.4, 2.2, and 4.10.3,). To achieve the greater numerical stability, we will take a geometric interpretation of penalized estimation for GAMs and extend it to our proposed method. Recall from equation (4.9) that the P-spline VGAM model is written in the form of

$$\boldsymbol{\eta} \; = \; \mathbf{X}_{\text{VAM}}\, \boldsymbol{\beta},$$

and this model can be estimated by the minimization of the penalized objective function (4.29). For practical computation, we re-write (4.29) as

$$\left\| \begin{bmatrix} \mathbf{W} & 0 \\ 0 & \mathbf{I} \end{bmatrix}^{1/2} \left(\begin{bmatrix} \boldsymbol{z} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{X}_{\text{VAM}} \\ \widetilde{\mathbf{P}}_{\lambda} \end{bmatrix} \boldsymbol{\beta} \right) \right\|^{2}, \tag{4.30}$$

where $\widetilde{\mathbf{P}_{\lambda}}$ is any square root of the matrix $\mathbf{P}_{\lambda}$ such that $\widetilde{\mathbf{P}}_{\lambda}^{T}\, \widetilde{\mathbf{P}}_{\lambda} \; = \; \mathbf{P}_{\lambda}$. It can be obtained by Cholesky decomposition. Here, $\widetilde{\mathbf{P}}_{\lambda} \; = \; \text{blockdiag}\left(\widetilde{\mathbf{P}}_{\lambda 1}, \ldots, \widetilde{\mathbf{P}}_{\lambda p}\right)$ is $\{\sum_{k=1}^{p} (\mathrm{S}_k - d_k) \cdot M\} \times (\sum_{k=1}^{p} \mathrm{S}_k \cdot M)$ matrix. Each diagonal element $\widetilde{\mathbf{P}}_{\lambda k} \; = \; \mathbf{D}_{[d]k} \otimes \text{diag}\left(\sqrt{\lambda_{(1)k}}, \ldots, \sqrt{\lambda_{(M)k}}\right)$ is a $\{(\mathrm{S}_k - d_k) \cdot M\} \times (\mathrm{S}_k \cdot M)$ matrix. The expression in (4.30) is simply the un-penalized GLS objective function for an augmented version of the P-spline VGAM models. So the sum of squares term in (4.30) is a GLS objective for a model in which the model matrix has been augmented by a square root of the penalty matrix, while the adjusted dependent vector has been augmented with $\sum_{k=1}^{p} (\mathrm{S}_k - d_k) \cdot M$ zeros. The working weights matrix has been augmented by a $\left\{\sum_{k=1}^{p} (\mathrm{S}_k - d_k) \cdot M\right\} \times nM$ zero matrix, and added an additional column of a 2×1 block matrix, where the first and second blocks are a $\left\{\sum_{k=1}^{p}(\mathrm{S}_k - d_k) \cdot M\right\} \times \left\{\sum_{k=1}^{p}(\mathrm{S}_k - d_k) \cdot M\right\}$ zero matrix, and a $\left\{\sum_{k=1}^{p}(\mathrm{S}_k - d_k) \cdot M\right\} \times \left\{\sum_{k=1}^{p}(\mathrm{S}_k - d_k) \cdot M\right\}$ identity matrix respectively. The augmented working weights matrix, $\mathbf{W}^{'}$ can be simply written as $\mathbf{W}^{'} = \operatorname{blockdiag}(\mathbf{W}, \mathbf{I}_{\vartheta})$, where $\vartheta = \sum_{k=1}^{p}(\mathrm{S}_k - d_k) \cdot M$. The augmented $\boldsymbol{z}$, $\mathbf{X}_{\text{VAM}}$, and $\mathbf{W}$, can be written as follows:

$$\boldsymbol{z}^{'} = \begin{pmatrix} \boldsymbol{z} \\ \mathbf{0}_{\vartheta} \end{pmatrix}, \qquad \mathbf{X}^{'}_{\text{VAM}} = \begin{pmatrix} \mathbf{X}_{\text{VAM}} \\ \widetilde{\mathbf{P}}_{\lambda} \end{pmatrix}, \qquad \mathbf{W}^{'} = \operatorname{blockdiag}(\mathbf{W}, \mathbf{I}_{\vartheta}). \tag{4.31}$$

Then, the equivalent minimization becomes

$$\text{minimize} \qquad \left(\boldsymbol{z}^{'} - \mathbf{X}^{'}_{\text{VAM}}\boldsymbol{\beta}\right)^{T} \mathbf{W}^{'} \left(\boldsymbol{z}^{'} - \mathbf{X}^{'}_{\text{VAM}}\boldsymbol{\beta}\right) \tag{4.32}$$

with respect to $\boldsymbol{\beta}$. In (4.32), the estimates of $\boldsymbol{\beta}$ can be obtained by solving the GLS problem,

$$\boldsymbol{z}^{'} = \mathbf{X}^{'}_{\text{VAM}}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{4.33}$$

where $\operatorname{Var}(\boldsymbol{\varepsilon}) = \operatorname{blockdiag}\left(\mathbf{W}^{-1}, \mathbf{I}_{\vartheta}\right) = \mathbf{W}^{'-1}$. The standard method based on a Cholesky decomposition is applied in order to convert the GLS system of equations to ordinary least squares (OLS). We premultiply both sides of the regression equation of (4.33) by a Cholesky decomposition of the $\mathbf{W}^{'}$. Importantly, $\mathbf{U}^{'}$ is a matrix square root such that

$$\mathbf{U}^{'T}\mathbf{U}^{'} = \mathbf{W}^{'} = \operatorname{blockdiag}\left(\mathbf{U}_1^{T}\mathbf{U}_1, \ldots, \mathbf{U}_n^{T}\mathbf{U}_n, \mathbf{I}_{\vartheta}\right), \tag{4.34}$$

where $\mathbf{U}_i^{T}\mathbf{U}_i = \mathbf{W}_i$, and the matrices $\mathbf{U}_i^{'}$ can be obtained by the Cholesky decomposition. By premultiplying (4.33) by $\mathbf{U}^{'}$, we obtain a new regression equation

$$\boldsymbol{z}^{''} = \mathbf{X}^{''}_{\text{VAM}}\boldsymbol{\beta} + \boldsymbol{\varepsilon}^{''}, \tag{4.35}$$

where $\boldsymbol{z}'' = \mathbf{U}^{'}\boldsymbol{z}^{'}$, $\mathbf{X}^{''}_{\text{VAM}} = \mathbf{U}^{'}\mathbf{X}^{'}_{\text{VAM}}$, $\boldsymbol{\varepsilon}^{''} = \mathbf{U}^{'}\boldsymbol{\varepsilon}^{'}$ and $\operatorname{Var}\left(\boldsymbol{\varepsilon}^{''}\right) = \sigma^2\mathbf{I}_{\left\{nM + \sum_{k=1}^{p}(\mathrm{S}_k - d_k)\cdot M\right\}}$. The augmented GLS problem (4.32) is now reduced to the OLS normal equations (4.35).

Given smoothing parameters, the maximum penalized log-likelihood estimates of $\boldsymbol{\beta}$ at iteration t are obtained by the following steps. Given a current coefficient vector $\boldsymbol{\beta}^{(t)}$, with corresponding linear predictor $\boldsymbol{\eta}^{(t)}$, construct the adjusted dependent variable $\boldsymbol{z}^{(t)}$ and working weights $\mathbf{W}^{(t)}$. Data augmentation is applied to obtain $\mathbf{X}^{'}_{\text{VAM}}$, $\boldsymbol{z}^{'(t)}$, and $\mathbf{W}^{'(t)}$. The algorithm proceeds by regressing $\boldsymbol{z}^{'(t)}$ on $\mathbf{X}^{'}_{\text{VAM}}$ with weights $\mathbf{W}^{'(t)}$ to obtain an improved estimate $\boldsymbol{\beta}^{(t+1)}$ using (4.35). Then, a new $\boldsymbol{\eta}^{(t+1)}$, $\boldsymbol{z}^{(t+1)}$, and $\mathbf{W}^{(t+1)}$ are computed, a new $\boldsymbol{z}^{'(t+1)}$, and $\mathbf{W}^{'(t+1)}$ are constructed, and the process is repeated, until the change in the coefficients is sufficiently small. This procedure forms the P-IRLS algorithm. Hence, given smoothing parameters, we maximize the P-spline VGAM penalized log-likelihood (4.17) by P-IRLS. For the moment, the smoothing parameters are taken as known. In Chapter 5, smoothing parameter estimation will be performed by minimizing the GCV or the UBRE score with respect to the smoothing parameters.

The OLS procedure (4.35) is the simplest type of estimation procedure used in statistical analyzes with its underlying computation being based on orthogonal methods which provide us with the greater numerical stability with respect to numerical problems due to ill-conditioned design matrices. As stated by Yee (2015b, chapter 3), to solve this OLS problem, the QR algorithm can be operated using modified LINPACK subroutines to give stable ordering and rank estimation. For a large $n \times m$ matrix, the QR decomposition costs approximately $2nm^2$ floating point operations (flops). Therefore, formulation of a major component of fitting P-spline VGAMs with *trivial* constraints costs about $2\left(nM + \sum_{k=1}^{p}(\mathrm{S}_k - d_k)\cdot M\right) M^2 \left(\sum_{k=1}^{p}\mathrm{S}_k\right)^2 \approx 2nM^3p^2\overline{\mathrm{S}}_k^2$ flops at each P-IRLS iteration for $\mathbf{X}^{''(t-1)}_{\text{VAM}}$. Also, the storage demand of $\mathbf{X}_{\text{VAM}}$ with dimensions $\left(nM + \sum_{k=1}^{p}(\mathrm{S}_k - d_k)\cdot M\right) \times \sum_{k=1}^{p}\mathrm{S}_k \cdot M$ involves approximately $nM^2p\overline{\mathrm{S}}_k$. These indicate that the computational load and storage requirements for fitting P-spline VGAMs rise rapidly with respect to M, followed by p, S_k, and then n. Following Yee (2015b, chapter 3), the storage costs of the algorithm proposed can be reduced by reducing the number of parameters, such as, imposing the constraints on the functions and reducing the number of knots.

4.3 P-spline VGAMs with constraint matrices

The definition of $\mathbf{X}_{\text{VAM}}$ in (4.10) can only accommodate "trivial constraints" ($\mathbf{H}_k = \mathbf{I}_M$ for all k). We saw in Section 3.3 that facility called "constraints on the functions" is one key idea allowing the VGAM approach to be much more useful in many more situations. In this section, we will generalize these ideas to the P-spline VGAM framework.

4.3.1 Setting up the constrained P-spline VGAMs as penalized VGLMs

Recall from (4.1) that $\eta_j(\boldsymbol{x}) = \sum_{k=1}^{p} f_{(j)k}(x_k)$. Then $\boldsymbol{\eta} = (\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_n)^T$ is given by $\boldsymbol{\eta} = \boldsymbol{f}_1(x_1) + \cdots + \boldsymbol{f}_p(x_p)$. We will use constraint matrices to constrain the behavior of $\eta_j(\boldsymbol{x})$ term (cf. Yee and Wild (1996)). The approach to handling constraints on the model terms can be written as

$$\begin{aligned}\boldsymbol{\eta} &= \boldsymbol{f}_1(x_1) + \cdots + \boldsymbol{f}_p(x_p) \\ &= \mathbf{H}_1\, \boldsymbol{f}_1^*(x_1) + \cdots + \mathbf{H}_p\, \boldsymbol{f}_p^*(x_p), \end{aligned} \tag{4.36}$$

where $\boldsymbol{f}_k^* = \left(f^*_{(1)k}(x_k), \ldots, f^*_{(\mathrm{R}_k)k}(x_k)\right)^T$, $k = 1, \ldots, p$, is a vector consisting of a possibly reduced set of smooth functions to be estimated, and each $\mathbf{H}_k$ is a "constraint matrix". Recall from Section 3.3 that the constraint matrices are known full column-rank matrices with the dimension $M \times \mathrm{R}_k$. The starred quantities in (4.36) are unknown and have to be estimated. When no constraints are imposed on a set of smooth functions, then the relevant constraint matrix is again an $M \times M$ identity matrix, so that $\boldsymbol{f}_k^* = \boldsymbol{f}_k = \left(f_{(1)k}(x_k), \ldots, f_{(M)k}(x_k)\right)^T$. In our implementation, each smooth component in $\boldsymbol{f}_k^*$ will share the same set of knots in the way we described for $\boldsymbol{f}_k$ in Section 4.1. The vector containing all smooth terms at the jth smooth predictor $\boldsymbol{f}_{(j)}$ is defined and subjected to centering constraints in the same manner as described in Section 4.1.

We now write the smooth predictor vector $\boldsymbol{\eta}$ in terms of the observations as follows:

$$\begin{aligned}
\boldsymbol{\eta}_i &= \mathbf{H}_1\, \boldsymbol{f}_1^*(x_{i1}) + \cdots + \mathbf{H}_p\, \boldsymbol{f}_p^*(x_{ip}) \\
&= \mathbf{H}_1 \begin{pmatrix} a_{(1)1:1} & \cdots & a_{(1)1:\mathrm{S}_1} \\ \vdots & & \vdots \\ a_{(\mathrm{R}_1)1:1} & \cdots & a_{(\mathrm{R}_1)1:\mathrm{S}_1} \end{pmatrix} \boldsymbol{x}_{i1}^* + \cdots + \mathbf{H}_p \begin{pmatrix} a_{(1)p:1} & \cdots & a_{(1)p:\mathrm{S}_p} \\ \vdots & & \vdots \\ a_{(\mathrm{R}_p)p:1} & \cdots & a_{(\mathrm{R}_p)p:\mathrm{S}_p} \end{pmatrix} \boldsymbol{x}_{ip}^* \\
&= \mathbf{H}_1 \left(\boldsymbol{\beta}_{1:1}^* \cdots \boldsymbol{\beta}_{1:\mathrm{S}_1}^*\right) \boldsymbol{x}_{i1}^* + \cdots + \mathbf{H}_p \left(\boldsymbol{\beta}_{p:1}^* \cdots \boldsymbol{\beta}_{p:\mathrm{S}_p}^*\right) \boldsymbol{x}_{ip}^*, \qquad (4.37)
\end{aligned}$$

where $\boldsymbol{\beta}_{k:s}^* = \left(a_{(1)k:s}, \ldots, a_{(\mathrm{R}_k)k:s}\right)^T$, $k = 1, \ldots, p$ and $s = 1, \ldots, \mathrm{S}_k$. By applying 'vectorization' (cf. equation (4.6)) to the matrix $\left(\boldsymbol{\beta}_{k:1}^* \ldots \boldsymbol{\beta}_{k:\mathrm{S}_k}^*\right)$, we obtain a $(\mathrm{S}_k \cdot \mathrm{R}_k) \times 1$ vector containing a possibly reduced set of B-spline coefficients $\boldsymbol{\beta}_k^*$ as follows:

$$\begin{aligned}
\boldsymbol{\beta}_k^* &= \mathrm{vec}\left(\boldsymbol{\beta}_{k:1}^* \ldots \boldsymbol{\beta}_{k:\mathrm{S}_k}^*\right) \\
&= \mathrm{vec} \begin{pmatrix} a_{(1)k:1} & \cdots & a_{(1)k:\mathrm{S}_k} \\ \vdots & & \vdots \\ a_{(\mathrm{R}_k)k:1} & \cdots & a_{(\mathrm{R}_k)k:\mathrm{S}_k} \end{pmatrix} \\
&= \mathrm{vec} \begin{pmatrix} \boldsymbol{\beta}_{(1)k}^{*T} \\ \vdots \\ \boldsymbol{\beta}_{(\mathrm{R}_k)k}^{*T} \end{pmatrix}, \qquad (4.38)
\end{aligned}$$

where $\boldsymbol{\beta}_{(j)k}^* = \left(a_{(j)k:1}, \ldots, a_{(j)k:\mathrm{S}_k}\right)^T$, $j = 1, \ldots, \mathrm{R}_k$. Therefore, equation (A.2.2) is equivalent to

$$\begin{aligned}
\boldsymbol{\eta}_i &= \mathbf{H}_1 \left(\boldsymbol{x}_{i1}^{*T} \otimes \mathbf{I}_{\mathrm{R}_1}\right) \boldsymbol{\beta}_1^* + \cdots + \mathbf{H}_p \left(\boldsymbol{x}_{ip}^{*T} \otimes \mathbf{I}_{\mathrm{R}_p}\right) \boldsymbol{\beta}_p^* \\
&= \mathbf{H}_1\, \mathbf{X}_{i1}^*\, \boldsymbol{\beta}_1^* + \cdots + \mathbf{H}_p\, \mathbf{X}_{ip}^*\, \boldsymbol{\beta}_p^* \\
&= \sum_{k=1}^{p} \mathbf{H}_k\, \mathbf{X}_{ik}^*\, \boldsymbol{\beta}_k^*.
\end{aligned}$$

Here, $\mathbf{X}_{ik}^* = \boldsymbol{x}_{ik}^{*T} \otimes \mathbf{I}_{\mathrm{R}_k}$ is an $\mathrm{R}_k \times (\mathrm{S}_k \cdot \mathrm{R}_k)$ matrix. Next, we will illustrate how the constraint ideas can be accommodated within the P-spline VGAM framework. All the essential points are illustrated by considering the bivariate logistic models (cf. Section 3.3.2).

1. **Bivariate logistic models**

As an example, let us revisit the example of two binary responses for the presence or absence of cataracts in elderly patients's eyes, where y_1 and y_2 are respectively the the presence or absence of cataracts for the left and right eye. We will now illustrate the P-spline VGAM analog of the VGLM in Section 3.3.2:

$$\begin{aligned} \operatorname{logit} \boldsymbol{p}_j(\boldsymbol{x}) &= \eta_1(\boldsymbol{x}), \qquad j = 1,2, \\ \log \psi(\boldsymbol{x}) &= \eta_3(\boldsymbol{x}) = \beta^*_{(2)1}. \end{aligned} \tag{4.39}$$

The smooth-function equivalent of (3.28) is

$$\begin{aligned} \eta_1(\boldsymbol{x}_i) &= \beta^*_{(1)1} + f^*_{(1)2}(x_{i2}) + f^*_{(1)3}(x_{i3}), \\ \eta_2(\boldsymbol{x}_i) &= \beta^*_{(1)1} + f^*_{(1)2}(x_{i2}) + f^*_{(1)3}(x_{i3}), \\ \eta_3(\boldsymbol{x}_i) &= \beta^*_{(2)1}. \end{aligned} \tag{4.40}$$

Here,

$$\begin{aligned} \boldsymbol{\eta}(\boldsymbol{x}_i) &= \begin{pmatrix} \eta_1(\boldsymbol{x}_i) \\ \eta_2(\boldsymbol{x}_i) \\ \eta_3(\boldsymbol{x}_i) \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \beta^*_{(1)1} \\ \beta^*_{(2)1} \end{pmatrix} x_{i1} + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} f^*_{(1)2}(x_{i2}) + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} f^*_{(1)3}(x_{i3}) \\ &= \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{X}^*_{i1}\, \boldsymbol{\beta}^*_1 + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \mathbf{X}^*_{i2}\, \boldsymbol{\beta}^*_2 + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \mathbf{X}^*_{i3}\, \boldsymbol{\beta}^*_3 \\ &= \sum_{k=1}^{3} \mathbf{H}_k\, \mathbf{X}^*_{ik}\, \boldsymbol{\beta}^*_k. \end{aligned} \tag{4.41}$$

We have the same set of constraint matrices $\mathbf{H}_1$, $\mathbf{H}_2$, and, $\mathbf{H}_3$

$$\mathbf{H}_1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \ \mathbf{H}_2 = \mathbf{H}_3 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

The vectors $\boldsymbol{\beta}_k^*$, $k = 1, 2, 3$, (cf. equation (4.38)) are as follows:

$$\boldsymbol{\beta}_1^* = \left(\beta_{(1)1}, \ \beta_{(2)1}\right)^T, \qquad \boldsymbol{\beta}_2^* = \operatorname{vec}\left(a_{(1)2:1} \cdots a_{(1)2:\mathrm{S}_2}\right), \text{ and}$$

$$\boldsymbol{\beta}_3^* = \operatorname{vec}\left(a_{(1)3:1} \cdots a_{(1)3:\mathrm{S}_3}\right).$$

Equation (4.41) illustrates how constraint matrices can be applied directly in the P-spline VGAM framework with the general form of

$$\boldsymbol{\eta}(\boldsymbol{x}_i) = \sum_{k=1}^{p} \mathbf{H}_k \, \mathbf{X}_{ik}^* \, \boldsymbol{\beta}_k^*. \tag{4.42}$$

We now develop the computational theory for the general constrained problem (4.42). Let $\boldsymbol{\beta}^* = \left(\boldsymbol{\beta}_1^{*T}, \ldots, \boldsymbol{\beta}_p^{*T}\right)^T$ be a $\left(\sum_{k=1}^{p} \mathrm{S}_k \cdot \mathrm{R}_k\right) \times 1$ vector containing all of the possibly reduced sets of B-spline coefficients in the models, and

$$\mathbf{X}_{\text{VAM}} = \left(\left(\mathbf{X}_{\text{AM}} \, \widetilde{\mathbf{E}}_1\right) \otimes \mathbf{H}_1 \,\middle|\, \left(\mathbf{X}_{\text{AM}} \, \widetilde{\mathbf{E}}_2\right) \otimes \mathbf{H}_2 \,\middle|\, \cdots \,\middle|\, \left(\mathbf{X}_{\text{AM}} \, \widetilde{\mathbf{E}}_p\right) \otimes \mathbf{H}_p\right) \tag{4.43}$$

be an $(nM) \times p^{**}$ model matrix (cf. $\mathbf{X}_{\text{VAM}}$ for the unconstrained problem in equation (4.10)). Here, $p^{**} = \sum_{k=1}^{p} \mathrm{S}_k \cdot \operatorname{ncol}(\mathbf{H}_k)$ and the matrix $\widetilde{\mathbf{E}}_k$'s are $\left(\sum_{k=1}^{p} \mathrm{S}_k\right) \times \mathrm{S}_k$ matrices defined as follows:

$$\widetilde{\mathbf{E}}_1 = \begin{pmatrix} \mathbf{I}_{(\mathrm{S}_1 \times \mathrm{S}_1)} \\ \mathbf{O} \\ \vdots \\ \mathbf{O} \end{pmatrix}, \qquad \widetilde{\mathbf{E}}_2 = \begin{pmatrix} \mathbf{O} \\ \mathbf{I}_{(\mathrm{S}_2 \times \mathrm{S}_2)} \\ \mathbf{O} \\ \vdots \\ \mathbf{O} \end{pmatrix}, \qquad \ldots,$$

$$\widetilde{\mathbf{E}}_p = \begin{pmatrix} \mathbf{O} \\ \vdots \\ \mathbf{O} \\ \mathbf{I}_{(\mathrm{S}_p \times \mathrm{S}_p)} \end{pmatrix}.$$

Analogous to $\boldsymbol{e}_k$ in the "vector linear model" model-matrix settings for the constrained VGLMs, we can simply write $\widetilde{\mathbf{E}}_k = \boldsymbol{e}_k \otimes \mathbf{I}_{\mathrm{S}_k}$ where $\boldsymbol{e}_k$ is a zero vector with a one in the kth position (cf. equation (3.30)).

The class of P-spline VGAMs subjected to the constraint matrices can be then written as

$$\boldsymbol{\eta} \;=\; \mathbf{X}_{\text{VAM}}\, \boldsymbol{\beta}^*, \tag{4.44}$$

which is the same form as (4.9) for the unconstrained problem, but with a redefined $\mathbf{X}_{\text{VAM}}$ matrix. The expression in (4.44) has the same form as a VGLM.

4.3.1.1 The penalty for the constrained P-spline VGAMs

We now turn our attention to controlling the model's smoothness by adding wiggliness penalties to the log-likelihood objective $\ell\left(\boldsymbol{\beta}^*\right)$ of (4.44) (cf. equation (4.11)). The wiggliness penalty for the constrained P-spline VGAMs can be constructed in the same way as we constructed the wiggliness penalty for the unconstrained P-spline VGAMs (cf. Section 4.1.2). Let $\boldsymbol{\lambda}_k^* = \left(\lambda_{(1)k}, \ldots, \lambda_{(\mathrm{R}_k)k}\right)^T$ be a $\mathrm{R}_k \times 1$ vector containing a possibly reduced set of smoothing parameters. We, therefore, write the penalty for the constrained P-spline VGAMs as follows:

$$\begin{aligned} J(\boldsymbol{\lambda}) \;&=\; \sum_{k=1}^{p} \sum_{j=1}^{\mathrm{R}_k} \lambda_{(j)k}\, \boldsymbol{\beta}^{*T}_{(j)k}\, \mathbf{D}^T_{[d]k}\, \mathbf{D}_{[d]k}\, \boldsymbol{\beta}^*_{(j)k} \\ &=\; \sum_{k=1}^{p} \boldsymbol{\beta}^{*T}_{k} \left\{ \left(\mathbf{D}^T_{[d]k}\, \mathbf{D}_{[d]k}\right) \otimes \operatorname{diag}\left(\lambda_{(1)k}, \ldots, \lambda_{(\mathrm{R}_k)k}\right) \right\} \boldsymbol{\beta}^*_k \\ &=\; \sum_{k=1}^{p} \boldsymbol{\beta}^{*T}_{k}\, \mathbf{P}^*_{\lambda k}\, \boldsymbol{\beta}^*_k, \end{aligned} \tag{4.45}$$

where $\mathbf{P}^*_{\lambda k} = \left(\mathbf{D}^T_{[d]k}\, \mathbf{D}_{[d]k}\right) \otimes \operatorname{diag}\left(\lambda_{(1)k}, \ldots, \lambda_{(\mathrm{R}_k)k}\right)$ is a $(\mathrm{S}_k \cdot \mathrm{R}_k) \times (\mathrm{S}_k \cdot \mathrm{R}_k)$ matrix. As before, the penalty matrix $\mathbf{P}^*_\lambda$ is

$$\mathbf{P}^*_\lambda \;=\; \operatorname{blockdiag}\left(\mathbf{P}^*_{\lambda 1}, \ldots, \mathbf{P}^*_{\lambda p}\right), \tag{4.46}$$

which has the dimensions $\left(\sum_{k=1}^{p} \mathrm{S}_k \cdot \mathrm{R}_k\right) \times \left(\sum_{k=1}^{p} \mathrm{S}_k \cdot \mathrm{R}_k\right)$. Given $\mathbf{P}^*_\lambda$, we re-write (4.45) as

$$J(\boldsymbol{\lambda}) \;=\; \boldsymbol{\beta}^{*T}\, \mathbf{P}^*_\lambda\, \boldsymbol{\beta}^*.$$

Given a wiggliness measure for each smooth function, the penalized log-likelihood for the constrained P-spline VGAMs is

$$\ell^*(\boldsymbol{\beta}^*) \;=\; \ell(\boldsymbol{\beta}^*) - \frac{1}{2}\sum_{k=1}^{p}\sum_{j=1}^{\mathrm{R}_k} \lambda_{(j)k}\, \boldsymbol{\beta}^{*T}_{(j)k}\, \mathbf{D}^T_{[d]k}\, \mathbf{D}_{[d]k}\, \boldsymbol{\beta}^*_{(j)k}. \tag{4.47}$$

We re-write the penalized log-likelihood objective (4.47) in matrix notation as

$$\ell^*(\boldsymbol{\beta}^*) \;=\; \ell(\boldsymbol{\beta}^*) - \frac{1}{2}\,\boldsymbol{\beta}^{*T}\,\mathbf{P}^*_{\lambda}\,\boldsymbol{\beta}^*. \tag{4.48}$$

4.3.2 Estimating $\boldsymbol{\beta}^*$ for given smoothing parameters

Given values for the $\lambda_{(j)k}$, the coefficient estimates $\widehat{\boldsymbol{\beta}}^*$ can be obtained by maximizing the $\ell^*\left(\boldsymbol{\beta}^*\right)$. We estimate $\ell^*(\boldsymbol{\beta}^*)$ in (4.48) in exactly the same way as in the unconstrained P-spline VGAM case (cf. Sections 4.2 and 4.2.2). The only change is a more complicated model matrix $\mathbf{X}_{\text{VAM}}$, parameter vector $\boldsymbol{\beta}^*$, and penalty matrix $\mathbf{P}^*_{\lambda}$. Following Section 4.2, the penalized maximum likelihood estimation of (4.48) is therefore performed by repeated solution of

$$\text{minimize} \qquad \sum_{i=1}^{n}\left(\boldsymbol{z}_i - \sum_{k=1}^{p}\mathbf{H}_k\,\mathbf{X}^*_{ik}\,\boldsymbol{\beta}^*_k\right)^T \mathbf{W}_i \left(\boldsymbol{z}_i - \sum_{k=1}^{p}\mathbf{H}_k\,\mathbf{X}^*_{ik}\,\boldsymbol{\beta}^*_k\right) + \sum_{k=1}^{p}\boldsymbol{\beta}^{*T}_k\,\mathbf{P}^*_{\lambda k}\,\boldsymbol{\beta}^*_k,$$

which is equivalent to

$$\text{minimize} \qquad \left(\boldsymbol{z} - \mathbf{X}_{\text{VAM}}\,\boldsymbol{\beta}^*\right)^T \mathbf{W}\left(\boldsymbol{z} - \mathbf{X}_{\text{VAM}}\,\boldsymbol{\beta}^*\right) + \boldsymbol{\beta}^{*T}\,\mathbf{P}^*_{\lambda}\,\boldsymbol{\beta}^* \tag{4.49}$$

with respect to $\boldsymbol{\beta}^*$. The updated adjusted dependent vector and a weight matrix are constructed in the same way as we demonstrated in Section 4.2. Following Section 4.2.2, data augmentation is applied to (4.49) in order to achieve penalization. This gives the minimization of

$$\left(\boldsymbol{z}' - \mathbf{X}'_{\text{VAM}}\,\boldsymbol{\beta}^*\right)^T \mathbf{W}' \left(\boldsymbol{z}' - \mathbf{X}'_{\text{VAM}}\,\boldsymbol{\beta}^*\right)$$

with respect to $\boldsymbol{\beta}^*$. Hence $\boldsymbol{\beta}^*$ is the solution to the GLS problem

$$\boldsymbol{z}' \;=\; \mathbf{X}'_{\text{VAM}}\,\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}'. \tag{4.50}$$

We note that the relevant minimization problem is again the GLS problem (cf. the minimization problem for the unconstrained problem in Section 4.2.2) but with a model matrix $\mathbf{X}_{\text{VAM}}$ and penalty matrix $\mathbf{P}_{\lambda}^{*}$ given by (4.43) and (4.46) respectively (cf. equation (4.31)). We then convert the GLS problem to OLS in the same manner as we did with the unconstrained case by premultiplying both sides of (4.50) by $\mathbf{U}^{'}$, where $\mathbf{U}^{'}$ is any matrix square root such that $\mathbf{U}^{'T}\mathbf{U}^{'} = \mathbf{W}^{'}$ (cf. equation (4.34)). This gives the OLS equation (cf. equation (4.35))

$$\boldsymbol{z}^{''} = \mathbf{X}_{\text{VAM}}^{''}\boldsymbol{\beta}^{*} + \boldsymbol{\varepsilon}^{''}, \tag{4.51}$$

where $\text{Var}\left(\boldsymbol{\varepsilon}^{''}\right) = \sigma^{2}\mathbf{I}_{\left\{nM+\sum_{k=1}^{p}(\text{S}_k-d_k)\cdot\text{R}_k\right\}}$. Hence given smoothing parameters, the maximum penalized likelihood estimates for $\boldsymbol{\beta}^{*}$ at iteration t are again obtained by regressing $\boldsymbol{z}^{'(t)}$ upon $\mathbf{X}_{\text{VAM}}^{'}$ with $\mathbf{W}^{'(t)}$ using (4.51) to obtain an updated $\boldsymbol{\beta}^{*(t+1)}$ until specified convergence, cf. the P-IRLS procedure for the unconstrained problem in Section 4.2. The entire procedure for fitting P-spline VGAMs is summarized in algorithm 1 to follow.

Algorithm 1 The P-spline VGAM estimation procedure: prescribed smoothing parameters with constraints

```
1. Construct:

   a) $\mathbf{X}_{\text{VAM}}$ from $\mathbf{X}_{\text{AM}}$ and $\mathbf{H}_1, \ldots, \mathbf{H}_p$ :

      $\mathbf{X}_{\text{VAM}} = \left( \left( \mathbf{X}_{\text{AM}} \widetilde{\mathbf{E}}_1 \right) \otimes \mathbf{H}_1 \middle| \left( \mathbf{X}_{\text{AM}} \widetilde{\mathbf{E}}_2 \right) \otimes \mathbf{H}_2 \middle| \cdots \middle| \left( \mathbf{X}_{\text{AM}} \widetilde{\mathbf{E}}_p \right) \otimes \mathbf{H}_p \right)$ (cf. equation (4.43)),

   b) $\mathbf{P}^*_\lambda = \text{blockdiag}\left( \mathbf{P}^*_{\lambda 1}, \ldots, \mathbf{P}^*_{\lambda p} \right)$, where $\mathbf{P}^*_{\lambda k} = \left( \mathbf{D}^T_{[d]k} \mathbf{D}_{[d]k} \right) \otimes \text{diag}\left( \lambda_{(1)k}, \ldots, \lambda_{(\text{R}_k)k} \right)$
      (cf. equation (4.46)).

2. Initialize: $\boldsymbol{\eta}^{(0)}$, e.g., from $\boldsymbol{\beta}^{*(0)}$ or $\boldsymbol{\mu}^{(0)}$ (if necessary, compute $\boldsymbol{\mu}^{(0)}$, $\ell^0$, define a $\boldsymbol{\beta}^{*(0)}$,
   etc).

3. Compute $\boldsymbol{u}^{(0)}$ and the working weights $\mathbf{W}^{(0)}$.

4. Compute an adjusted dependent variable from $\boldsymbol{z}^{(0)} = \boldsymbol{\eta}^{(0)} + \left( \mathbf{W}^{(0)} \right)^{-1} \boldsymbol{u}^{(0)}$.

5. Construct the augmented $\mathbf{X}^{'}_{\text{VAM}} = \begin{pmatrix} \mathbf{X}_{\text{VAM}} \\ \widetilde{\mathbf{P}}^*_\lambda \end{pmatrix}$, where $\widetilde{\mathbf{P}}^{*T}_\lambda \widetilde{\mathbf{P}}^*_\lambda = \mathbf{P}^*_\lambda$ (cf. equation (4.31)).

6. For $t = 0, 1, 2, \ldots$

   a) Construct the augmented $\boldsymbol{z}^{(t)}$ and $\mathbf{W}^{(t)}$ (cf. equation (4.31)):

      $\boldsymbol{z}^{'(t)} = \begin{pmatrix} \boldsymbol{z}^{(t)} \\ \mathbf{0}_\vartheta \end{pmatrix}, \qquad \mathbf{W}^{'(t)} = \text{blockdiag}\left( \mathbf{W}^{(t)}, \mathbf{I}_\vartheta \right).$

   b) Regress $\boldsymbol{z}^{'(t)}$ upon $\mathbf{X}^{'}_{\text{VAM}}$ with weights $\mathbf{W}^{'(t)}$ to obtain estimated $\widehat{\boldsymbol{\beta}}^{*(t+1)}$ using

      $\boldsymbol{z}^{''(t)} = \mathbf{X}^{''(t)}_{\text{VAM}} \boldsymbol{\beta}^* + \boldsymbol{\varepsilon}^{''(t)}$ (cf. equation (4.51)).

   c) Evaluate the smooth predictor $\boldsymbol{\eta}^{(t+1)}$.

   d) Compute the convergence criterion

      $\Delta \left( \boldsymbol{\beta}^{*(t+1)}, \boldsymbol{\beta}^{*(t)} \right) = \left\| \boldsymbol{\beta}^{*(t+1)} - \boldsymbol{\beta}^{*(t)} \right\|.$

   e) Compute $\boldsymbol{u}^{(t+1)}$ and the working weights $\mathbf{W}^{(t+1)}$.

   f) Compute the adjusted dependent variable $\boldsymbol{z}^{(t+1)} = \boldsymbol{\eta}^{(t+1)} + \left( \mathbf{W}^{(t+1)} \right)^{-1} \boldsymbol{u}^{(t+1)}$.

7. Repeat step 6, replacing $\boldsymbol{\eta}^{(t)}$ by $\boldsymbol{\eta}^{(t+1)}$ until $\Delta \left( \boldsymbol{\beta}^{*(t+1)}, \boldsymbol{\beta}^{*(t)} \right)$ is sufficiently small.
```

4.4 Degrees of freedom for P-spline VGAMs

As mentioned in Section 2.2.1, the effective degrees of freedom (EDF), defined as $\mathrm{tr}(\mathbf{A})$ is one method of measuring the flexibility of the fitted model of GAMs based on penalized regression splines. The question is how to measure the flexibility of the fitted model of P-spline VGAMs and how many degrees of freedom do the fitted P-spline VGAMs have? Since P-spline VGAMs are developed under the GAMs based on the penalized regression splines, the influence matrix (or hat matrix) $\mathbf{A}_\lambda$ is naturally evaluated from the fitted model, and the flexibility of the fitted model for P-spine VGAMs can be measured using $\mathrm{tr}(\mathbf{A}_\lambda)$, in the same way that it is evaluated in GAMs based on the penalized likelihood-based approach (cf. Wood (Section 4.4, 2006b)).

In our computation, data augmentation is applied to convert the penalized iteratively reweighted least squares problem in (4.29) (for the unconstrained P-spline VGAMs) to the GLS problem in (4.32), and a Cholesky decomposition of the working weights $\mathbf{W}$ is then used to convert (4.32) to the OLS problem in (4.35). This gives the final expression of the parameter estimates $\widehat{\boldsymbol{\beta}}$ as

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}^{''T}_{\text{VAM}}\,\mathbf{X}^{''}_{\text{VAM}}\right)^{-1}\left(\mathbf{X}^{''T}_{\text{VAM}}\,\boldsymbol{z}^{''}\right). \tag{4.52}$$

Given (4.52), the augmented influence-matrix can be written:

$$\mathbf{X}^{''}_{\text{VAM}}\left(\mathbf{X}^{''T}_{\text{VAM}}\,\mathbf{X}^{''}_{\text{VAM}}\right)^{-1}\mathbf{X}^{''T}_{\text{VAM}}. \tag{4.53}$$

Recall from equation (4.35) that $\mathbf{X}^{''}_{\text{VAM}} = \mathbf{U}^{'}\,\mathbf{X}^{'}_{\text{VAM}}$, where $\mathbf{U}^{'}$ is any square root of the matrix $\mathbf{W}^{'}$ such that $\mathbf{U}^{'T}\mathbf{U}^{'} = \mathbf{W}^{'}$, and can be written as

$$\mathbf{U}^{'} = \begin{pmatrix}\mathbf{U} & 0\\ 0 & \mathbf{I}_\vartheta\end{pmatrix}, \qquad \text{and} \qquad \mathbf{X}^{'}_{\text{VAM}} = \begin{pmatrix}\mathbf{X}_{\text{VAM}}\\ \widetilde{\mathbf{P}}_\lambda\end{pmatrix}$$

(cf. equations (4.31) and (4.34)), so that

$$\mathbf{X}^{''}_{\text{VAM}} = \mathbf{U}^{'}\begin{pmatrix}\mathbf{X}_{\text{VAM}}\\ \widetilde{\mathbf{P}}_\lambda\end{pmatrix} = \begin{pmatrix}\mathbf{U}\,\mathbf{X}_{\text{VAM}}\\ \widetilde{\mathbf{P}}_\lambda\end{pmatrix}.$$

We will now see that $\mathrm{tr}(\mathbf{A}_\lambda)$ can be obtained from the sum of the first $n \cdot M$ elements on the leading diagonal of (4.53).

$$
\begin{aligned}
&\mathbf{X}''_{\text{VAM}} \left(\mathbf{X}''^{T}_{\text{VAM}} \mathbf{X}''_{\text{VAM}}\right)^{-1} \mathbf{X}''^{T}_{\text{VAM}} \\
&= \begin{pmatrix} \mathbf{U}\,\mathbf{X}_{\text{VAM}} \\ \widetilde{\mathbf{P}}_{\lambda} \end{pmatrix} \left(\mathbf{X}^{T}_{\text{VAM}} \mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1} \begin{pmatrix} \mathbf{X}^{T}_{\text{VAM}} \mathbf{U}^{T} & \widetilde{\mathbf{P}}^{T}_{\lambda} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{U}\,\mathbf{X}_{\text{VAM}} \left(\mathbf{X}^{T}_{\text{VAM}} \mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1} \mathbf{X}^{T}_{\text{VAM}} \mathbf{U}^{T} & \mathbf{U}\,\mathbf{X}_{\text{VAM}} \left(\mathbf{X}^{T}_{\text{VAM}} \mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1} \widetilde{\mathbf{P}}^{T}_{\lambda} \\ \widetilde{\mathbf{P}}_{\lambda} \left(\mathbf{X}^{T}_{\text{VAM}} \mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1} \mathbf{X}^{T}_{\text{VAM}} \mathbf{U}^{T} & \widetilde{\mathbf{P}}_{\lambda} \left(\mathbf{X}^{T}_{\text{VAM}} \mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1} \widetilde{\mathbf{P}}^{T}_{\lambda} \end{pmatrix}.
\end{aligned}
$$

The influence matrix $\mathbf{A}_{\lambda}$ for the P-spline VGAM model can be then taken as

$$
\mathbf{A}_{\lambda} \;=\; \mathbf{U}\,\mathbf{X}_{\text{VAM}} \left(\mathbf{X}^{T}_{\text{VAM}} \mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1} \mathbf{X}^{T}_{\text{VAM}} \mathbf{U}^{T}, \tag{4.54}
$$

computed upon convergence. However,

$$
\begin{aligned}
\operatorname{tr}(\mathbf{A}_{\lambda}) \;&=\; \operatorname{tr}\left\{\mathbf{X}_{\text{VAM}} \left(\mathbf{X}^{T}_{\text{VAM}} \mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1} \mathbf{X}^{T}_{\text{VAM}} \mathbf{W}\right\} && (4.55)\\
&=\; \operatorname{tr}\left\{\mathbf{X}^{T}_{\text{VAM}} \mathbf{W}\,\mathbf{X}_{\text{VAM}} \left(\mathbf{X}^{T}_{\text{VAM}} \mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1}\right\}, && (4.56)
\end{aligned}
$$

and we use the latter equation, following Marx and Eilers (1998) as it leads more efficient computation. The matrix operations performed in (4.56) are always cheaper to evaluate than those of (4.55). We note that for the constrained P-spline VGAMs, $\mathbf{X}_{\text{VAM}}$ and $\mathbf{P}_{\lambda}$ for the influence matrix $\mathbf{A}_{\lambda}$ in (4.54) are given by (4.43) and (4.46) respectively.

Since $\operatorname{tr}(\mathbf{A}_{\lambda})$ in (4.55) gives the EDF for the whole model, it is natural to divide this EDF into the effective degrees of freedom for each smooth term at each smooth predictor. For GAMs constructed using penalized regression splines, Wood (Section 4.4, 2006b) decomposed the elements on the leading diagonal of $\mathbf{A}$ (cf. equation (2.14)) into components relating to the different terms within the model to obtain the EDF for each smooth term. Following Wood (Section 4.4, 2006b), the EDF for each smooth term for our cases can be obtained as follows. We first define $\mathbf{F} = \left(\mathbf{X}^{T}_{\text{VAM}} \mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_{\lambda}\right)^{-1} \mathbf{X}^{T}_{\text{VAM}} \mathbf{W}$, which yields $\widehat{\boldsymbol{\beta}} = \mathbf{F}\,\boldsymbol{z}$ (in the unconstrained P-spline VGAM case). Hence $\mathbf{A}_{\lambda} = \mathbf{X}_{\text{VAM}}\,\mathbf{F}$ (cf. equation (4.55)) and each row of $\mathbf{F}$ is associated with one parameter. Then, let $\mathbf{F}_{(j)k}$ be the matrix $\mathbf{F}$ with all rows zeroed except for those associated with the parameters of the jth smooth predictor and the kth smooth term. Therefore,

$$\mathbf{A}_\lambda \;=\; \sum_{k=1}^{p}\sum_{j=1}^{M} \mathbf{X}_{\text{VAM}}\,\mathbf{F}_{(j)k}. \tag{4.57}$$

Equation (4.57) gives a straightforward way of calculating the EDF of the kth smooth term at the jth smooth predictor, i.e., the EDF for each smooth term for our cases can be taken as $\text{tr}(\mathbf{X}_{\text{VAM}}\,\mathbf{F}_{(j)k})$. For the constrained P-spline VGAMs, equation (4.57) is replaced by $\mathbf{A}_\lambda = \sum_{k=1}^{p}\sum_{j=1}^{\text{R}_k} \mathbf{X}_{\text{VAM}}\,\mathbf{F}_{(j)k}$.

4.5 Confidence Intervals

The generalization of GAMs using penalized regression splines allows the models to be estimated by penalized regression methods. This way, GAMs is reduced to GLMs. Furthermore, inference is straightforward. Following Marx and Eilers (1998) and Wood (Chapter 4, 2006b), confidence interval estimation for P-spline VGAMs can be described as follows. Recall that our parameter estimators are of the form (cf. equation (4.28))

$$\widehat{\boldsymbol{\beta}} \;=\; \left(\mathbf{X}_{\text{VAM}}^T\,\mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_\lambda\right)^{-1}\mathbf{X}_{\text{VAM}}^T\,\mathbf{W}\,\boldsymbol{y}, \tag{4.58}$$

where the data or the adjusted dependent variable $\boldsymbol{y}$ have variance-covariance matrix $\mathbf{W}^{-1}$. At the convergence, the variance-covariance matrix for the estimators $\widehat{\boldsymbol{\beta}}$ is

$$\text{Var}\left(\widehat{\boldsymbol{\beta}}\right) \;=\; \left(\mathbf{X}_{\text{VAM}}^T\,\mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_\lambda\right)^{-1}\mathbf{X}_{\text{VAM}}^T\,\mathbf{W}\,\mathbf{X}_{\text{VAM}}\,\left(\mathbf{X}_{\text{VAM}}^T\,\mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_\lambda\right)^{-1} \tag{4.59}$$

and so

$$\text{Var}\left(\widehat{\mathbf{f}}\right) \;\approx\; \mathbf{X}_{\text{VAM}}\,\left(\mathbf{X}_{\text{VAM}}^T\,\mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_\lambda\right)^{-1}\mathbf{X}_{\text{VAM}}^T\,\mathbf{W}\,\mathbf{X}_{\text{VAM}}\,\left(\mathbf{X}_{\text{VAM}}^T\,\mathbf{W}\,\mathbf{X}_{\text{VAM}} + \mathbf{P}_\lambda\right)^{-1}\mathbf{X}_{\text{VAM}}^T. \tag{4.60}$$

The diagonal elements of (4.60) are used to construct twice standard error bands for P-spline VGAM smooth term. As stated by Wood (Chapter 4, 2006b) and Wood (2006a), generally $\mathbf{E}\left(\widehat{\boldsymbol{\beta}}\right) \neq \boldsymbol{\beta}$ because the penalty induced bias, so that confidence intervals based on the use of

$\widehat{\boldsymbol{\beta}}$ and the corresponding variance-covariance matrix in (4.59) generally provide poor results in terms of realized coverage probabilities. Wood (2006a) overcame the poor performance of these intervals using a Bayesian approach proposed by Wahba (1983). Full details of approximate Bayesian interval estimates for GAMs are given by Wood (2006a).

4.6 Model Comparison

In model building process, we might be sometimes interested in testing a null hypothesis that the simpler of the two nested models is correct, against the alternative that the more complex of the two models is correct. Wood (Sections 2.1.6 and 4.10.1, 2006b) stated that the generalized log-likelihood ratio test is often used for such comparison, that is, the difference in deviance between the two models is applied as a test statistic. Consider testing,

$$\mathbf{H}_0 : \mathbf{X}_0\boldsymbol{\beta}_0 \quad \text{against} \quad \mathbf{H}_1 : \mathbf{X}_1\boldsymbol{\beta}_1,$$

where $\mathbf{X}_j$ is a model matrix including any strictly parametric model components and the terms representing spline bases for the smooth components with corresponding parameter vector $\boldsymbol{\beta}_j$, and the column space of $\mathbf{X}_0$ is contained in the column space of $\mathbf{X}_1$. If $\mathbf{H}_0$ is true, then a rough approximation in the large sample limit is given by

$$\mathrm{Dev}_0 - \mathrm{Dev}_1 \sim \chi^2_{\mathrm{EDF}_1 - \mathrm{EDF}_0}, \tag{4.61}$$

where Dev_0 and Dev_1 are the deviance under $\mathbf{H}_0$ and $\mathbf{H}_1$ respectively. The difference $\mathrm{Dev}_0 - \mathrm{Dev}_1$ is treated as following a χ^2 distribution with degrees of freedom given by the difference of EDF between the two models. If the scale parameter is unknown and has to be estimated, the F-ratio test is used for this comparison

$$F = \frac{(\mathrm{Dev}_0 - \mathrm{Dev}_1)\,/\,(\mathrm{EDF}_1 - \mathrm{EDF}_0)}{\mathrm{Dev}_1/\,(n - \mathrm{EDF}_1)} \sim F_{\mathrm{EDF}_1 - \mathrm{EDF}_0, n - \mathrm{EDF}_1}. \tag{4.62}$$

In (4.62), the F-ratio test is treated as following an F distribution with degrees of freedom given by $\mathrm{EDF}_1 - \mathrm{EDF}_0$ and $n - \mathrm{EDF}_1$.

Since P-spline VGAMs are developed based on the penalized regression splines, following Wood (Section 4.10.1 2006b), for any comparison of nested models in the P-spline VGAMs approach can be formed in the same way as Wood (2006b) did with GAMs using (4.61), or (4.62) if the scale parameter has to be estimated, and with the residual degrees of freedom approximated by (cf. Section 4.4.1, Wood, 2006b)

$$df^{\text{err}} \approx nM - \text{tr}(\mathbf{A}_{\lambda}).$$

4.7 Conclusions

In this chapter, we have proposed a fitting procedure based on generalizing the penalized regression spline approach of Marx and Eilers (1998) and Wood (2006b) to the VGAM class. We have shown a very natural extension from the P-spline implementation of GAMs to P-spline implementations of VGAMs. We have shown how penalized likelihood maximization for P-spline VGAMs can be implemented using penalized iteratively reweighted least squares (P-IRLS). Importantly, we have developed a model structure which allows the data to reveal the nature of an effects curve in a fairly unconstrained way and by constraining it to have suitable forms by taking the "constraints on the functions" approach of Yee and Wild (1996), and incorporating this idea into the P-spline VGAM framework to cover the full range of VGAM models. We concluded by defining "effective degrees of freedom" (EDF) and confidence intervals for P-spline VGAMs and providing further tools that are useful for applied modeling with our purposed method such as comparing models. We have also developed the necessary computational procedures. These have also been coded as R functions (see appendix A).

Chapter 5

Smoothing parameter selection and simulation study

In chapter 4, we generalized the ideas of penalized regression splines, based on P-spline smoothers from classical GAM modeling to the VGAM class, and developed an estimation procedure based on the penalized likelihood approach, which we called "P-spline VGAMs". This approach maximized the penalized log-likelihood which was of the form (cf. section 4.2 and 4.3)

$$\ell^*(\boldsymbol{\beta}^*) \;=\; \ell(\boldsymbol{\beta}^*) - \frac{1}{2}\,\boldsymbol{\beta}^{*T}\,\mathbf{P}^*_\lambda\,\boldsymbol{\beta}^*. \tag{5.1}$$

In Chapter 4, the smoothing parameters involved in (5.1) were taken as known rather than estimated. As noted by Wood (2006b), the smoothing parameters cannot be estimated by maximizing (5.1) jointly over $\boldsymbol{\beta}^*$ and $\boldsymbol{\lambda}$ because this maximization always yields an estimated smoothing-parameter of zero. The most complex model will always be chosen since the highest value for (5.1) would be obtained when smoothing parameters equal zero. Therefore, the smoothing parameters have to be estimated using an alternative criterion. If the model has only one smooth term, then estimation of smoothing parameters by applying direct grid-search optimization to a criterion such as generalized cross validation (GCV) or the Akaike information criterion (AIC) works well. But if there is more than one smooth term in the model, then this estimation can be computationally expensive. In Section 2.2, we discussed a number of possible techniques for automatic selection of the multiple smoothing parameters within the penalized likelihood framework, mainly drawn from work by Wood who proposed suitable parametric representations for the smooth functions, and a computational method to control and choose the degree of smoothness appropriately. He developed an efficient computational method for automatic multiple smoothing-parameter selection for GAMs based on penalized regression splines. In a series of papers, Marra and Radice (2011), Marra et al. (2013b), Marra (2013), Marra et al. (2013a), and Radice et al. (2015) extended GAMs based on penalized regression splines to a simultaneous system of two binary equations. Their model constructions allowed the degree of smoothness of the model component functions to be estimated using Wood (2004)'s method. They showed how the minimization of the UBRE score version developed for a simultaneous system of two binary outcomes can be achieved using the approach of Wood (2004).

In this chapter, we will discuss the theory and computational details for automatic smoothing-parameter estimation developed by Wood (2004) and an extension of the penalized likelihood approach presented in Marra and Radice (2011) for the simultaneous system of two binary equations in which the amount of smoothing for the smooth components is allowed to be chosen automatically through minimization of the UBRE score. We will then extend the smoothness-selection procedures discussed in Wood (2004) and Marra and Radice (2011) to the P-spline VGAM formulated using penalized regression splines described in Chapter 4, and develop the general form of smoothing-parameter selection criteria such as the GCV and UBRE scores for the full range of VGAM models including those complications such as constraints on model terms. We will then discuss simulation studies conducted to investigate the empirical performance of the method proposed.

5.1 Stable and efficient multiple smoothing parameter estimation for GAMs

As discussed in Chapter 2, GAMs represented using penalized regression splines can be employed in a straightforward manner using penalized regression methods and this makes it possible to integrate the smoothing parameter selection into model fitting in a manner that is both computationally efficient and stable using well-founded criteria such as GCV or UBRE. Smoothing-parameter estimation under such criteria can be included as part of the P-IRLS scheme used to fit GAMs by penalized likelihood maximization. Wood (2004) proposed the method for doing this. In this section, we will discuss some theoretical aspects that underpin the method for smoothing-parameter selection developed by Wood (2004) and also discuss in some detail how this method can be integrated into the model fitting in an efficient manner.

Recall from equation (2.13) that the basic GAM fitting problem can be written in the form

$$\text{minimize } \left\| \sqrt{\mathbf{W}} \left(\boldsymbol{z} - \mathbf{X}\boldsymbol{\beta} \right) \right\|^2 + \boldsymbol{\beta}^T \mathbf{S} \boldsymbol{\beta}. \tag{5.2}$$

For given the smoothing parameters, (5.2) can be easily solved, but the smoothing parameters have to be estimated. As discussed in Section 2.2.2, for the problem (5.2) the smoothing parameters can be estimated by minimizing the GCV or the UBRE scores:

$$\mathcal{V}_g^w = \frac{n \| \sqrt{\mathbf{W}} \left(\boldsymbol{z} - \mathbf{X}\boldsymbol{\beta} \right) \|^2}{\left[n - \operatorname{tr}\left(\mathbf{A} \right) \right]^2}, \quad \text{or} \tag{5.3}$$

$$\mathcal{V}_u^w = \frac{1}{n} \| \sqrt{\mathbf{W}} \left(\boldsymbol{z} - \mathbf{X}\boldsymbol{\beta} \right) \|^2 - \sigma^2 + \frac{2}{n} \operatorname{tr}(\mathbf{A})\, \sigma^2, \tag{5.4}$$

with respect to the smoothing parameter vector $\boldsymbol{\lambda}$. Here, $\mathbf{A}$ is the influence matrix for the model and depends on the smoothing parameters. Wood (2004) used an approach based on an optimization strategy of Gu (1992) known as “performance iteration”, for estimating smoothing parameters using (5.3) or (5.4) minimization. This approach uses the fact that at each P-IRLS iteration a penalized weighted least squares problem is solved, and the smoothing parameters of that problem can be estimated by minimizing the GCV or UBRE scores (cf. Section 2.2.2).

Therefore, for each working penalized linear model of the P-IRLS iteration, Wood (2004) minimized the score $\mathcal{V}$, in (5.3) or (5.4) with respect to the smoothing parameters using a numerical approach based on Newton's method parameterized in terms of the log smoothing parameters:

$$\mathcal{V}(\boldsymbol{\lambda}) \;\simeq\; \mathcal{V}(\boldsymbol{\lambda}^{(t)}) + (\boldsymbol{\lambda}-\boldsymbol{\lambda}^{(t)})^T\boldsymbol{m} + \frac{1}{2}(\boldsymbol{\lambda}-\boldsymbol{\lambda}^{(t)})^T\mathbf{M}\,(\boldsymbol{\lambda}-\boldsymbol{\lambda}^{(t)}),$$

where $\boldsymbol{m}$ and $\mathbf{M}$ are the first derivative vector and second derivative matrix of $\mathcal{V}(\boldsymbol{\lambda})$ with respect to the smoothing parameters. He worked with log-transformed smoothing parameters to ensure that the smoothing parameter estimates stayed positive. The t^{th} estimate of $\boldsymbol{\lambda}$ was updated by minimizing the approximating quadratic to give

$$\boldsymbol{\lambda}^{(t+1)} \;=\; \boldsymbol{\lambda}^{(t)} - \mathbf{M}^{-1}\boldsymbol{m}. \tag{5.5}$$

The process is repeated, till convergence is met. Importantly, he implemented (5.5) in a way that the GCV and UBRE scores and their derivatives can be evaluated in a manner that is both computationally efficient and stable. To do this, he used a method based on pivoted QR decomposition and a singular value decomposition. He first transformed the influence matrix: $\mathbf{A} \;=\; \mathbf{X}\left(\mathbf{X}^T\mathbf{W}\,\mathbf{X}+\mathbf{S}\right)^{-1}\mathbf{X}^T\mathbf{W}$ (cf. equation (2.14)) involved in the expression for $\mathcal{V}$ and therefore is $\boldsymbol{m}$ and $\mathbf{M}$, into a more useful form using the QR decomposition of $\sqrt{\mathbf{W}}\,\mathbf{X}$,

$$\sqrt{\mathbf{W}}\,\mathbf{X} \;=\; \mathbf{Q}\,\mathbf{R}, \tag{5.6}$$

where $\mathbf{Q}$ is made up of columns of an orthogonal matrix and $\mathbf{R}$ is an upper triangular matrix. He defined $\mathbf{S} \;=\; \sum_{k=1}^{p}\lambda_k\,\mathbf{S}_k$ and $\mathbf{B}$ to be any matrix square root of $\mathbf{S}$ such that $\mathbf{B}^T\mathbf{B} \;=\; \mathbf{S}$, (e.g., obtained by Cholesky decomposition or by eigen decomposition) and formed a singular value decomposition of an augmented matrix of $\mathbf{R}$ with $\mathbf{B}$:

$$\begin{pmatrix}\mathbf{R}\\ \mathbf{B}\end{pmatrix} \;=\; \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T. \tag{5.7}$$

Here, the matrix $\mathbf{U}$ has orthogonal columns, $\mathbf{V}$ is an orthogonal matrix and $\mathbf{D}$ is the diagonal matrix of singular values. Wood (2004) stated that numerical rank deficiency of the fitting problem can be reliably identified by examining the singular values in matrix $\mathbf{D}$. He then defined $\mathbf{U}_1$ is the sub-matrix of $\mathbf{U}$, so gives the explicit form, e.g., $\mathbf{U} = \left(\mathbf{U}_1^T, \mathbf{U}_2^T\right)^T$, so that $\mathbf{R} = \mathbf{U}_1 \mathbf{D} \mathbf{V}^T$. This gives

$$\sqrt{\mathbf{W}}\,\mathbf{X} = \mathbf{Q}\,\mathbf{U}_1\,\mathbf{D}\,\mathbf{V}^T, \qquad \text{while} \quad \mathbf{X}^T\mathbf{W}\,\mathbf{X} + \mathbf{S} = \mathbf{V}\,\mathbf{D}^2\,\mathbf{V}^T. \tag{5.8}$$

As a consequence, $\mathbf{A}$ and $\operatorname{tr}(\mathbf{A})$ can be written as

$$\mathbf{A} = \mathbf{Q}\mathbf{U}_1\mathbf{U}_1^T\mathbf{Q}^T, \ \text{ and } \ \operatorname{tr}(\mathbf{A}) = \operatorname{tr}(\mathbf{U}_1\mathbf{U}_1^T). \tag{5.9}$$

Since the main computational cost is by forming the QR decomposition, the calculations of the quantities related to $\mathbf{A}$, for example $\operatorname{tr}(\mathbf{A})$ (5.9), for new trial values of $\boldsymbol{\lambda}$ become relatively cheap. Using the orthogonal matrix factorizations above, Wood (2004) obtained convenient expressions for the component derivatives of the GCV and UBRE scores. For each updated $\boldsymbol{\lambda}$, these derivatives can be evaluated in a stable manner, giving an efficient implementation of Newton's method for finding the optimum $\boldsymbol{\lambda}$. Further details of the stable multiple smoothing-parameter estimation by GCV or UBRE can be found in Wood (2004). An implementation of Wood (2004)'s method is the function `magic()` in the `mgcv` package.

Function `magic` implemented the computational method of applying GCV or UBRE to problem of smoothing parameter selection in the context of a penalized least squares problem (5.2). The smoothing parameters are then chosen to minimize either the GCV score (5.3) or the UBRE score (5.4) using Newton's method parameterized in terms of the log smoothing parameters described above. The major inputs for `magic()` are the model matrix ($\mathbf{X}$ for the unweighted case and $\sqrt{\mathbf{W}}\,\mathbf{X}$ for the weighted case), the response vector or the adjusted dependent vector ($\boldsymbol{y}$ for the unweighted case and $\sqrt{\mathbf{W}}\,\boldsymbol{z}$ for the weighted case), the array of smoothing parameters ($\{\lambda_1, \ldots, \lambda_p\}$) and a list of penalty matrices ($\{\mathbf{S}_1, \ldots, \mathbf{S}_p\}$). The function returns a list of several items and the major outputs are as follows: the estimated parameters given the estimated smoothing parameters, the estimated (GCV) or supplied (UBRE) scale parameter, the minimized GCV or UBRE score and the estimated smoothing parameters. In Section 5.3, we will extend the smoothness selection procedure described in this section to our framework and the `magic()` function will be employed to minimize the GCV score or the UBRE score with respect to the smoothing parameters for our problem.

5.2 Penalized likelihood approach to binary response modeling and smoothness component selection

In this section, we will discuss the work done by Marra and Radice (2011) that extended the penalized likelihood approach to the simultaneous system of two binary equations in which their model structure allows the degree of smoothness for the smooth components to be estimated automatically by minimizing the UBRE score.

Marra and Radice (2011) introduced a fitting procedure based on the penalized regression spline approach for nonstandard semiparametric bivariate probit model (whose model exhibits the recursive structure) of the form

$$\begin{aligned} y_{1i}^* &= \boldsymbol{x}_{1i}^{+}\boldsymbol{\alpha}_1 + \sum_{k_1=1}^{K_1} f_{k_1}(z_{1k_1 i}) + \varepsilon_{1i}, \\ & \qquad\qquad\qquad\qquad\qquad i = 1,\ldots,n, \\ y_{2i}^* &= \beta\, y_{1i} + \boldsymbol{x}_{2i}^{+}\boldsymbol{\alpha}_2 + \sum_{k_2=1}^{K_2} f_{k_2}(z_{2k_2 i}) + \varepsilon_{2i}, \end{aligned} \tag{5.10}$$

where the observed binary outcomes y_{1i} and y_{2i} are specified according to the rule:

$$y_{vi} = \begin{cases} 1 & \text{if } y_{vi}^* > 0 \\ 0 & \text{if } y_{vi}^* \leq 0 \end{cases} \; ; v = 1, 2,$$

and $\boldsymbol{x}_{1i}^{+}$ and $\boldsymbol{x}_{2i}^{+}$ are respectively the ith row vector of the model matrices $\boldsymbol{x}_{1}^{+}$ and $\boldsymbol{x}_{2}^{+}$ for any parametric model components, with corresponding parameter vectors $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$; and f_{k_1} and f_{k_2} are unknown smooth terms for $z_{1k_1 i}$ and $z_{2k_2 i}$. The error terms $(\varepsilon_{1i}, \varepsilon_{2i})$ are identically distributed as bivariate normal with zero mean, unit variance and correlation coefficient θ_3, independently across observations:

$$\begin{pmatrix} \varepsilon_{1i} \\ \varepsilon_{2i} \end{pmatrix} \overset{iid}{\sim} N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \theta_3 \\ \theta_3 & 1 \end{bmatrix} \right).$$

Marra and Radice (2011) represented the smooth model components using the regression spline approach, where basis functions such as P-splines, cubic regression and thin plate regression splines were used. They defined the linear predictor for (5.10) as

$$\begin{aligned} \eta_{1i} &= \boldsymbol{x}_{1i}\boldsymbol{\theta}_1, \\ \eta_{2i} &= \boldsymbol{x}_{2i}\boldsymbol{\theta}_2, \end{aligned}$$

where $\boldsymbol{x}_{1i}$ consists of $\boldsymbol{x}_{1i}^{+}$ and the quantities corresponding to the spline bases for the smooth functions, and $\boldsymbol{\theta}_1$ contains parameter $\boldsymbol{\alpha}_1$ and the regression coefficients associated with the smooths. They defined $\boldsymbol{x}_{2i}$ and $\boldsymbol{\theta}_2$ in the same way as for $\boldsymbol{x}_{1i}$ and $\boldsymbol{\theta}_1$, except that, $\boldsymbol{x}_{2i}$ and $\boldsymbol{\theta}_2$ contain y_{i1} and β, respectively.

Replacing the smooth components in (5.10) with their regression spline expressions results in a fully parametric recursive bivariate probit model. Marra and Radice (2011) therefore expressed two binary responses y_{1i} and y_{2i} as the bivariate probit model of the form

$$\Pr(y_{1i} = 1, y_{2i} = 1 | \boldsymbol{x}_{1i}, \boldsymbol{x}_{2i}) = \Phi_2(\eta_{1i}, \eta_{2i}; \theta_3), \tag{5.11}$$

where Φ_2 is the cumulative distribution function of a standard bivariate normal distribution with zero mean, unit variance and correlation coefficient θ_3. They then employed penalized likelihood maximization to estimate (5.11). The penalized likelihood for their models is given by

$$\ell_p(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}) - \frac{1}{2}\boldsymbol{\theta}^T \mathbf{S}_\lambda \boldsymbol{\theta},$$

where $\mathbf{S}_\lambda$ is the penalty matrix of the model and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \theta_3)$ is the parameter vector. Given values for smoothing parameters, they maximized $\ell_p(\boldsymbol{\theta})$ with respect to $\widehat{\boldsymbol{\theta}}$ using Fisher scoring. The current estimate of $\boldsymbol{\theta}$ at the tth iteration is updated using

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \hat{\boldsymbol{\theta}}^{(t)} + \left(\boldsymbol{\mathcal{I}}^{(t)} + \mathbf{S}_\lambda\right)^{-1} \left(\boldsymbol{g}^{(t)} - \mathbf{S}_\lambda \hat{\boldsymbol{\theta}}^{(t)}\right), \tag{5.12}$$

where $\boldsymbol{\mathcal{I}}$ is the Fisher information matrix, and $\boldsymbol{g}$ is defined by the two subvectors $\boldsymbol{g}_1 = \partial\ell/\partial\boldsymbol{\theta}_1$ and $\boldsymbol{g}_2 = \partial\ell/\partial\boldsymbol{\theta}_2$, and a scalar $g_3^* = \partial\ell/\partial\theta_3^*$, where

$$\theta_3^* = \tanh^{-1}(\theta_3) = (1/2)\log\{(1+\theta_3)/(1-\theta_3)\}$$

is the common transform for correlation θ_3 (cf. Marra and Radice (2011)). Parameter estimation for model (5.11) can also be achieved using the VGAM approach of Yee and Wild (1996).

Given values for the smoothing parameters, Marra and Radice (2011) considered Fisher scoring step (5.12) in the P-IRLS form, and estimated $\hat{\boldsymbol{\theta}}^{(t+1)}$ by solving the problem

$$\text{minimize} \qquad \left\| \sqrt{\mathbf{W}}^{(t)} \left(\boldsymbol{z}^{(t)} - \mathbf{X}\,\boldsymbol{\theta} \right) \right\|^2 + \boldsymbol{\theta}^T \mathbf{S}_\lambda\,\boldsymbol{\theta}, \tag{5.13}$$

with respect to $\boldsymbol{\theta}$. Here, $\sqrt{\mathbf{W}}^{(t)}$ is any matrix square root of $\mathbf{W}^{(t)}$, i.e., satisfying

$$\sqrt{\mathbf{W}}^{(t)T}\sqrt{\mathbf{W}}^{(t)} = \mathbf{W}^{(t)},$$

while $\boldsymbol{z}^{(t)}$ is a pseudo data-vector, and $\mathbf{X}_i = \operatorname{diag}(\boldsymbol{x}_{1i}, \boldsymbol{x}_{2i}, 1)$. They estimated smoothing parameters for problem (5.13) by minimizing a prediction error criterion such as the UBRE score

$$\mathcal{V}_u^w = \frac{1}{n}\|\sqrt{\mathbf{W}}\,(\boldsymbol{z} - \mathbf{X}\,\boldsymbol{\theta})\,\|^2 - 1 + \frac{2}{n}\,\gamma\,\operatorname{tr}(\mathbf{A}), \tag{5.14}$$

where $\mathbf{A} = \mathbf{X}(\mathbf{X}^T\mathbf{W}\mathbf{X} + \mathbf{S}_\lambda)^{-1}\mathbf{X}^T\mathbf{W}$ is the influence matrix, and γ is a tuning parameter used to obtain smoother models. They applied the "performance iteration" approach of Gu (1992) for smoothing-parameter estimation. They then minimized the UBRE score with respect to the smoothing parameters using the computational procedure of Wood (2004), and selected smoothing parameters for each working penalized linear model of the P-IRLS iteration. The two steps of estimating the model parameters $\boldsymbol{\theta}$ fixing $\boldsymbol{\lambda}$ and estimating the smoothing parameters $\boldsymbol{\lambda}$ fixing $\boldsymbol{\theta}$ were iterated until convergence is met. Their simulation results show that their proposed method performs well, and yields precise estimates in most data settings and under correct model specification.

Marra et al. (2013b) and Radice et al. (2015) extended the procedures discussed in Marra and Radice (2011) to deal simultaneously with two binary regression models involving semiparametric predictors in the presence of non-random sample selection, and the binary outcome in the presence of unobserved confounding, respectively. Both approaches are based on the penalized likelihood approach, and the smoothing parameters are estimated by minimizing the UBRE score in the same manner as proposed in Marra and Radice (2011). The methods discussed in Marra and Radice (2011), Marra et al. (2013b), and Radice et al. (2015) are implemented in the SemiParBIVProbit package (Marra and Radice, 2013).

5.3 Smoothing parameter estimation for P-spline VGAMs

Section 5.1 showed how smoothing-parameter estimation for GAMs based on the penalized likelihood approach can be achieved by minimizing the GCV or UBRE scores. This estimation procedure can be included as part of the P-IRLS scheme used to fit GAMs by penalized likelihood maximization in an efficient and stable manner. Section 5.2 showed that smoothing-parameter estimation for the penalized likelihood approach to the binary response modeling can be achieved by minimizing the UBRE score in the same manner as it was done for GAMs. This leads to a solution to the unresolved question from the previous chapter: how to choose $\boldsymbol{\lambda}$ for P-spline VGAMs? Since P-spline VGAMs are based on the penalized likelihood approach, following Sections 5.1 and 5.2, the smoothing parameters can be estimated by minimizing the GCV or UBRE scores in the same manner as proposed in Wood (2004) and Marra and Radice (2011). The only change is that such criteria are formulated using a more complicated model matrix, penalty matrix and influence matrix (cf. equations (5.3), (5.4) and (5.14)). In this section, we will formulate the general form of smoothing-parameter selection criteria such as the UBRE score for our problem. We will then show how the minimization of our UBRE score can be achieved using the computational approach of Wood (2004) and show how this minimization can be included in our P-IRLS scheme using the “performance iteration” approach.

Recall from equation (4.28) that the Fisher scoring method for the constrained P-spline

VGAMs gave an iterative estimation for $\boldsymbol{\beta}^*$ in the form of

$$\boldsymbol{\beta}^{*(t+1)} = \left(\mathbf{X}_{\text{VAM}}^T \mathbf{W}^{(t)} \mathbf{X}_{\text{VAM}} + \mathbf{P}_\lambda^*\right)^{-1} \left(\mathbf{X}_{\text{VAM}}^T \mathbf{W}^{(t)} \boldsymbol{z}^{(t)}\right), \tag{5.15}$$

which we implemented by solving the equivalent weighted penalized least squares problem

$$\text{minimize} \quad \left\| \mathbf{U}^{(t)} \left(\boldsymbol{z}^{(t)} - \mathbf{X}_{\text{VAM}} \boldsymbol{\beta}^*\right) \right\|^2 + \boldsymbol{\beta}^{*\boldsymbol{T}} \mathbf{P}_\lambda^* \boldsymbol{\beta}^*, \tag{5.16}$$

with respect to $\boldsymbol{\beta}^*$(cf. the weighted penalized least squares problem of Wood (2004) in (5.2) and Marra and Radice (2011) in (5.13)). Here, as we saw in (4.34) $\mathbf{U}^{(t)}$ is the Cholesky decomposition of $\mathbf{W}^{(t)}$ such that

$$\mathbf{U}^{T(t)}\mathbf{U}^{(t)} = \mathbf{W}^{(t)},$$

$\mathbf{W}^{(t)}$ is working weights, $\boldsymbol{z}^{(t)}$ is the adjusted dependent vector given by

$$\boldsymbol{z}^{(t)} = \mathbf{X}_{\text{VAM}} \boldsymbol{\beta}^{*(t)} + \left(\mathbf{W}^{(t)}\right)^{-1} \boldsymbol{u}^{(t)},$$

$\mathbf{X}_{\text{VAM}}$ is the “vector additive model” model matrix and $\mathbf{P}_\lambda^*$ is the penalty matrix (cf. equations (4.43) and (4.46)). Again, for given smoothing parameters (5.16) can be easily solved, but the smoothing parameters have to be estimated. Following Sections 5.1 and 5.2, the smoothing parameters for our problem (5.16) can be estimated by minimizing the GCV or UBRE scores. To do this, we first formulate the general forms of smoothing parameter selection criteria such as the UBRE score for our problem.

To obtain the general form of the UBRE score for our case, we re-write the penalized iterative solution of (5.16) in the form of

$$\left\| \widetilde{\boldsymbol{z}}^{(t)} - \widetilde{\mathbf{X}}_{\text{VAM}}^{(t)} \boldsymbol{\beta}^* \right\|^2 + \boldsymbol{\beta}^{*\boldsymbol{T}} \mathbf{P}_\lambda^* \boldsymbol{\beta}^*. \tag{5.17}$$

We note that (5.17) is equivalent to the unweighted form for the penalized least squares objective used for the smoothing parameter selection method by Wood (2004). Here, $\widetilde{\boldsymbol{z}}^{(t)} = \mathbf{U}^{(t)} \boldsymbol{z}^{(t)} = \mathbf{U}^{(t)}\mathbf{X}_{\text{VAM}} \boldsymbol{\beta}^{*(t)} + \mathbf{U}^{(t)} \left(\mathbf{W}^{(t)}\right)^{-1} \boldsymbol{u}^{(t)}$, and $\widetilde{\mathbf{X}}_{\text{VAM}}^{(t)} = \mathbf{U}^{(t)} \mathbf{X}_{\text{VAM}}$. Following Radice et al. (2015), we treat $\widetilde{\boldsymbol{z}}$ as if it had a standard linear model. Then $\widetilde{\boldsymbol{z}} = E(\widetilde{\boldsymbol{z}}) + \boldsymbol{\varepsilon}$, where $E(\widetilde{\boldsymbol{z}}) = \boldsymbol{\mu}_{\widetilde{\boldsymbol{z}}} =$

$\mathbf{U}\,\mathbf{X}_{\text{VAM}}\,\boldsymbol{\beta}_0^*$, $\boldsymbol{\beta}_0^*$ is a true parameter vector and $\boldsymbol{\varepsilon} = \mathbf{U}\,\mathbf{W}^{-1}\,\boldsymbol{u}$, having $\mathbf{0}$ mean and an identity covariance matrix $\mathbf{I}$. The predicted vector for $\widetilde{\boldsymbol{z}}$ is given by $\widehat{\boldsymbol{\mu}}_{\widetilde{\boldsymbol{z}}} = \mathbf{A}_\lambda\,\widetilde{\boldsymbol{z}}$. Recall from equation (4.54) that $\mathbf{A}_\lambda$ is the "influence matrix" for the model being fitted, and is given by

$$\mathbf{A}_\lambda \;=\; \widetilde{\mathbf{X}}_{\text{VAM}}\left(\widetilde{\mathbf{X}}_{\text{VAM}}^T\,\widetilde{\mathbf{X}}_{\text{VAM}} + \mathbf{P}_\lambda^*\right)^{-1}\widetilde{\mathbf{X}}_{\text{VAM}}^T \tag{5.18}$$

We now wish to select λ_k in a way that leads the estimated smooth functions to get closer to the true functions. In our cases, we would like to choose the smoothing parameters that tend to make $\widehat{\boldsymbol{\mu}}_{\widetilde{\boldsymbol{z}}}$ as close as possible to the true $\boldsymbol{\mu}_{\widetilde{\boldsymbol{z}}}$. An appropriate metric is the expected mean square error (MSE) of the model (cf. Chapter 4 (Wood, 2006b)). Following Wood (2006b), the expected MSE for our cases can be obtained in the following way

$$\begin{aligned}
E\left(\|\boldsymbol{\mu}_{\widetilde{\boldsymbol{z}}} - \widehat{\boldsymbol{\mu}}_{\widetilde{\boldsymbol{z}}}\|^2/\widetilde{n}\right) \;&=\; E\left(\|\boldsymbol{\mu}_{\widetilde{\boldsymbol{z}}} - \mathbf{A}_\lambda\,\widetilde{\boldsymbol{z}}\|^2\right)/\widetilde{n} \;=\; E\left(\|\widetilde{\boldsymbol{z}} - \boldsymbol{\varepsilon} - \mathbf{A}_\lambda\,\widetilde{\boldsymbol{z}}\|^2\right)/\widetilde{n} \;=\; E\left(\|\widetilde{\boldsymbol{z}} - \mathbf{A}_\lambda\,\widetilde{\boldsymbol{z}} - \boldsymbol{\varepsilon}\|^2\right)/\widetilde{n}\\
&=\; E\left(\|\widetilde{\boldsymbol{z}} - \mathbf{A}_\lambda\,\widetilde{\boldsymbol{z}}\|^2\right)/\widetilde{n} + E\left(\boldsymbol{\varepsilon}^T\,\boldsymbol{\varepsilon}\right)/\widetilde{n} - E\left(2\boldsymbol{\varepsilon}^T\left(\widetilde{\boldsymbol{z}} - \mathbf{A}_\lambda\,\widetilde{\boldsymbol{z}}\right)\right)/\widetilde{n}\\
&=\; E\left(\|\widetilde{\boldsymbol{z}} - \mathbf{A}_\lambda\,\widetilde{\boldsymbol{z}}\|^2\right)/\widetilde{n} + E\left(\boldsymbol{\varepsilon}^T\,\boldsymbol{\varepsilon}\right)/\widetilde{n} - E\left(2\boldsymbol{\varepsilon}^T\left(\boldsymbol{\mu}_{\widetilde{\boldsymbol{z}}} + \boldsymbol{\varepsilon}\right) - 2\boldsymbol{\varepsilon}^T\mathbf{A}_\lambda\left(\boldsymbol{\mu}_{\widetilde{\boldsymbol{z}}} + \boldsymbol{\varepsilon}\right)\right)/\widetilde{n}\\
&=\; E\left(\|\widetilde{\boldsymbol{z}} - \mathbf{A}_\lambda\,\widetilde{\boldsymbol{z}}\|^2\right)/\widetilde{n} + E\left(-\boldsymbol{\varepsilon}^T\,\boldsymbol{\varepsilon} - 2\,\boldsymbol{\varepsilon}^T\,\boldsymbol{\mu}_{\widetilde{\boldsymbol{z}}} + 2\,\boldsymbol{\varepsilon}^T\,\mathbf{A}_\lambda\,\boldsymbol{\mu}_{\widetilde{\boldsymbol{z}}} + 2\,\boldsymbol{\varepsilon}^T\,\mathbf{A}_\lambda\,\boldsymbol{\varepsilon}\right)/\widetilde{n}\\
&=\; E\left(\|\widetilde{\boldsymbol{z}} - \mathbf{A}_\lambda\,\widetilde{\boldsymbol{z}}\|^2\right)/\widetilde{n} - 1 + 2\,\text{tr}(\mathbf{A}_\lambda)/\widetilde{n},
\end{aligned} \tag{5.19}$$

where

$$\begin{aligned}
\widetilde{n} \;=\; nM,\; E\left(\boldsymbol{\varepsilon}^T\,\boldsymbol{\varepsilon}\right) \;=\; nM,\; E\left(\boldsymbol{\varepsilon}^T\,\boldsymbol{\mu}_{\widetilde{\boldsymbol{z}}}\right) \;=\; 0,\; E\left(\boldsymbol{\varepsilon}^T\,\mathbf{A}_\lambda\,\boldsymbol{\mu}_{\widetilde{\boldsymbol{z}}}\right) \;=\; 0,\\
E\left(\boldsymbol{\varepsilon}^T\,\mathbf{A}_\lambda\,\boldsymbol{\varepsilon}\right) \;=\; E\{\text{tr}\left(\boldsymbol{\varepsilon}^T\,\mathbf{A}_\lambda\,\boldsymbol{\varepsilon}\right)\} \;=\; E\{\text{tr}\left(\mathbf{A}_\lambda\,\boldsymbol{\varepsilon}\,\boldsymbol{\varepsilon}^T\right)\} \;=\; \text{tr}\left(\mathbf{A}_\lambda\right)
\end{aligned}$$

Then, an estimate of the expected MSE of the model (5.19) can be written as

$$\mathcal{V}_u\left(\boldsymbol{\lambda}\right) \;=\; \frac{1}{\widetilde{n}}\|\widetilde{\boldsymbol{z}} - \mathbf{A}_\lambda\,\widetilde{\boldsymbol{z}}\|^2 - 1 + \frac{2}{\widetilde{n}}\,\text{tr}(\mathbf{A}_\lambda). \tag{5.20}$$

The expression (5.20) is equivalent to the UBRE score defined in Wood (2004) (cf. equation (5.4)) and the approximate UBRE version of Marra and Radice (2011), Marra et al. (2013b), and Radice et al. (2015) (cf. equation (5.14)). It is the form used for automatic multiple smoothing-parameter estimation by Wood (2004). (5.20) is the approximate UBRE score for the P-spline

VGAM. Smoothing parameter selection for our case can be now achieved by applying the UBRE estimation of $\boldsymbol{\lambda}$ to each working linear model of the P-IRLS scheme used to fit the model. The method for doing this will be described in the next subsection.

5.3.1 Estimating smoothing parameters $\boldsymbol{\lambda}$ for given $\widehat{\boldsymbol{\beta}}^*$

Given an estimate for $\boldsymbol{\beta}^*$, we now can estimate $\boldsymbol{\lambda}$ by minimizing (5.20). That is, $\boldsymbol{\lambda}^{(t+1)}$ solves the problem

$$\min_{\boldsymbol{\lambda}} \qquad \mathcal{V}_u\left(\boldsymbol{\lambda}\right) = \frac{1}{\widetilde{n}}\|\widetilde{\boldsymbol{z}}^{(t)} - \mathbf{A}_{\lambda}^{(t)}\,\widetilde{\boldsymbol{z}}^{(t)}\|^2 - 1 + \frac{2}{\widetilde{n}}\,\mathrm{tr}(\mathbf{A}_{\lambda}^{(t)}). \tag{5.21}$$

Here, the working penalized linear model quantities related to (5.21) are calculated using the parameter estimates from the optimization step (5.16). Recall that the influence matrix is $\mathbf{A}_{\lambda}$ and the smoothing parameter vector $\boldsymbol{\lambda}$ enters (5.21) via $\mathbf{A}_{\lambda}$. For each working penalized linear model of the P-IRLS iteration, we minimize (5.20) with respect to the smoothing parameter vector. Following Wood (2004) and Marra and Radice (2011), the two steps of our estimations are as follows: at each iteration, (i) the penalized iteratively re-weighted least squares problem in (5.16) is solved and (ii) the smoothing parameters for the problem (5.16) are estimated by minimizing the approximate UBRE score (5.20). Steps (i) and (ii) are iterated until $\Delta\left(\boldsymbol{\beta}^{*(t+1)},\, \boldsymbol{\beta}^{*(t)}\right) = \left\|\boldsymbol{\beta}^{*(t+1)} - \boldsymbol{\beta}^{*(t)}\right\|$ is sufficiently small. The two steps can be stated more precisely as follows:

Step 1 The smoothing parameter vector is fixed at $\boldsymbol{\lambda}^{(t)}$, the estimate of $\boldsymbol{\beta}^*$ is updated to

$$\boldsymbol{\beta}^{*(t+1)} = \underset{\boldsymbol{\beta}^*}{\mathrm{argmax}}\ \ell^*(\boldsymbol{\beta}^*).$$

Step 2 $\boldsymbol{\beta}^*$ is fixed at $\boldsymbol{\beta}^{*(t+1)}$, the estimate of $\boldsymbol{\lambda}$ is updated to

$$\boldsymbol{\lambda}^{(t+1)} = \underset{\boldsymbol{\lambda}}{\mathrm{argmin}}\ \mathcal{V}_u\left(\boldsymbol{\lambda}\right).$$

The working penalized linear model quantities related to $\mathcal{V}_u(\boldsymbol{\lambda})$ in (5.20) are calculated using (5.16). We solve the problem (5.21) using the computational approach by Wood (2004) (cf.

Section 5.1). Following Wood (2004)'s method, our UBRE score (5.20) can be estimated using Newton's method with log-transformed smoothing parameters in which the UBRE score and their derivatives can be evaluated in a manner that is both computationally efficient and stable using a method based on pivoted QR decomposition and a singular value decomposition. To do this, our influence matrix $\mathbf{A}_\lambda$ in (5.18) involved in the expression for $\mathcal{V}_u\left(\boldsymbol{\lambda}\right)$ in (5.20) and therefore is $\boldsymbol{m}$ and $\mathbf{M}$ in (5.5), is first transformed into a more useful form. For our case, the relevant QR decomposition of the model matrix is $\widetilde{\mathbf{X}}_{\text{VAM}} = \mathbf{Q}\,\mathbf{R}$ (cf. equation (5.6)) and the relevant singular value decomposition of an augmented matrix of $\mathbf{R}$ with the matrix square root of the penalty is $\begin{pmatrix}\mathbf{R}\\ \widetilde{\mathbf{P}}^*\end{pmatrix} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$ (cf. equation (5.7)), where $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{U}$, $\mathbf{D}$ and $\mathbf{V}$ are defined in the same way as they are defined in Section 5.1 and recall that $\widetilde{\mathbf{P}}_\lambda^{*T}\,\widetilde{\mathbf{P}}_\lambda^* = \mathbf{P}_\lambda^*$ (cf. equation (4.30)). Based on this, the evaluation of the quantities relating to $\mathbf{A}_\lambda$ (5.18), for new trial values of $\boldsymbol{\lambda}$ is relatively cheap and therefore the derivatives of $\mathcal{V}_u\left(\boldsymbol{\lambda}\right)$ in (5.20) can be evaluated in a efficient and stable manner (cf. Section 5.1).

Recall from Section 5.1 that Wood (2006b) suggested two basic approaches to smoothing-parameter estimation. The first approach attempts to minimize the expected mean square error which leads to estimation by UBRE, while the second approach attempts to minimize prediction error which leads to estimation by GCV. In this research, we will adapt the arguments of the GCV score in the generalized case for univariate GAMs described in Wood (2004) to our current context as well. Following Wood (2004), given an estimate for $\boldsymbol{\beta}^{*(t)}$, smoothing parameter selection for the problem (5.16) can be achieved by minimization of the GCV score

$$\begin{aligned}\min_{\boldsymbol{\lambda}} \qquad \mathcal{V}_g\left(\boldsymbol{\lambda}\right) &= \frac{\widetilde{n}\left\|\,\mathbf{U}^{(t)}\left(\boldsymbol{z}^{(t)} - \mathbf{X}_{\text{VAM}}\,\boldsymbol{\beta}^*\right)\,\right\|^2}{\left[\,\widetilde{n} - \operatorname{tr}(\mathbf{A}_\lambda^{(t)})\,\right]^2}, \\ \boldsymbol{\lambda}^{(t+1)} &= \underset{\boldsymbol{\lambda}}{\operatorname{argmin}}\ \mathcal{V}_g\left(\boldsymbol{\lambda}\right). \end{aligned} \tag{5.22}$$

For the univariate GAMs, if the scale parameter is known, smoothing-parameter estimation can be achieved through the minimization of the UBRE but GCV is used otherwise. For most VGLMs used in practice, the scale parameter is $\phi = 1$ (cf. Yee (2015b, Chapter 3)) and does not have to be estimated.

As explained by Wood (2004), a penalized version of IRLS does not guarantee convergence in all circumstances, especially when the smoothing parameters have to be estimated. Wood (Chapter 4, 2006b) stated that one technical problem for performance iteration relates to divergence of the P-IRLS algorithm. This divergence can occasionally occur with all models, e.g., GLMs fitting using iteratively reweighted least squares but this can be always dealt with by repeatedly halving the length of the step until the step is found that actually increase the likelihood if divergence is detected. For performance iteration, this divergence cannot be detected easily. This is because model likelihood, GCV/UBRE score and penalized likelihood may all fairly increase may as well decrease from one iteration to the next.

The iteration must be examined in order to detect convergence. In VGAM, three obvious methods for doing this are to record relative changes in parameter estimates, absolute changes in the log-likelihood and absolute changes in deviance. Regarding the divergence issue for performance iteration, the first method is applicable to our problem. In VGAM, the exact criteria for testing the convergence of the coefficients is given by

$$\frac{\left|\beta^{*(t+1)}_{(j)k} - \beta^{*(t)}_{(j)k}\right|}{\texttt{epsilon} + \left|\beta^{*(t+1)}_{(j)k}\right|} < \texttt{epsilon} \qquad (\text{Yee, 2015b}) \tag{5.23}$$

for all $j = 1, \ldots, \text{ncol}(\mathbf{H}_k)$ and $k = 1, \ldots, p$. Note that the default for `epsilon` is `1e-7`, as given in `vglm.control()`. The convergence criteria in (5.23) will be adapted to our framework.

5.4 Simulation study

To gain insight into the practical effectiveness of the method that we developed, a simulation study was conducted to investigate the performance of the method proposed. We examined the performance of P-spline VGAMs by considering three multivariate response types and models involving semiparametric predictors, in which their model structures involve constraints on the model terms. Under a wide range of settings and using a number of test functions for these three models, P-spline VGAMs and VGAMs were compared in terms of predictive accuracy. The three models were as follows:

(i) a semiparametric bivariate probit model under a non-exchangeable structure and an intercept-only model for the correlation parameter ρ,

(ii) a semiparametric bivariate logistic model under a non-exchangeable structure and an intercept-only model for the odds ratio, and

(iii) a semiparametric bivariate logistic model under an exchangeable structure and an intercept-only model for the odds ratio.

P-spline VGAMs were fitted using the `psvgam()` function which implements the ideas proposed in Chapter 4 and Section 5.3, and VGAMs based on backfitting computation were fitted using the VGAM package. We measured and compared the performance of each test function for model (i) – (iii) using the root mean squared error (RMSE) defined as (cf. Marra et al. (2013b))

$$\text{RMSE}\left(\widehat{f}_{(j)k}(x_{ik})\right) = \left\{\frac{1}{n}\sum_{i=1}^{n}\left(\widehat{f}_{(j)k}(x_{ik}) - f_{(j)k}(x_{ik})\right)^2\right\}^{1/2}.$$

5.4.1 Semiparametric bivariate probit model

The standard bivariate probit model is a joint model for two binary responses introduced by Ashford and Sowden (1970). The model is given by

$$\begin{aligned}\Pr(Y_j = 1|\boldsymbol{x}) &= \Phi\{\eta_j(\boldsymbol{x})\}, \qquad j = 1,2,\\ \Pr(Y_1 = 1, Y_2 = 1|\boldsymbol{x}) &= \Phi_2\{\eta_1(\boldsymbol{x}), \eta_2(\boldsymbol{x}); \rho(\boldsymbol{x})\},\end{aligned} \tag{5.24}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal distribution and $\Phi_2(\cdot,\cdot;\rho)$ is the cumulative distribution function of a standard bivariate normal distribution with zero mean, unit variance and correlation coefficient ρ. The association between the responses is modeled via the parameter ρ. As $-1 < \rho(\boldsymbol{x}) < 1$, Yee and Wild (1996) used the default link for $\eta_3(\boldsymbol{x}) = \log\left((1+\rho(\boldsymbol{x}))\,/\,(1-\rho(\boldsymbol{x}))\right)$ and therefore they used

$$\rho\left(\boldsymbol{x}\right) = \frac{\exp\{\eta_3(\boldsymbol{x})\}-1}{\exp\{\eta_3(\boldsymbol{x})\}+1} \tag{5.25}$$

to satisfy the range restrictions. The additive model extension of (5.24) can be achieved by replacing all linear predictors of (5.24) with (cf. Yee and Wild (1996))

$$\eta_j\left(\boldsymbol{x}\right) = \sum_{k=1}^{p} f_{(j)k}\left(x_k\right), \qquad j = 1,\ldots,M.$$

The bivariate probit model can be simply expressed using the latent variable representation:

$$y^*_{1i} = \eta_1 + \varepsilon_{1i}, \tag{5.26}$$

$$y^*_{2i} = \eta_2 + \varepsilon_{2i}, \tag{5.27}$$

where the binary responses y_{1i} and y_{2i} are determined according to the rule:

$$y_{vi} = \begin{cases} 1 & \text{if } y^*_{vi} > 0 \\ 0 & \text{if } y^*_{vi} \leq 0 \end{cases}, v = 1,2. \tag{5.28}$$

The error terms $(\varepsilon_{1i}, \varepsilon_{2i})$ are identically distributed as bivariate normal with zero mean, unit variance and correlation coefficient ρ, independently across observations (e.g. Greene, 2007; Marra et al., 2013b):

$$\begin{pmatrix}\varepsilon_{1i}\\ \varepsilon_{2i}\end{pmatrix} \overset{iid}{\sim} N\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}\right). \tag{5.29}$$

5.4.1.1 Design and model fitting details

We conducted a simulation study based on equations (5.26) to (5.29) with

$$\eta_1(\boldsymbol{x}_i) = \beta_{(1)1} + \beta_{(1)2}\, x_{i2} + f_{(1)3}(x_{i3}) + f_{(1)4}(x_{i4}), \tag{5.30}$$

$$\eta_2(\boldsymbol{x}_i) = \beta_{(2)1} + \beta_{(2)2}\, x_{i2} + f_{(2)3}(x_{i3}) + f_{(2)4}(x_{i4}), \tag{5.31}$$

$$\eta_3(\boldsymbol{x}_i) = \beta_{(3)1}, \tag{5.32}$$

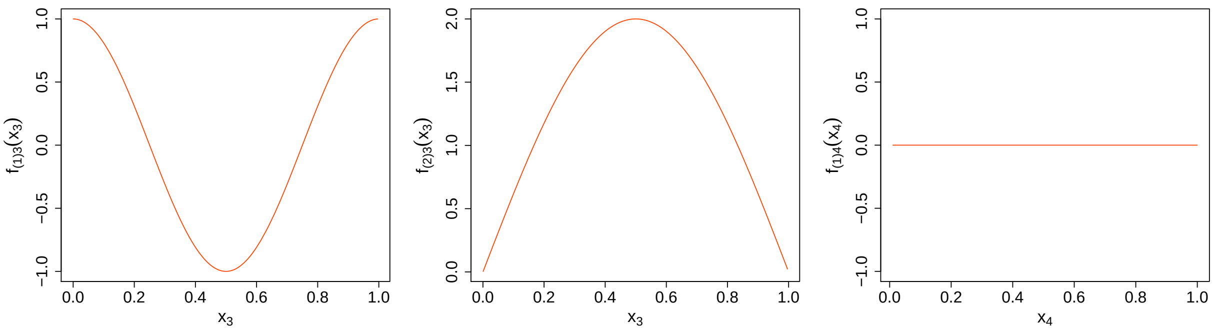


Figure 5.1: The three test functions used for the simulation study of a semiparametric bivariate probit model under a non-exchangeable structure and an intercept-only model for the correlation ρ.

where η_3 is related to correlation coefficient ρ through (5.25). The model (5.32) is a “non-exchangeable” structure in which the marginal probabilities are not the same and η_3 is modeled as a single parameter only, that is, ‘intercept-only’ for the correlation parameter ρ (cf. Yee (2015b)). We used the three test functions: $f_{(1)3}(x_{i3}) = \cos(2\pi x_{i3})$, $f_{(2)3}(x_{i3}) = 2\sin(\pi x_{i4})$, and $f_{(1)4}(x_{i4}) = f_{(2)4}(x_{i4}) = 0$, displayed in Fig. 5.1. These functions are often used by Wood (2006b) in simulations. Covariates were simulated independently from a uniform distribution on $(0, 1)$. Bivariate normal correlated errors $(\varepsilon_{1i}, \varepsilon_{2i})$ were generated using the `rmvnorm()` function from the mvtnorm package (Genz et al., 2008).

In this simulation study, we considered three different correlation levels $\rho = (0.1, 0.5, 0.9)$, and three different sample sizes: 1000, 2000 and 3000. For given $\boldsymbol{x}$-values and the correlation ρ, we generated bivariate binary observations for (5.30) and (5.31) with the intercepts $(\beta_{(1)1}, \beta_{(2)1}) = (-1.55, -0.25)$ and the linear-term coefficients $(\beta_{(1)2}, \beta_{(2)2}) = (2, -1.25)$. A total 500 replicate data sets were generated. P-spline VGAMs were then fitted to each of 500 replicates at each sample size and correlation combination. For P-spline VGAMs, the smooth terms for (5.30) and (5.31) were estimated using penalized B-splines of degree 3, together with a second order penalty, and 8 equally spaced B-spline knots with the smoothing parameters being selected automatically through minimization of the UBRE score as described in Section 5.3. VGAMs based on backfitting were fitted to the same data-generating process described above using the vector cubic smoothing splines with the default 4 degrees of freedom for each smooth term. For each replicate and for each test function, we calculated the RMSE of the estimated smooth functions of interest.

5.4.1.2 Results

The simulation results are summarized in the following figures. Boxplots of the estimates of the linear-term coefficients $\beta_{(1)2}$ and $\beta_{(2)2}$ are given in Figs. 5.2 and 5.3 respectively, while those in Fig. 5.4 plot the estimates of the correlation parameter ρ. Fig. 5.5 shows the boxplots of the RMSEs of the estimated smooth functions $\widehat{f}_{(1)3}$, $\widehat{f}_{(2)3}$, $\widehat{f}_{(1)4}$ and $\widehat{f}_{(2)4}$ based on 500 replications when employing P-spline VGAMs at each sample size and correlation combination. Figs. 5.6 – 5.8 display the boxplots of the RMSEs of the estimated smooth functions $\widehat{f}_{(1)3}$, $\widehat{f}_{(2)3}$ and $\widehat{f}_{(1)4}$ comparing P-spline VGAMs and VGAMs. The plots of the RMSEs of the estimated smooth function $\widehat{f}_{(2)4}$ are omitted as they lead to the same conclusions as that of $\widehat{f}_{(1)4}$. The results can be summarized as follows:

(i) From Figs. 5.2 – 5.4, for the estimates of the linear-term coefficients $\beta_{(1)2}$ and $\beta_{(2)2}$, and the correlations ρ, the two methods perform very similarly. There is very little evidence of bias and the variability in estimate reduces with sample size as expected.

(ii) From Fig. 5.5, the RMSEs obtained using P-spline VGAMs are largely unaffected by the correlation level and get smaller as the sample size increases regardless of correlation (the reductions in RMSE are small for $\widehat{f}_{(1)3}$.).

(iii) From Figs. 5.6 – 5.8, the RMSEs for the estimated smooth functions $\widehat{f}_{(1)3}$, $\widehat{f}_{(2)3}$, and $\widehat{f}_{(1)4}$ indicate that the P-spline VGAM approach outperforms the default VGAM method, at all correlations, sample sizes, for these test functions. With the exception of the fitted curve $\widehat{f}_{(1)4}$, the differences in performance are greater at higher correlations. In all cases, a Wilcoxon signed rank test yields a p-value of $< 10^{-16}$. The effective degrees of freedom used by the P-spline VGAMs does appear to adapt to the data as desired.

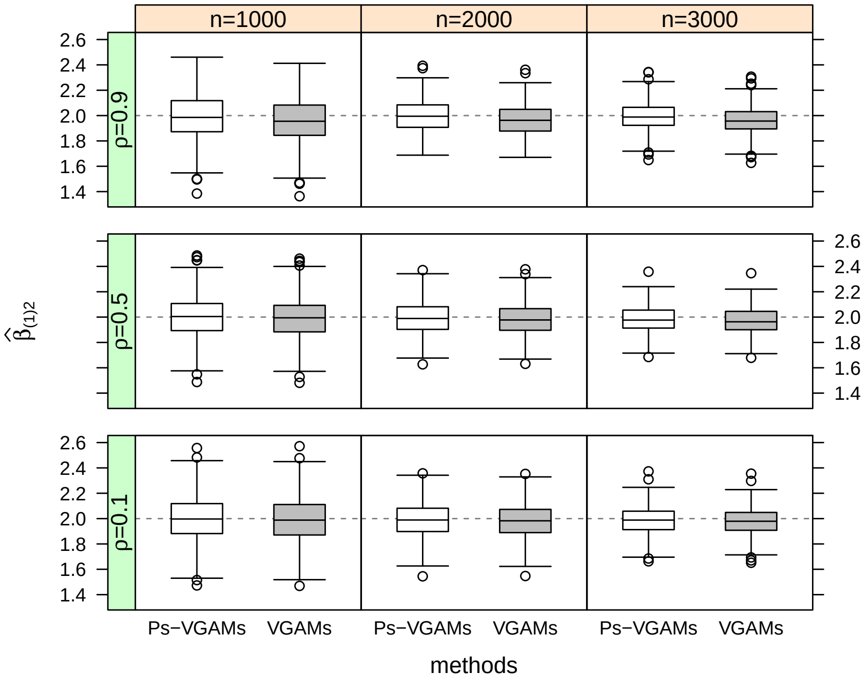


Figure 5.2: Boxplots of the estimates of the linear-term coefficient $\beta_{(1)2}$ when employing P-spline VGAMs and VGAMs. The true value (dashed lines) is 2. The parameter ρ denotes the correlation coefficients.

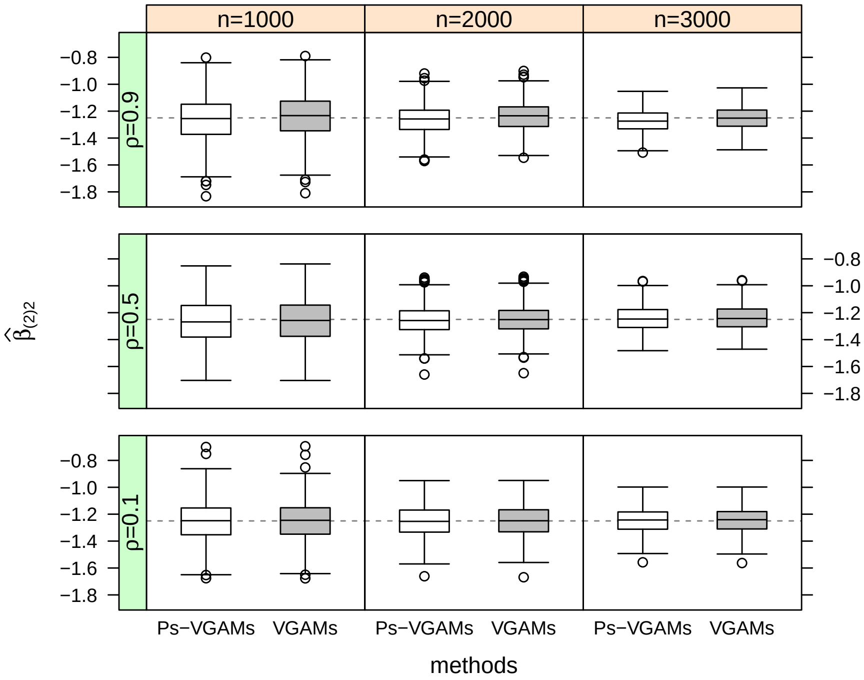


Figure 5.3: Boxplots of the estimates of the linear-term coefficient $\beta_{(2)2}$ when employing P-spline VGAMs and VGAMs. The true value (dashed lines) is -1.25.

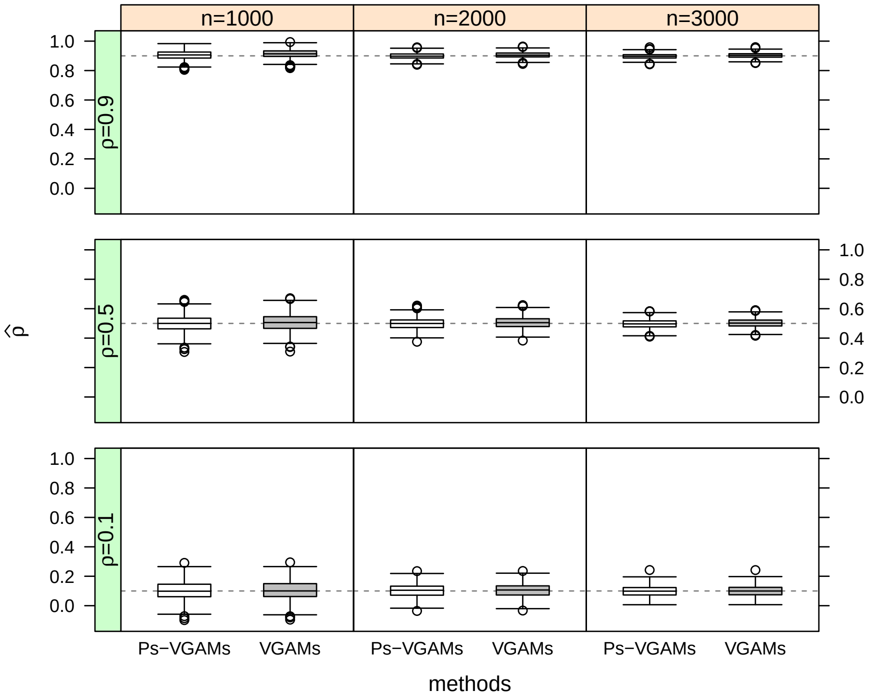


Figure 5.4: Boxplots of the estimates of the correlation parameter ρ when employing P-spline VGAMs and VGAMs. The true values are indicated by dashed lines.

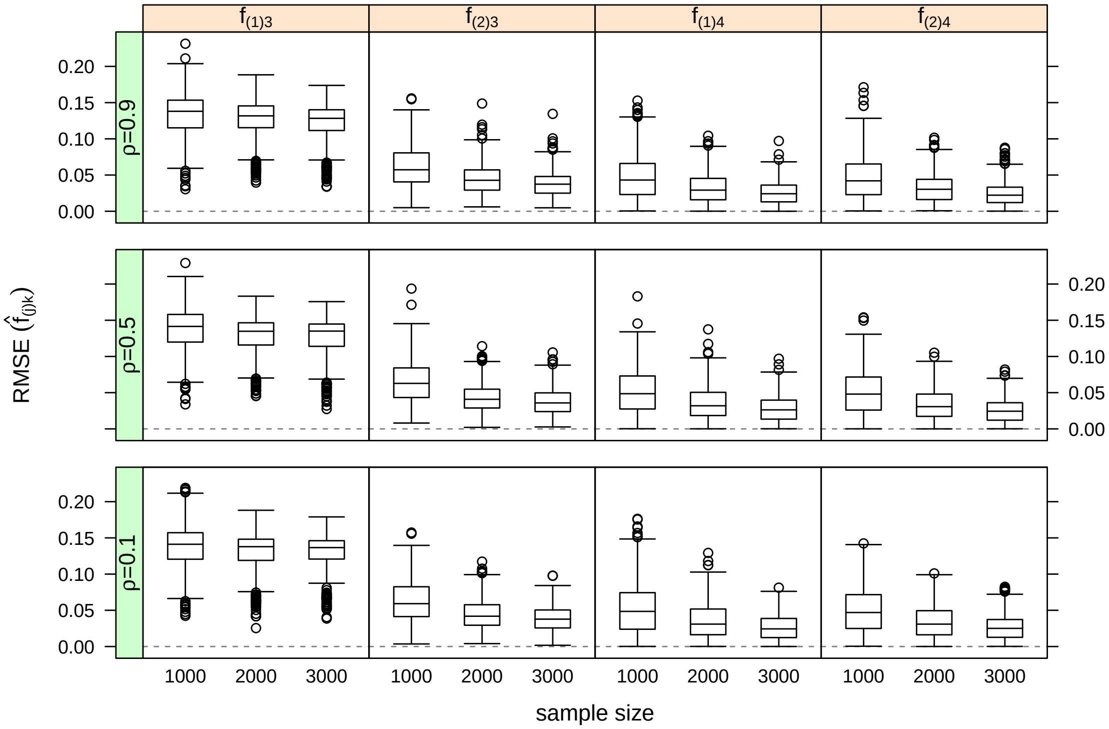


Figure 5.5: Boxplots of the RMSEs of the estimated smooth functions $\widehat{f}_{(1)3}, \widehat{f}_{(2)3}, \widehat{f}_{(1)4}$, and $\widehat{f}_{(2)4}$ when employing P-spline VGAMs.

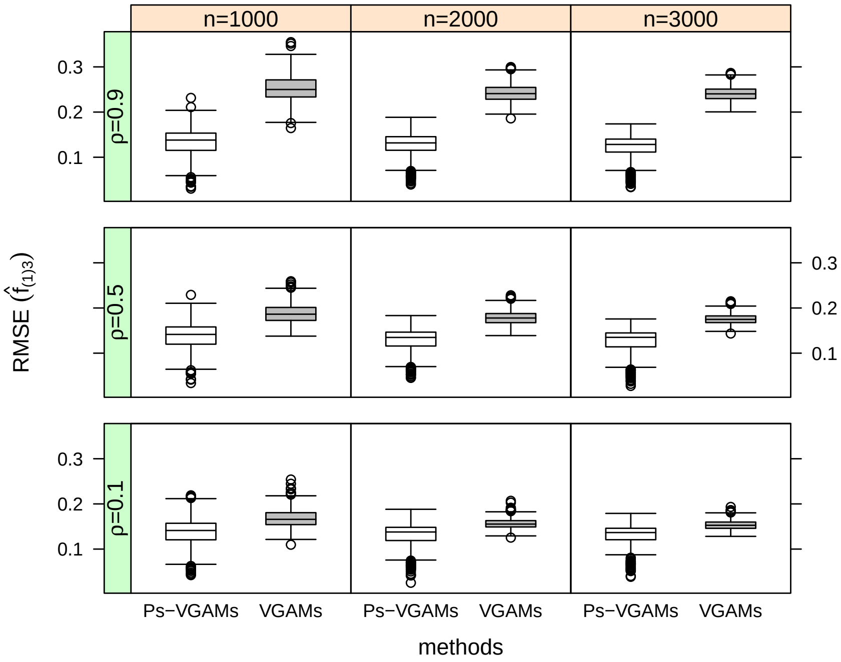


Figure 5.6: Boxplots of the RMSEs of the estimated smooth function $\widehat{f}_{(1)3}$ when employing P-spline VGAMs and VGAMs.

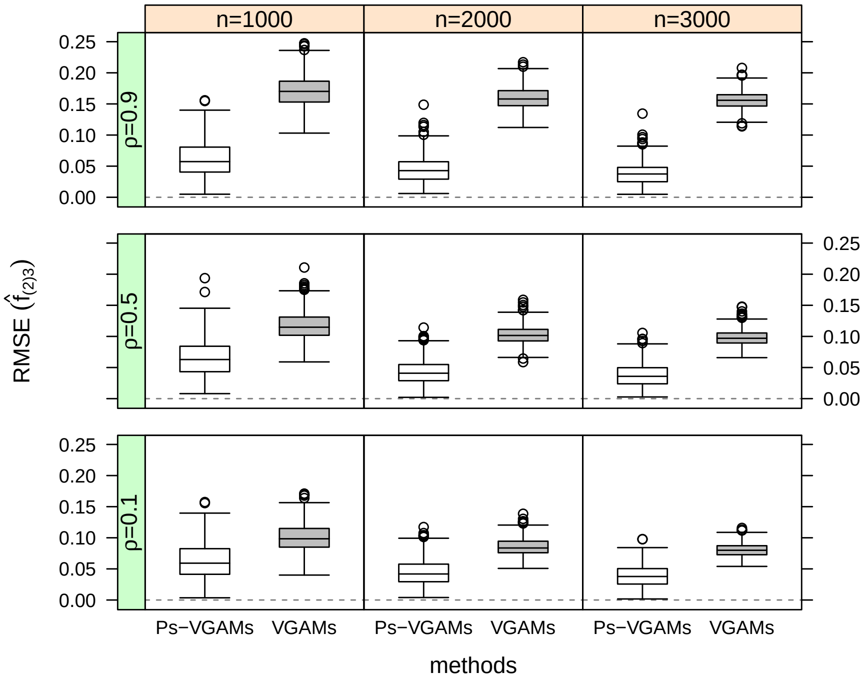


Figure 5.7: Boxplots of the RMSEs of the estimated smooth function $\widehat{f}_{(2)3}$ when employing P-spline VGAMs and VGAMs.

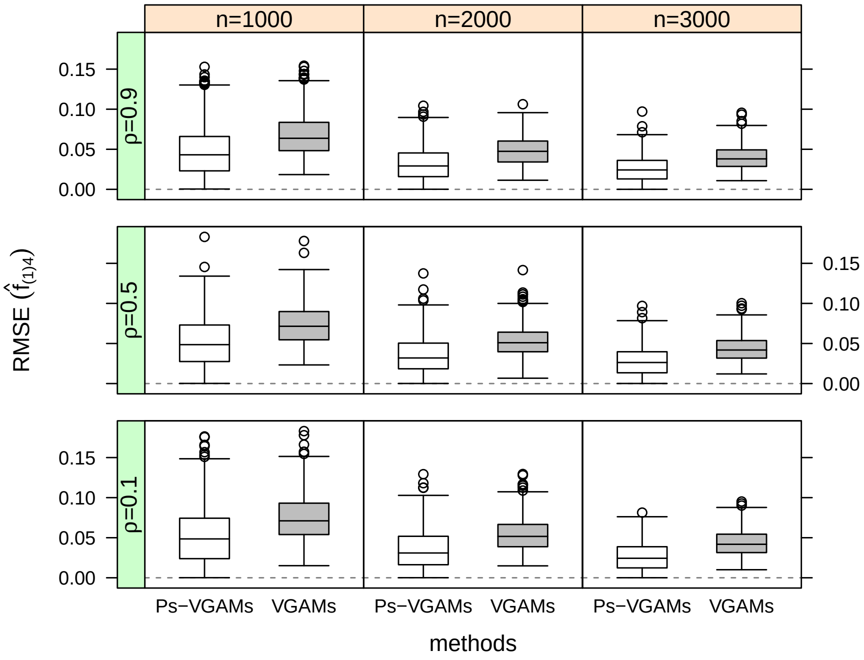


Figure 5.8: Boxplots of the RMSEs of the estimated smooth function $\widehat{f}_{(1)4}$ when employing P-spline VGAMs and VGAMs.

5.4.2 Non-exchangeable semiparametric bivariate logistic model

In this section we discuss simulations for the cases in which the responses of the model are generally dependent and the association between the two binary responses is modeled in terms of the odds ratio. A natural regression model here is the bivariate logistic model discussed in Section 3.3.2. The non-exchangeable model is specified in the form of

$$\begin{aligned} \operatorname{logit} p_j(\boldsymbol{x}) &= \eta_j(\boldsymbol{x}), \qquad j = 1, 2, \\ \log \psi(\boldsymbol{x}) &= \eta_3(\boldsymbol{x}), \end{aligned} \tag{5.33}$$

where ψ is the odds ratio.

5.4.2.1 Design and model fitting details

We conducted a simulation study based on the bivariate logistic model (5.33) with:

$$\begin{aligned}\eta_1 &= \operatorname{logit} P\left(Y_1 = 1|\boldsymbol{x}_i\right) = \beta_{(1)1} + \beta_{(1)2}x_{i2} + f_{(1)3}(x_{i3}) + f_{(1)4}(x_{i4}),\\ \eta_2 &= \operatorname{logit} P\left(Y_2 = 1|\boldsymbol{x}_i\right) = \beta_{(2)1} + \beta_{(2)2}x_{i2} + f_{(2)3}(x_{i3}) + f_{(2)4}(x_{i4}),\\ \eta_3 &= \log\, \psi(\boldsymbol{x}_i) = \beta_{(3)1}. \end{aligned} \tag{5.34}$$

The model (5.34) taken here has a "non-exchangeable" structure in which no constraints are imposed on η_1 and η_2, and η_3 is modeled as an intercept-only for the odds ratio (see equation (3.28)). Following Wood (2006b) and Marra et al. (2013b), we used:

$$f_{(1)3}(x_{i3}) = \cos(2\pi x_{i3}), \tag{5.35}$$

$$f_{(2)3}(x_{i3}) = 2\sin(\pi x_{i3}), \tag{5.36}$$

$$f_{(1)4}(x_{i4}) = -0.7\left(4x_{i4} + 2.5x_{i4}^2 + 0.7\sin(5x_{i4}) + \cos(7.5x_{i4})\right), \tag{5.37}$$

$$f_{(2)4}(x_{i4}) = \exp(2x_{i4}) - 3.75. \tag{5.38}$$

These are plotted in Fig. 5.9. We used three different levels of odds ratio $\psi = (1.5, 2, 3)$ and three different sample sizes: 1000, 2000 and 3000. We set the intercepts $(\beta_{(1)1},\ \beta_{(2)1}) = (-1, -2)$, and the linear-term coefficients $(\beta_{(1)2},\ \beta_{(2)2}) = (1.5, 1)$. Covariates were simulated independently from a uniform distribution on $(0, 1)$. The linear predictors were made up of a sum of linear terms and smooth terms of the form (5.34), and applied to the simulated covariates to give the true linear predictor. The inverse of link for model (5.34) was applied to the linear predictors to give the true response means, and then data were simulated from a bivariate binary regression model using an odds ratio as the measure of dependency. These data were generated using the function `rbinom2.or()` from the VGAM package. We fitted the model using the data generating process described above with the two estimation frameworks to each of 500 replicates at each sample size, and odds ratio combination. Again, the non-linear terms for P-spline VGAMs were based on B-splines of degree 3, a second order penalty, and 8 equally-spaced B-spline knots with the smoothing parameters being estimated by minimizing the UBRE score. VGAMs

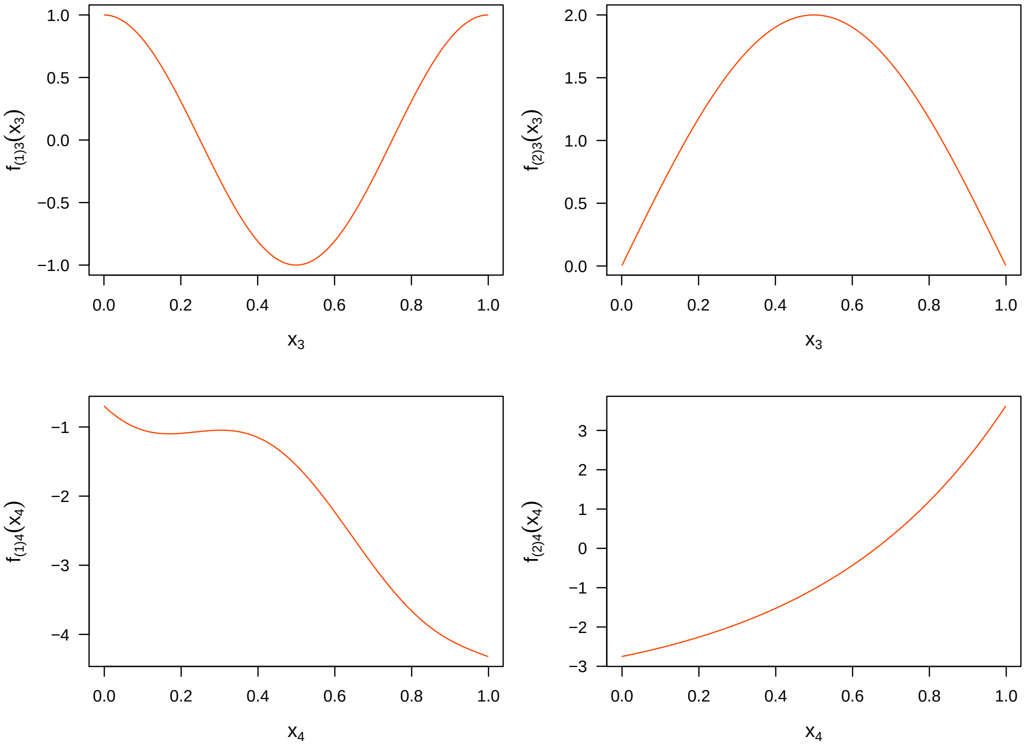


Figure 5.9: The four test functions used for the simulation study of a semiparametric bivariate logistic model under a non-exchangeable structure and an intercept-only model for the odds ratio (see equations (5.35) – (5.38)).

were fitted to the same data described above using the default settings. For each dataset and estimation procedure, we obtained the estimates of the linear-term coefficients, the odds ratio and the RMSEs for the estimated smooth function of the variable of interest.

5.4.2.2 Results

The simulation results are summarized in the following figures. Boxplots of the estimates of the linear-term coefficients $\beta_{(1)2}$ and $\beta_{(2)2}$ are given respectively in Figs. 5.10 and 5.11 and those of the odds ratio ψ are shown in Fig. 5.12. Fig. 5.13 shows the boxplots of the RMSEs of the estimated smooth functions for $f_{(1)3}$, $f_{(2)3}$, $f_{(1)4}$, and $f_{(2)4}$ when employing the P-spline VGAM approach at each sample size and odds ratio combination. Figs. 5.14 – 5.17 show the boxplots of the RMSEs of the estimated smooth functions $\widehat{f}_{(1)3}$, $\widehat{f}_{(2)3}$, $\widehat{f}_{(1)4}$, and $\widehat{f}_{(2)4}$ comparing P-spline VGAMs and VGAMs. The results can be summarized as follows:

(i) From Figs. 5.10 – 5.11, for the estimates of the linear-term coefficients $\beta_{(1)2}$ and $\beta_{(2)2}$, and the odds ratio ψ, the two methods perform very similarly. There is again very little evidence of bias and the variability of the estimates become smaller as the sample size increases.

(ii) From Fig. 5.13, we see that the RMSEs obtained using the method proposed are largely unaffected by the levels of the odds ratio and become smaller as the sample size n increases irrespective of the odds ratio.

(iii) From Figs. 5.14 – 5.17, the RMSEs for the estimated smooth functions $\widehat{f}_{(1)3}$, $\widehat{f}_{(2)3}$, $\widehat{f}_{(1)4}$ and $\widehat{f}_{(2)4}$ indicate that P-spline VGAMs perform significantly better than the default VGAM method, at all levels of odds ratio, sample sizes, for these test functions. In all cases, a Wilcoxon signed rank test yields a p-value of $< 10^{-16}$. The results again show that the automatic choice of degree of smoothness obtained using P-spline VGAMs does better capture the shape of non-linear terms from the data as desired.

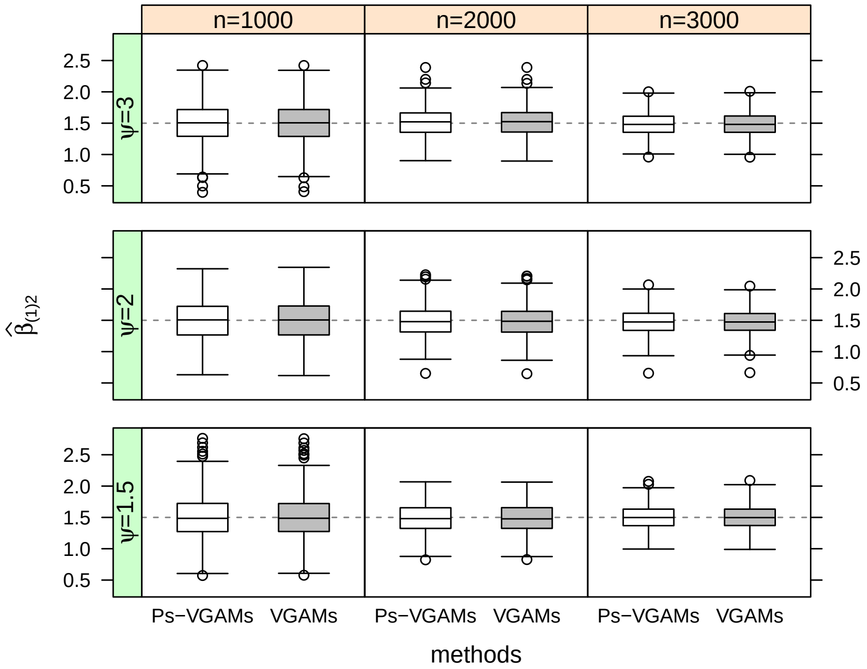


Figure 5.10: Boxplots of the estimates of the linear-term coefficient $\beta_{(1)2}$ when employing P-spline VGAMs and VGAMs. The true value (dashed lines) is 1.5. The parameter ψ denotes the odds ratio.

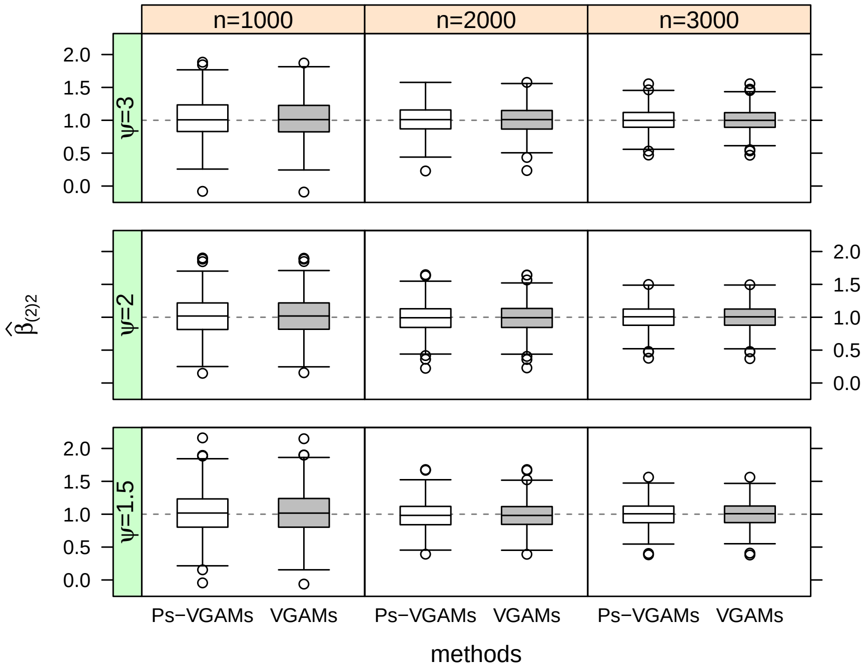


Figure 5.11: Boxplots of the estimates of the linear-term coefficient $\beta_{(2)2}$ when employing P-spline VGAMs and VGAMs. The true value (dashed lines) is 1.

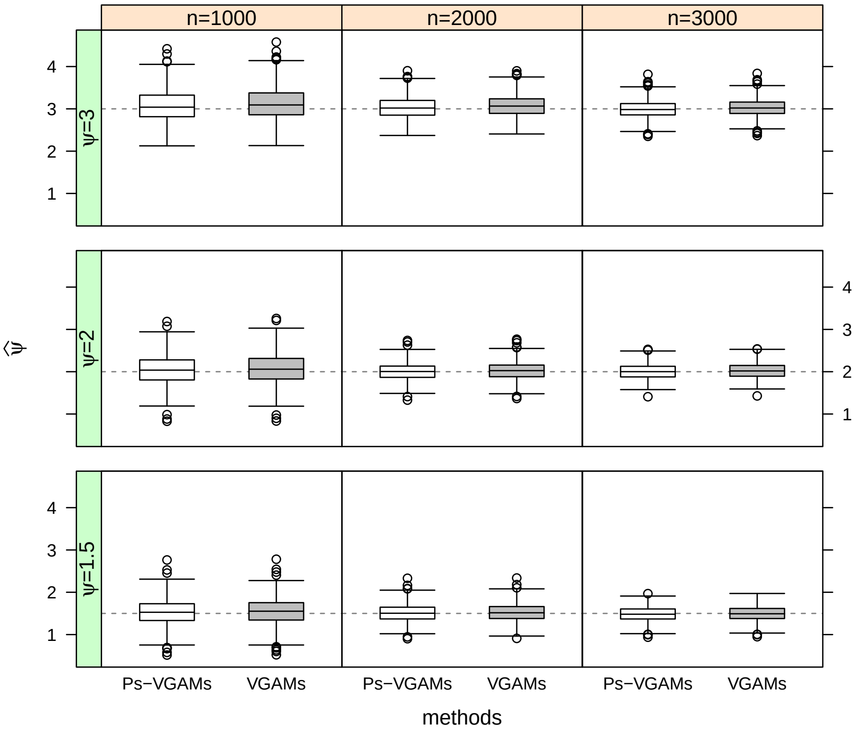


Figure 5.12: Boxplots of the estimates of the odds ratio ψ when employing P-spline VGAMs and VGAMs. The true values are indicated by dashed lines.

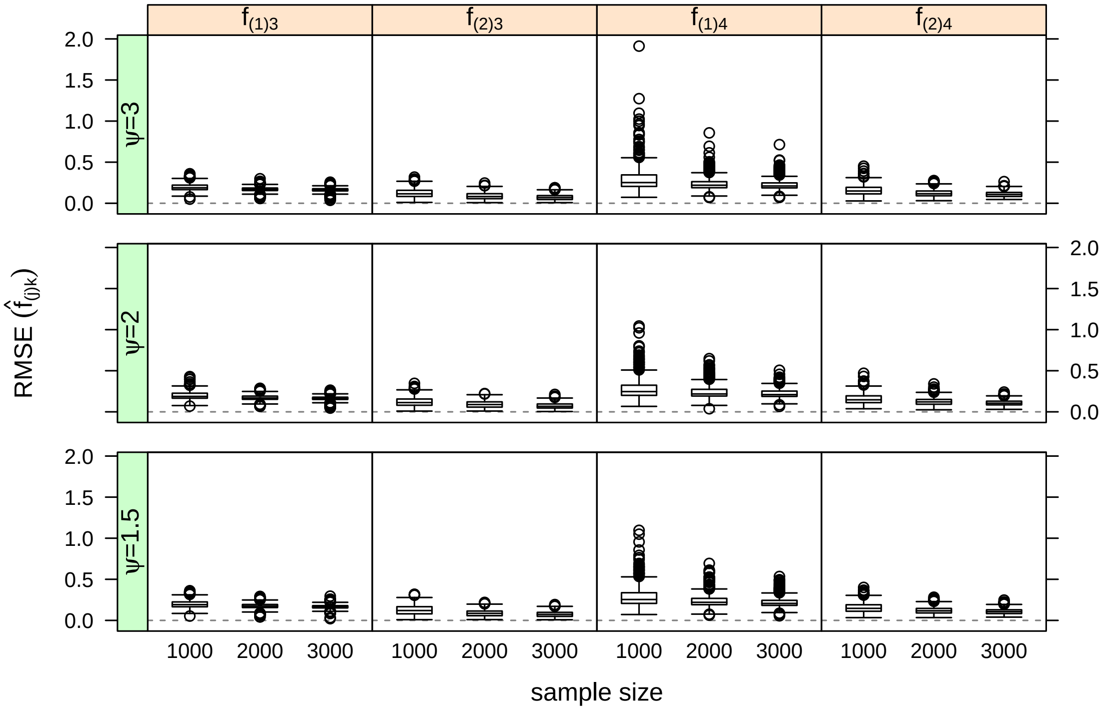


Figure 5.13: Boxplots of the RMSEs of the estimated smooth functions $\widehat{f}_{(1)3}, \widehat{f}_{(2)3}, \widehat{f}_{(1)4}$ and $\widehat{f}_{(2)4}$ when employing P-spline VGAMs.

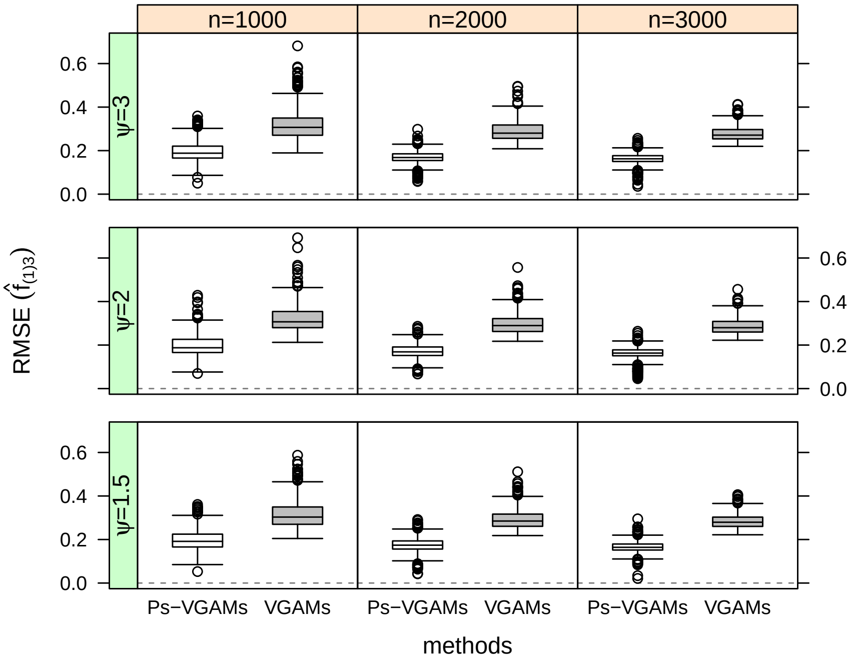


Figure 5.14: Boxplots of the RMSEs of the estimated smooth function $\widehat{f}_{(1)3}$ when employing P-spline VGAMs and VGAMs.

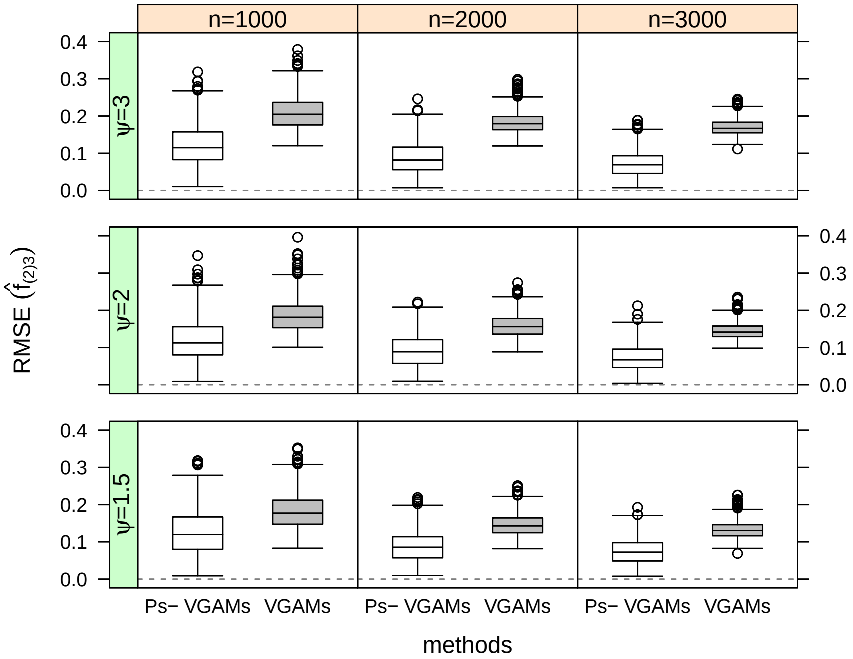


Figure 5.15: Boxplots of the RMSEs of the estimated smooth function $\widehat{f}_{(2)3}$ when employing P-spline VGAMs and VGAMs.

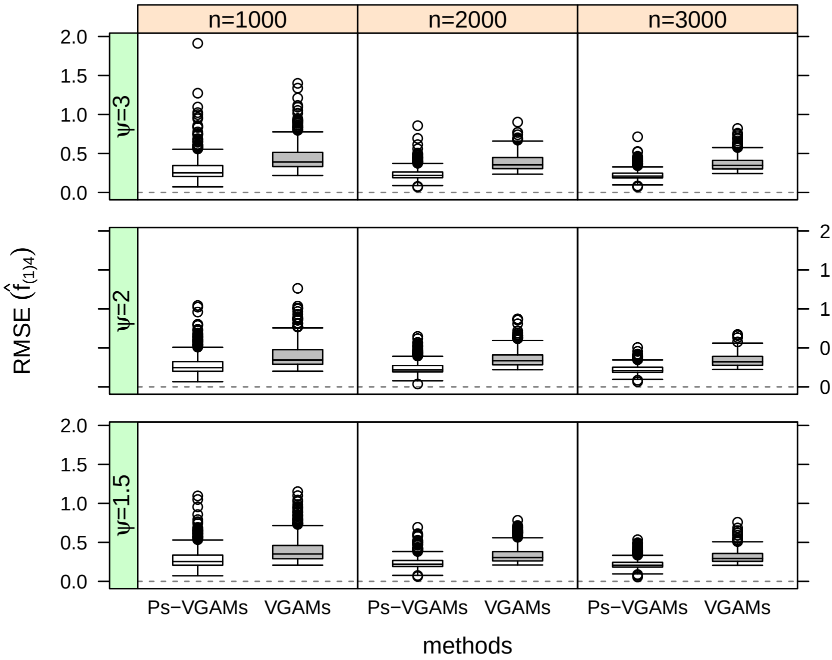


Figure 5.16: Boxplots of the RMSEs of the estimated smooth function $\widehat{f}_{(1)4}$ when employing P-spline VGAMs and VGAMs.

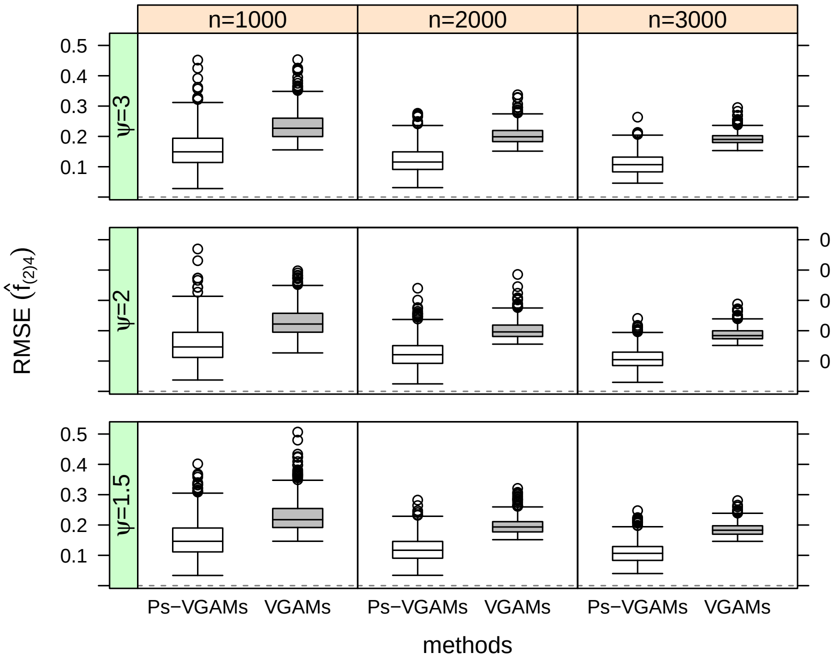


Figure 5.17: Boxplots of the RMSEs of the estimated smooth function $\widehat{f}_{(2)4}$ when employing P-spline VGAMs and VGAMs.

5.4.3 Exchangeable semiparametric bivariate logistic model

The last section investigated the non-exchangeable version of the semiparametric bivariate logistic model. In this section we investigate the exchangeable version.

5.4.3.1 Design and model fitting details

Our simulation study was based on the model:

$$\begin{aligned} \eta_1 &= \operatorname{logit} P\,(Y_1 \;=\; 1|\boldsymbol{x}_i) \;=\; \beta_{(1)1} + \beta_{(1)2}x_{i2} + f_{(1)3}(x_{i3}) + f_{(1)4}(x_{i4}), \\ \eta_2 &= \operatorname{logit} P\,(Y_2 \;=\; 1|\boldsymbol{x}_i) \;=\; \beta_{(1)1} + \beta_{(1)2}x_{i2} + f_{(1)3}(x_{i3}) + f_{(1)4}(x_{i4}), \\ \eta_3 &= \log\,\psi(\boldsymbol{x}_i) \;=\; \beta_{(2)1}, \end{aligned} \tag{5.39}$$

with

$$f_{(1)3}(x_{i3}) \;=\; \cos(2\pi x_{i3}), \tag{5.40}$$

$$f_{(1)4}(x_{i4}) \;=\; -0.7\left(4x_{i4} + 2.5x_{i4}^2 + 0.7\sin(5x_{i4}) + \cos(7.5x_{i4})\right). \tag{5.41}$$

These are plotted in Fig. 5.18. We again used three different levels of odds ratio and three different sample sizes as described in Section 5.4.2.1. We set the intercept $\beta_{(1)1} = -1$ and the linear-term coefficients $\beta_{(1)2} = 1.5$. Both P-spline VGAMs and default VGAMs were again compared in terms of the RMSEs in predicting the test functions.

5.4.3.2 Results

The simulation results are summarized in the following figures. Boxplots of the estimates of the linear-term coefficient $\beta_{(1)2}$ and those of the odds ratio ψ are given respectively in Figs. 5.19 and 5.20. Fig. 5.21 shows the boxplots of the RMSEs of the estimated smooth functions for $f_{(1)3}$ and $f_{(1)4}$ when employing the P-spline VGAM approach at each sample size and odds ratio combination. Figs. 5.22 – 5.23 show the boxplots of the RMSEs of the estimated smooth functions $\widehat{f}_{(1)3}$ and $\widehat{f}_{(1)4}$ comparing P-spline VGAMs and VGAMs. Overall, our findings are much the same as for the previous section (as might be expected).

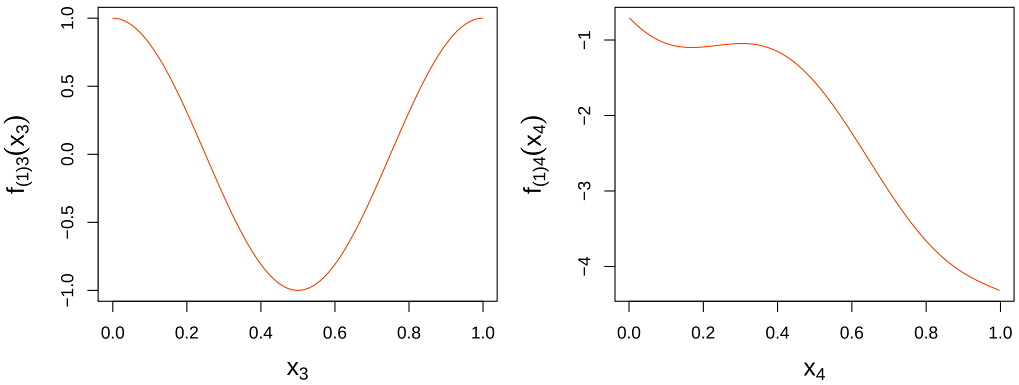


Figure 5.18: The two test functions used for the simulation study of a semiparametric bivariate logistic model under an exchangeable structure and an intercept-only model for the odds ratio (see equations (5.40) – (5.41)).

(i) From Figs. 5.19 – 5.20, for the estimates of the linear-term coefficient $\beta_{(1)2}$ and the odds ratio ψ, the two methods perform very similarly. There is very little evidence of bias and the variability in estimate becomes smaller with sample size as expected.

(ii) From Fig. 5.21, the RMSEs of the estimated smooth functions obtained using P-spline VGAMs become smaller as the sample size increases regardless of the odds ratio.

(iii) From Figs. 5.22 – 5.23, the RMSEs for the estimated smooth functions $\widehat{f}_{(1)3}$ and $\widehat{f}_{(1)4}$ indicate that P-spline VGAMs perform better than the default VGAM method, at all levels of odds ratio, sample sizes, for these test functions. In all cases, a Wilcoxon signed rank test yields a p-value of $< 10^{-16}$.

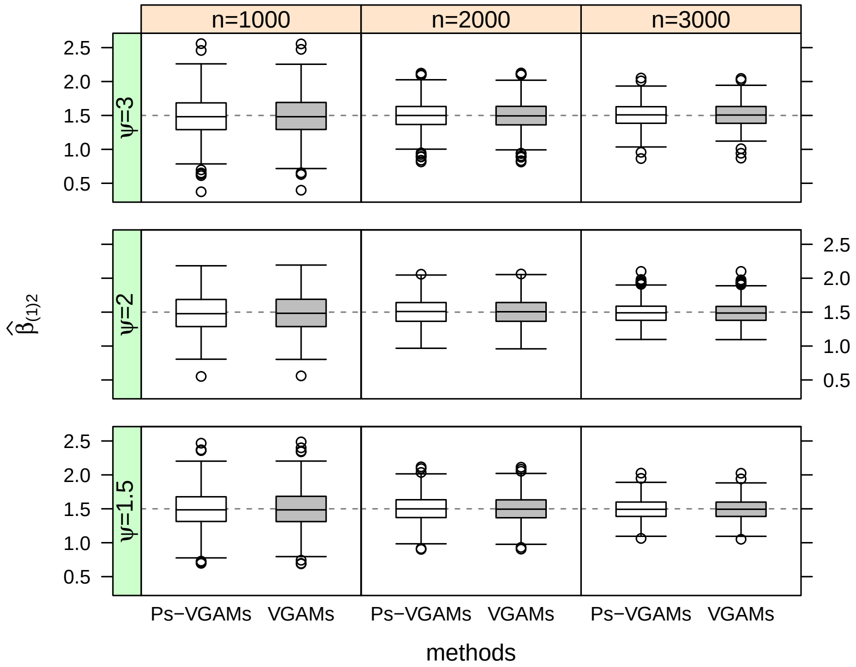


Figure 5.19: Boxplots of the estimates of the linear-term coefficient $\beta_{(1)2}$ when employing P-spline VGAMs and VGAMs. The true value (dashed lines) is 1.5. The parameter ψ denotes the odds ratio.

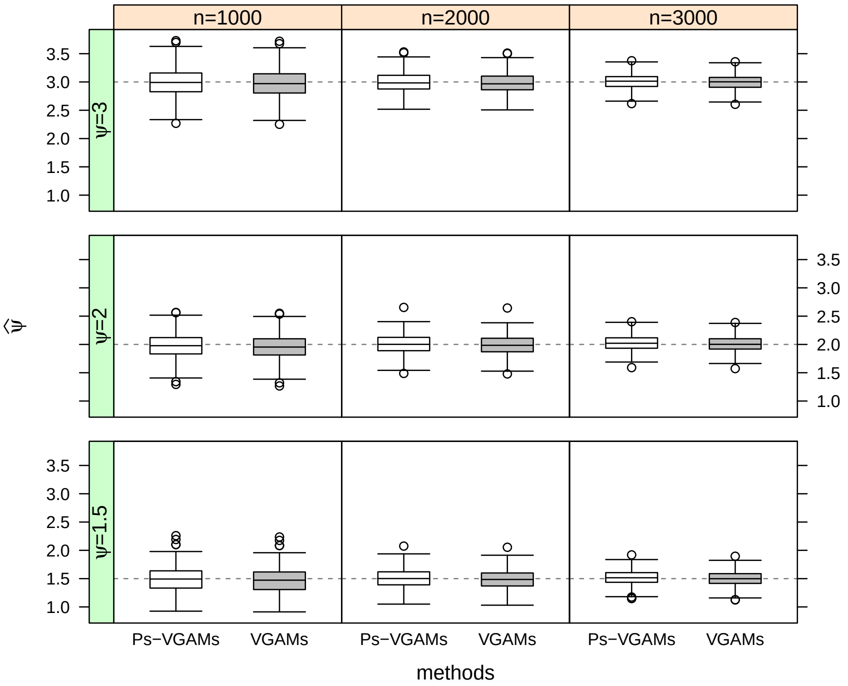


Figure 5.20: Boxplots of the estimates of the odds ratio, ψ, when employing P-spline VGAMs and VGAMs. The true values are indicated by dashed lines.

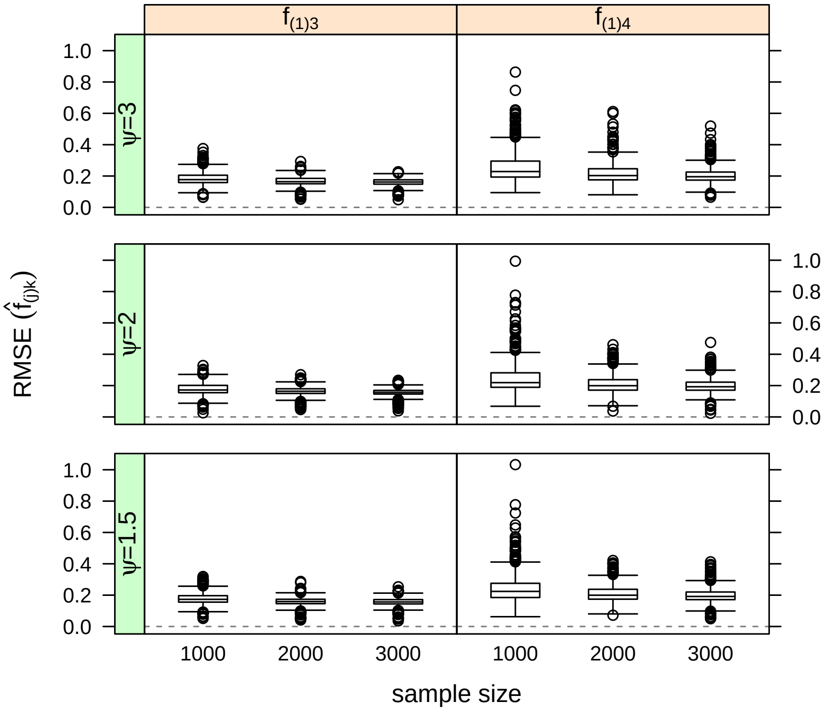


Figure 5.21: Boxplots of the RMSEs of the estimated smooth functions $\widehat{f}_{(1)3}$ and $\widehat{f}_{(1)4}$ when employing P-spline VGAMs.

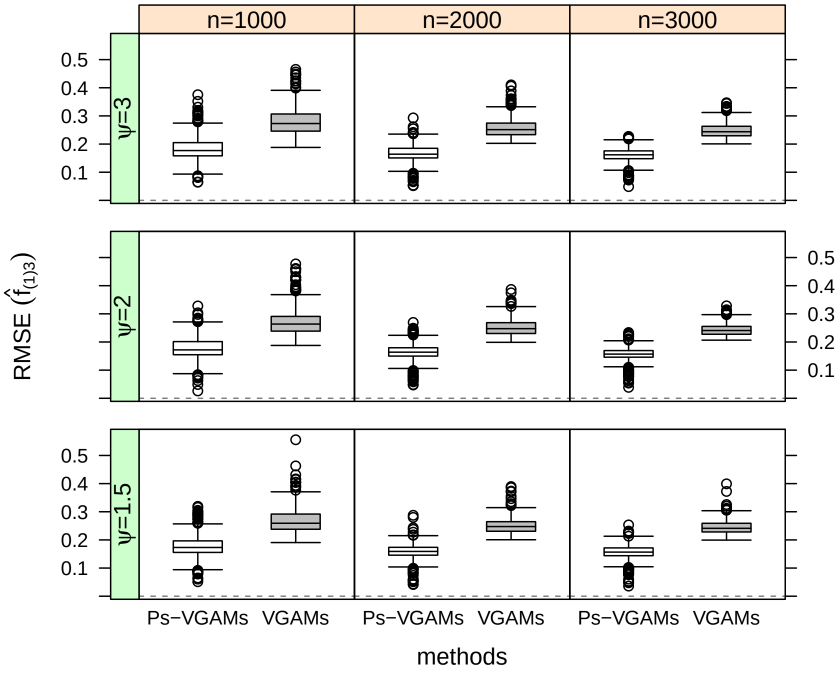


Figure 5.22: Boxplots of the RMSEs of the estimated smooth function $\widehat{f}_{(1)3}$ when employing P-spline VGAMs and VGAMs.

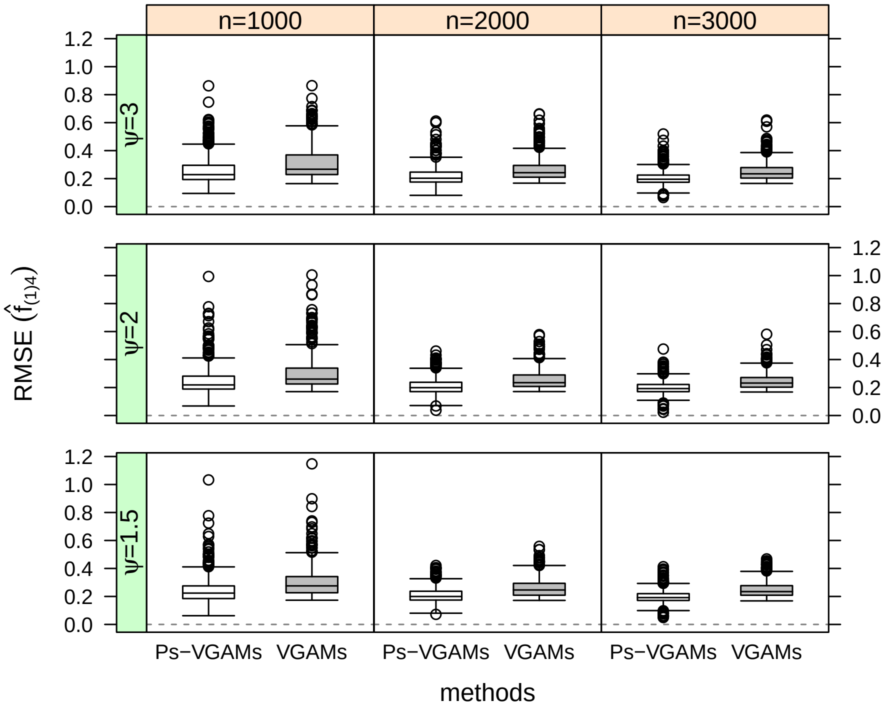


Figure 5.23: Boxplots of the RMSEs of the estimated smooth function $\widehat{f}_{(1)4}$ when employing P-spline VGAMs and VGAMs.

5.5 Conclusions

This chapter discussed the problem of incorporating the automatic estimation of smoothing parameters in P-spline VGAMs. We do this using GCV or the UBRE, adapting the methods of Wood (2004) and Marra and Radice (2011). We have also discussed implementation details for our methods which we have coded as R functions (see appendix A).

We then conducted a simulation study to compare the performance of our implementation of P-spline VGAMs to default VGAMs related models. The automatic smoothness estimation performed well in these simulations in terms of estimating smooth components for model structures involving constraints on the model terms and that did not. The P-spline VGAM approach performs significantly better than the default VGAM method in terms of predictive accuracy. The new methods do appear to adapt well to simplicity or complexity in the effects of covariates.

One limitation of the P-spline VGAMs that has not been reported in this chapter so far is that this method may suffer occasionally from convergence problems, especially at small sample sizes. For example we have had some problems for sample sizes under 500 for the simulation study of the non-exchangeable semiparametric bivariate probit model, especially for fairly high correlations between two responses (cf. Section 5.4.1). Similar convergence problems were also reported by Freedman and Sekhon (2010) and Marra and Radice (2011).

Chapter 6

Applications

In this chapter, we apply P-spline VGAMs to two real datasets with multivariate response types and models, and show some of the scope of P-spline VGAMs for additive modeling. We will use several statistical models: the multinomial logit, proportional and non-proportional odds models, bivariate logistic model, and the LMS method for quantile regression. The two data sets used are: (i) the `xs.nz` data frame in the VGAMdata package which contains data from a cross-sectional workforce study combined with a health survey from New Zealand during the 1990s, and (ii) the `birth` data frame in the catdata package which contains data from a survey study of the pregnancy and birth process during $1990-2004$. We will compare fitted models from VGAMs and P-spline VGAMs. Inference based on the deviance test described in section 4.6 is used for comparing models. Hypothesis testing used the two types of approximation from Wood (2006b, chapter 5), that is, the tests are based on estimated smoothing parameters, and treating the P-spline VGAM penalized fits as if they were un-penalized fits, with degrees of freedom given by the effective degrees of freedom of the models.

6.1 Some implementation details

Before moving on to the applications discussed above, we will briefly review implementation details for the P-spline VGAM approach. All necessary functions to construct P-spline VGAMs are written in R. P-spline VGAMs are based methodologically on penalized regression splines using P-spline smoothers of Marx and Eilers (1998) and the VGLM/VGAM framework. Therefore, the underlying functions for fitting P-spline VGAMs are mainly developed from P-spline GAMs and VGLM/VGAM frameworks. The primary function `psvgam()` and the support functions, namely, `psvglm.fit()` and `psvlm.wfit()` were adapted from `vgam()`, `vglm()`, `vglm.fit()` and `vlm.wfit()` in the VGAM package, and have similar functionality. The call sequence of the `psvgam()` function is:

```
psvgam <- function(formula, family, data, weights, etastart,
                   mustart, coefstart,
                   control = vglm.control(maxit = 50,...),
                   constraints method = "psvglm.fit", ...)
```

The `formula` argument is the most interesting. It has the form `(vector) response` ∼ `(linear/additive predictor)` and provides a symbolic description of the model to be fit. The RHS of the formula is applied to each linear/additive predictor. By default, constraint matrices are the $M \times M$ identity matrix unless arguments in the family function itself override these values, e.g., parallelism and exchangeability. In the P-spline VGAM framework, we use a new function called `ps()` adapted from Marx and Eilers's (1998) `ps()` function, in the definition of (vector) smooth terms within `psvgam()` formulas. The `ps()` function for P-spline VGAMs has call sequence:

```
ps <- function(x, ps.intervals = NULL, lambda = 0, degree = 2,
               order = 2, ridge.adj = 1e-005, ridge.inv = 1e-004)
```

The arguments in the `ps()` function are defined in the same way as the `ps()` function in P-spline GAMs, except that the argument `lambda` in the `ps()` of P-spline VGAMs allows for smoothing parameters, which are in the form of either scalar or vectors. To illustrate the usage of `psvgam()`, the nonparametric proportional-odds model (McCullagh and Nelder, 1989, p.179) can be fitted to the pneumoconiosis data from the `pneumo` data frame in the VGAM package as in the expression:

```
pneumo <- transform(pneumo, let = log(exposure.time))
fit <- psvgam(cbind(normal, mild, severe) ~ ps(let, ps.interval = 5),
              family = cumulative(reverse = TRUE, parallel = TRUE),
              data = pneumo)
```

The term on the RHS of the `formula` notation of the model above indicates that P-splines are used in a penalized maximization likelihood to fit the variable `let`, where the number of knots is given by `ps.interval = 5`. B-splines of degree 3 with 2^{nd} order difference penalty are used as the default settings, while smoothing parameters are chosen automatically through the minimization of the UBRE score. Further details of codes and implementation are given in appendix A.

6.2 Multinomial logit fits to marital status data

The multinomial logit model was introduced by Nerlove and Press (1973). This model is a generalization of logistic regression to model categorical responses with more than two categories, and is commonly used when the responses are “nominal” (consisting of unordered categories). The model is used to predict the probability of falling in each of the levels of the “nominal” response, given a set of independent variables. In VGAMs, the multinomial logit models the probability of Y falling into category j when there are $(M+1)$ categories as follows (Yee and Mackenzie, 2002),

$$P(Y = j|x) = \frac{\exp\{\eta_j(x)\}}{\sum_{t=1}^{M+1} \exp\{\eta_t(x)\}}, \qquad \eta_{M+1} \equiv 0, \qquad j = 1, \ldots, M+1.$$

We will now compare P-spline VGAM and VGAM fits to the marital status data from the `xs.nz` data frame in the VGAMdata package. These data were presented by Yee (2010) as an application of categorical data analysis. MacMahon et al. (1995) and Yee and Wild (1996) stated that the `xs.nz` data set can be reasonably used for representing the white male New Zealand population in the early 1990s. The specific outcome of interest is the nominal response $Y =$ marital status of European males, categorized according to divorced or separated (`Divorced/Separated`), married or partnered (`Married/Partnered`), single (`Single`) and widower (`Widowed`), where these levels are defined as $Y = 1, 2, 3, 4$ respectively. The married or partnered is chosen as the baseline category ($Y = 2$). The predictor available is `age`. We confine our analysis to a subset of 6053 European males. Missing values are removed. P-spline VGAMs are performed using the `psvgam()` function and VGLMs/VGAMs are performed using the VGAM package.

The objective of our analysis is to investigate how the relative chances of falling into the marital status categories depend upon the covariate `age`. We fit a nonparametric multinomial logit model to this data set. The model is formulated as (cf. Yee (2010))

$$\eta_j = \log\left(P(Y = t)/P(Y = 2)\right) = \beta_{(j)1} + f_{(j)2}(x_2). \tag{6.1}$$

Here, $j = 1, 2, 3$ indicates the jth additive predictor, $t = 1, 3, 4$ indicates the category of the response, and x_2 is `age`. For purposes of comparison, we will analyze this dataset using the proposed modeling framework as well as three models from the VGLM/VGAM approach. The four models and the corresponding calls are as follows:

(i) default VGAMs with 4 degrees of freedom for each smooth term ($\mathrm{df}_{(j)} = 4$) (default of `s()`)

```
fit.ms1 <- vgam(mstatus ~ s(age),
                family = multinomial(refLevel = 2),
                data = marital.nz)
```

(ii) VGAMs with 3 degrees of freedom for each smooth term ($\mathrm{df}_{(j)} = 3$) (Yee, 2010) [Note: this means 3 degrees of freedom allocated to nonlinearity (1 df = linear part).]

```r
fit.ms <- vgam(mstatus ~ s(age, df = 3),
               family = multinomial(refLevel = 2),
               data = marital.nz)
```

(iii) a parametric polynomial model (quadratic in `age` for `Divorced/Separated` group, piecewise quadratic in `age` for `Single` group, and linear in `age` for `Widowed` group) using VGLMs (Yee, 2010)

```r
foo <- function(x, elbow = 50) poly(pmin(x, elbow), 2)
clist <- list("(Intercept)" = diag(3),
              "poly(age, 2)" = rbind(1, 0, 0),
              "foo(age)" = rbind(0, 1, 0), age = rbind(0, 0, 1))
fit2.ms <- vglm(mstatus ~ poly(age, 2) + foo(age) + age,
                family = multinomial(refLevel = 2),
                constraints = clist, data = marital.nz)
```

(iv) P-spline VGAMs using penalized B-splines of degree 3, together with a second order penalty, and 10 equally spaced B-spline knots with the smoothing parameters being selected automatically through minimization of the UBRE score

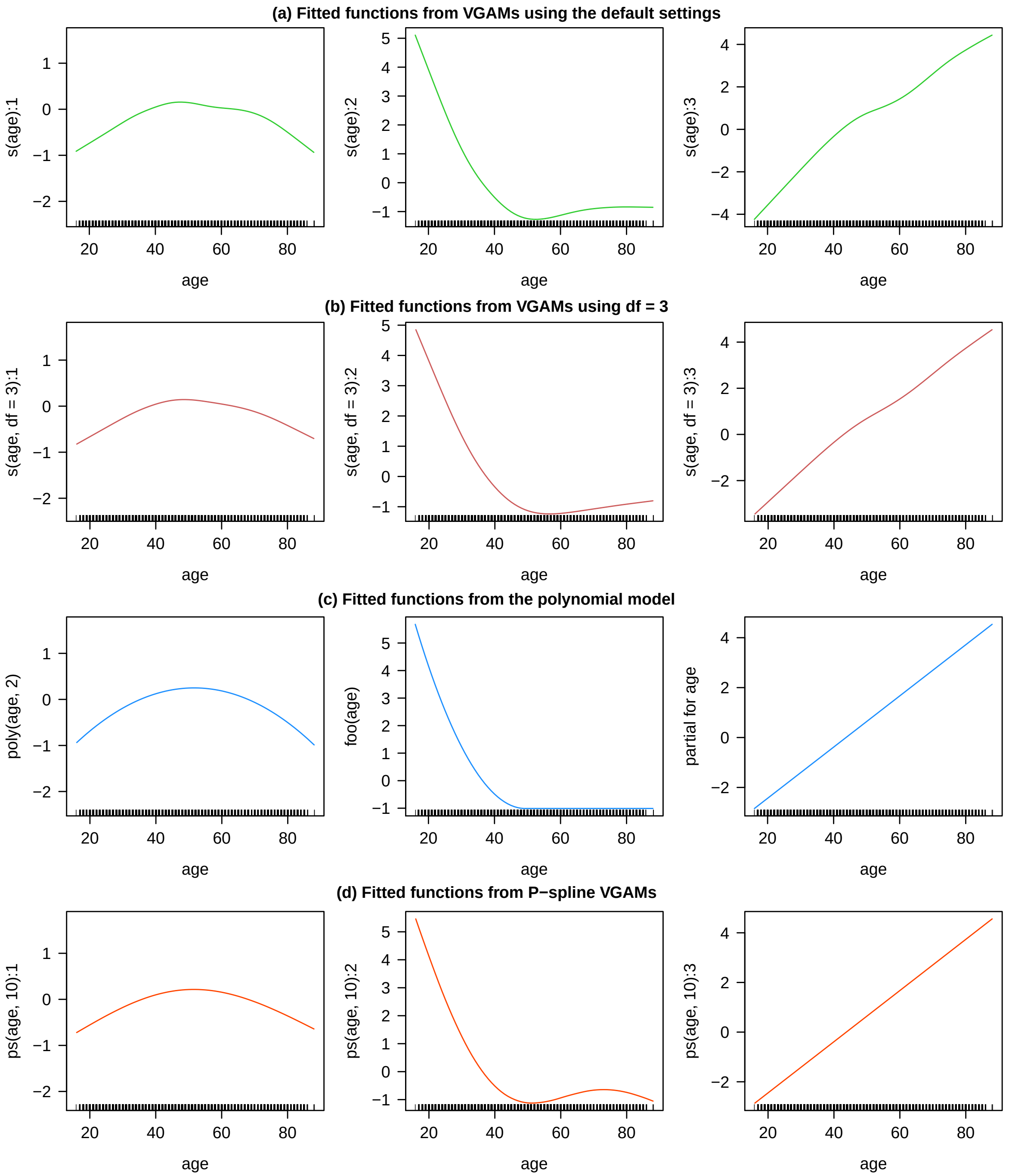
```r
fit.ps <- psvgam(mstatus ~ ps(age, ps.interval = 10),
                 family = multinomial(refLevel = 2),
                 data = marital.nz)
```

Table 6.1: EDF estimates for each smooth term obtained from (i) default VGAMs (`fit.ms1`), (ii) VGAMs ($\text{df}_{(j)} = 3$) (`fit.ms`) and (iii) P-spline VGAMs (`fit.ps`).

Model	$\widehat{f}_{(1)2}(x_2)$	$\widehat{f}_{(2)2}(x_2)$	$\widehat{f}_{(3)2}(x_2)$
(i) `s(age)`	3.7	3.6	3.6
(ii) `s(age,3)`	2.8	2.7	2.7
(iii) `ps(age,10)` ($\widehat{\lambda}_{(j)2}$)	1.9 (1.32)	3.4 (0.015)	1.0 (2.0×10^7)

The resulting curves for the four models are shown in Fig. 6.1. Table 6.1 shows the EDF estimates for each smooth term from the three models and the optimal smoothing parameters ($\widehat{\lambda}_{(j)2}$) obtained from P-spline VGAMs (`fit.ps`). The EDF estimates obtained from the P-spline VGAM approach suggest that $\widehat{f}_{(1)2}(x_2)$ and $\widehat{f}_{(2)2}(x_2)$ are nonlinear, while $\widehat{f}_{(3)2}(x_2)$ is linear.

We now interpret each plot obtained from the method proposed with interest. The $\widehat{f}_{(2)2}(x_2)$ in Fig. 6.1 ((d): middle panel) indicates that from the age of approximately 16 to about 40, the log relative risk of being single relative to being married/partnered goes down significantly and then is roughly horizontal between ages 45 and 80. The fitted function for the `Widowed` group increases in an approximately linear fashion as the age increases (Fig. 6.1 (d): right panel). One might surmise that marital conflict reaches its highest level at approximately 50 years of age (Fig. 6.1 (d): left panel).



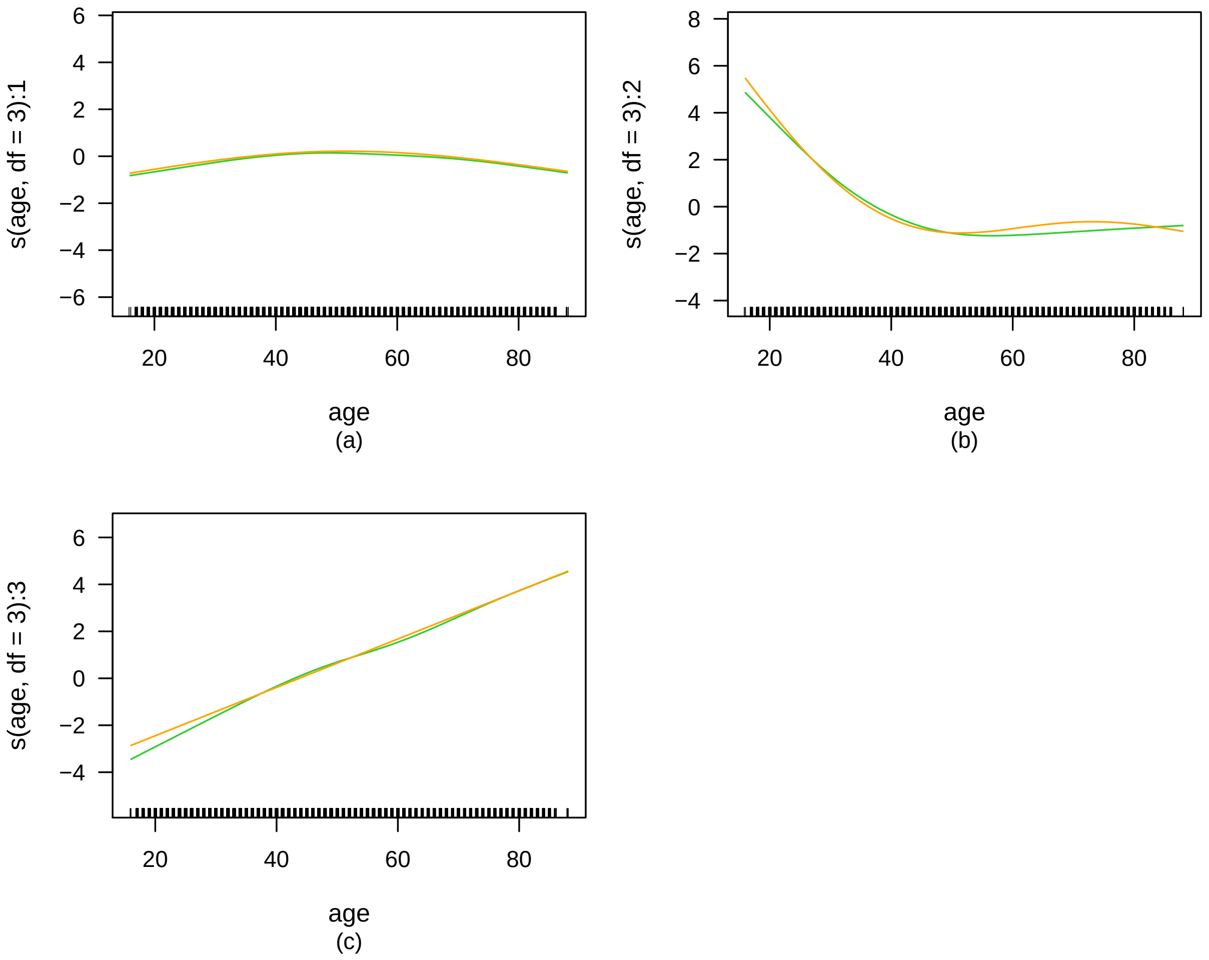
(a) Fitted functions from VGAMs using the default settings

(b) Fitted functions from VGAMs using df = 3

(c) Fitted functions from the polynomial model

(d) Fitted functions from P−spline VGAMs

Figure 6.1: Fitted functions $\widehat{f}_{(j)2}(x_2)$, $j = 1, 2, 3$ (see equation (6.1)) using (a) default VGAMs, (b) VGAMs ($\mathrm{df}_{(j)} = 3$), (c) VGLMs using polynomial terms and (d) P-spline VGAMs fitted to the NZ marital status data.



(a)

(b)

(c)

Figure 6.2: (a) – (c) Fitted functions $\widehat{f}_{(j)2}(x_2)$, $j = 1, 2, 3$. VGAMs ($\text{df}_{(j)} = 3$) (`fit.ms`) and P-spline VGAMs (`fit.ps`) are overlaid and respectively given by the green and orange lines.

Figs. 6.1 (a) – (d) show the estimated smooth functions of $f_{(j)2}(x_2)$ for η_j, $j = 1, 2, 3$ obtained using default VGAMs (`fit.ms1`), VGAMs ($\text{df}_{(j)} = 3$) (`fit.ms`), VGLMs (`fit2.ms`) and P-spline VGAMs (`fit.ps`) respectively. In particular, the differences are more pronounced for functions $f_{(1)2}(x_2)$ and $f_{(3)2}(x_2)$ (Fig. 6.1 (a): left and right panels) when employing default VGAMs. These plots indicate that the estimated curves for these functions are more wiggly than they should be since backfitting does not have any procedures to prevent complex smooth components when the data are not complex. Reducing a value for degrees of freedom of functions $f_{(1)2}(x_2)$ and $f_{(3)2}(x_2)$, as Yee (2010) did, improves the situation, as shown in Fig. 6.1 ((b): left and right panels). The P-spline VGAM approach (Fig. 6.1 (d)), on the other hand, performs

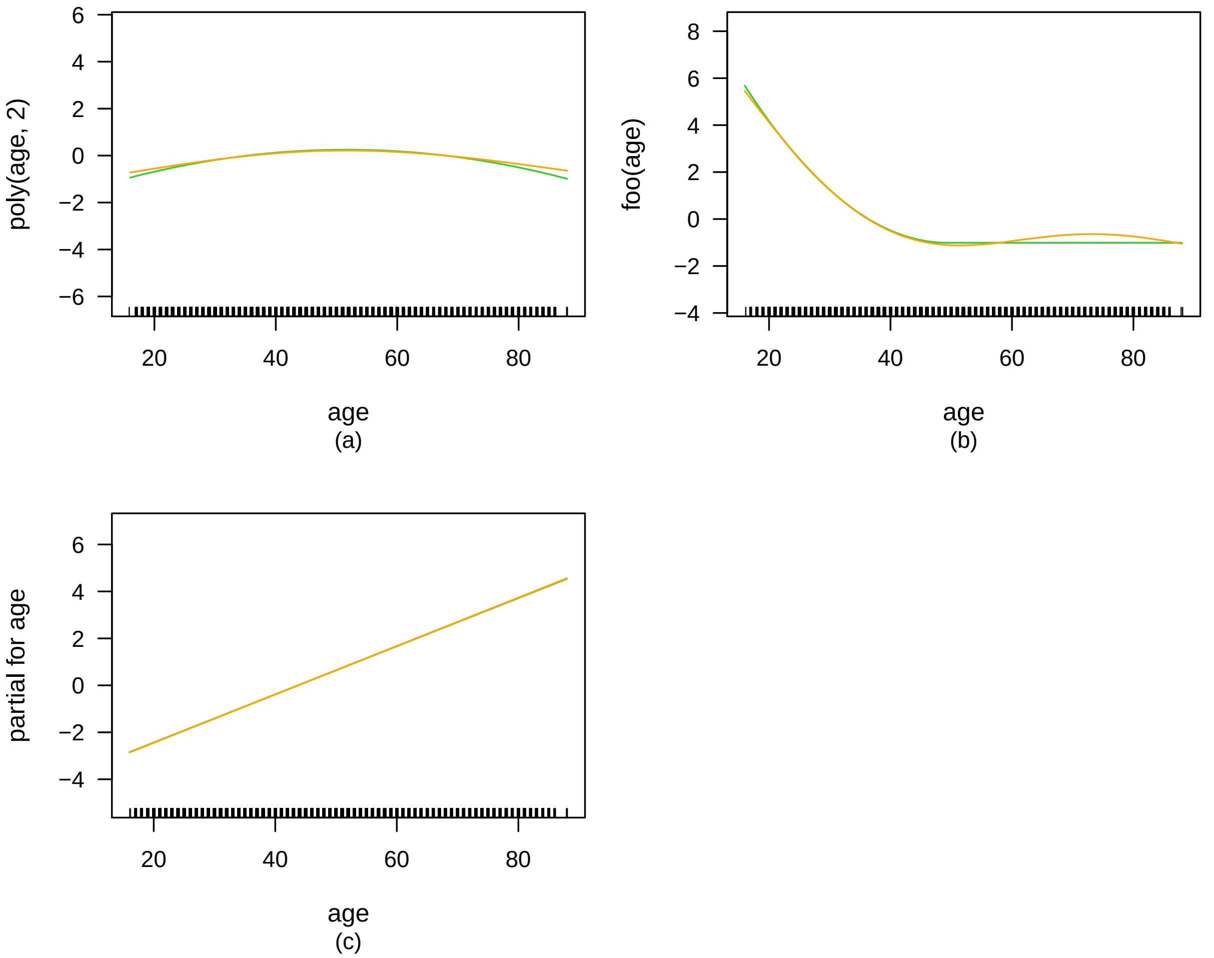


Figure 6.3: (a) – (c) Fitted functions $\widehat{f}_{(j)2}(x_2)$, $j = 1, 2, 3$. VGLMs (`fit2.ms`) and P-spline VGAMs (`fit.ps`) are overlaid and respectively given by the green and orange lines.

quite well compared to the default VGAMs (Fig. 6.1 (a)) since the amount of smoothing is estimated automatically. Overlaying plots of the fitted functions of $f_{(j)2}(x_2)$ for η_j, $j = 1, 2, 3$ in Fig. 6.2 indicate that the fitted curves obtained using the method proposed are very similar to the reasonable fitted models (`fit.ms`) given by Yee (2010). Yee (2010) suggested that functions $f_{(1)2}(x_2)$ and $f_{(2)2}(x_2)$ may be naturally quadratic, while function $f_{(3)2}(x_2)$ may be well fitted using a linear parametric component. He then simplified the nonparametric version of VGAMs ($\text{df}_{(j)} = 3$) (`fit.ms`) to a parametric polynomial model (quadratic in `Divorced/Separated`, piecewise quadratic in `Single`, and linear in `Widowed`) (`fit2.ms`), which the results are shown in Fig. 6.1 (c). Overlaying plots of the fitted functions of $f_{(j)2}(x_2)$ for η_j, $j = 1, 2, 3$ in Fig. 6.3

again indicate that the fitted curves obtained using the method proposed are very similar to the parametric polynomial model given by Yee (2010). This illustration depicts that P-spline VGAMs can be an effective approach to automatically find parametric components of VGAMs.

Fig. 6.4 shows the estimated smooth functions for $f_{(j)2}(x_2)$, $j = 1, 2, 3$ and their 2-standard-error bands obtained using the four models. As with GAMs, the 2-standard-error bands are useful as they give a rough indication of how much to trust the fitted functions. It may be seen that when there is almost no data in the region, the confidence limits obtained using backfitting, e.g., for the `Widowed` group and the `Single` group (Fig. 6.4 (b): left and right panels) are particularly wider than the estimated smooths obtained using the method proposed (Fig. 6.4 (d): left and right panels). This result depicts that the fitted curves obtained from the method proposed appear quite reasonable as compared with backfitting.

The deviance for the fitted VGAM model (`fit.ms`) considered as reasonable given by Yee (2010) is 6542.64 with $df^{\text{err}} = 18147.85$. Here, df^{err} is the residual "degrees of freedom" (cf. the residual degrees of freedom approximated for P-spline VGAMs in Section 4.6). Using P-spline VGAMs (`fit.ps`), the deviance drops to 6532.10 with $df^{\text{err}} = 18149.71$. The results indicate that the fit of the P-spline VGAM model, as measured by deviance, is better with a "smaller" model.

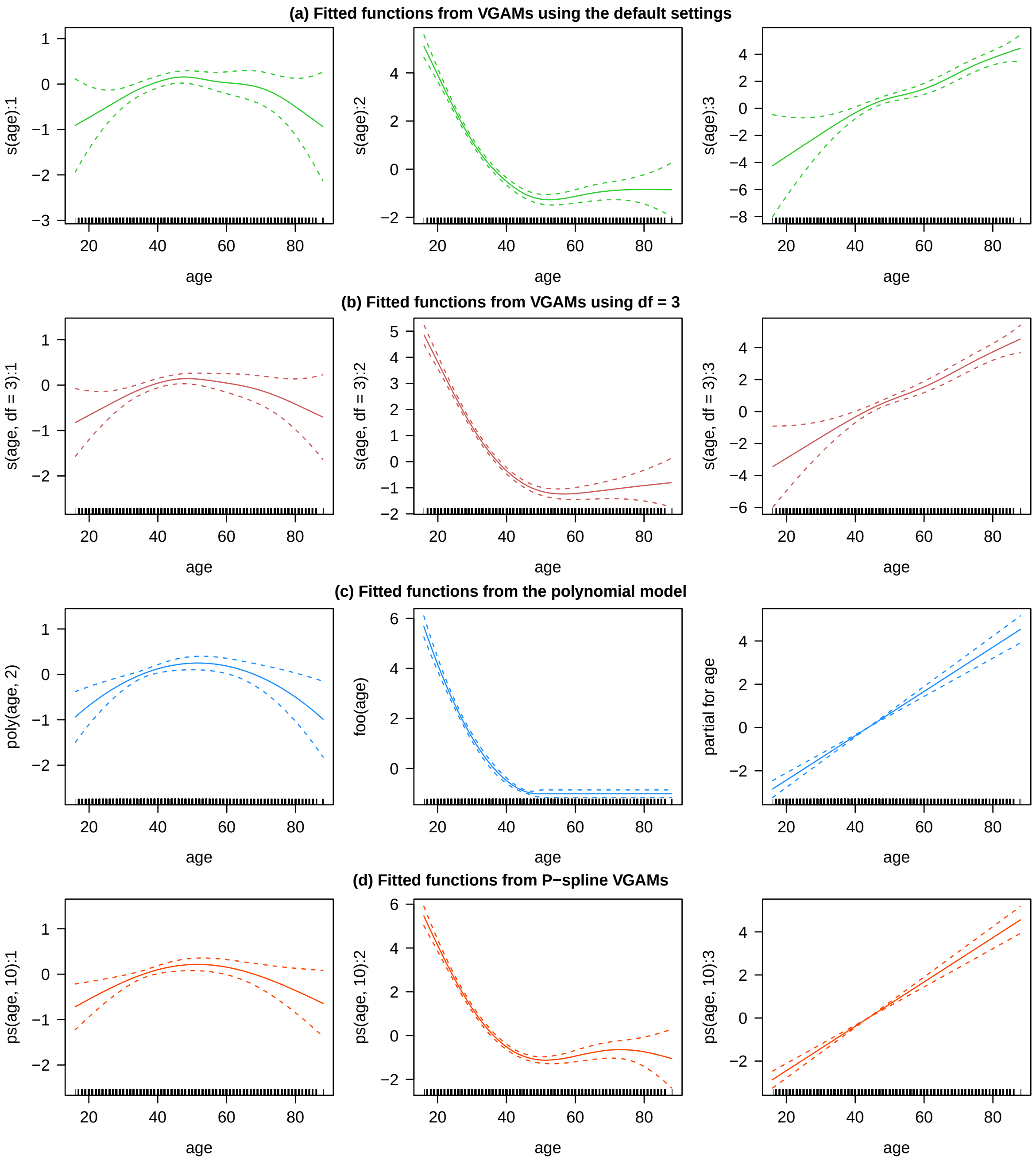


Figure 6.4: Fitted functions $\widehat{f}_{(j)2}(x_2)$, $j = 1, 2, 3$ (see equation (6.1)) and their 2-standard-error bands using (a) default VGAMs, (b) VGAMs ($\text{df}_{(j)} = 3$), (c) VGLMs using polynomial terms and (d) P-spline VGAMs fitted to the NZ marital status data.

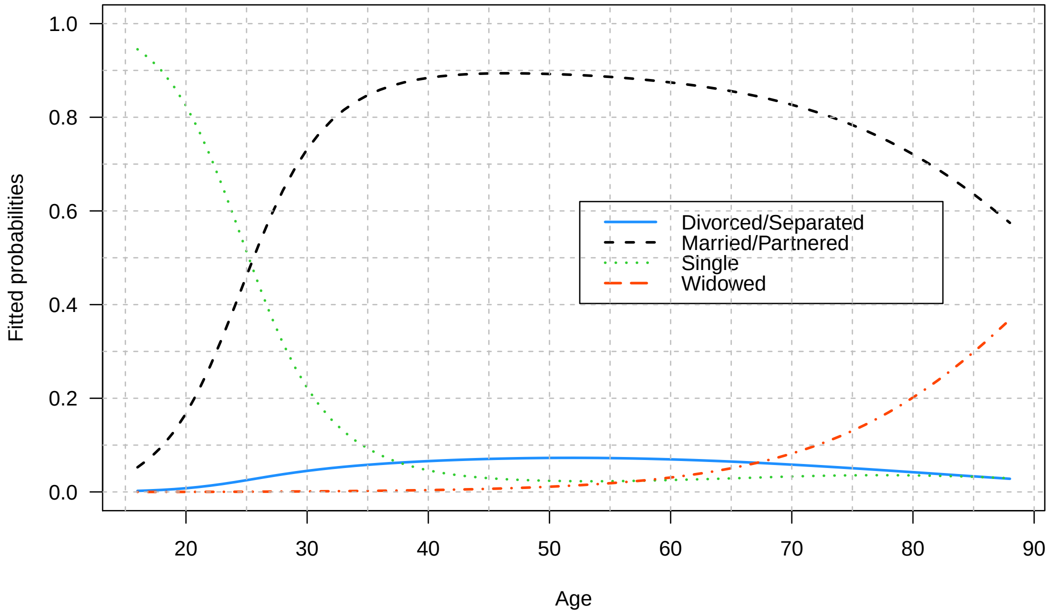


Figure 6.5: Estimated probabilities for each group of the NZ male European marital status using the P-spline VGAM approach.

Fig. 6.5 shows the four estimated response probabilities of the outcomes (divorced or separated, married or partnered, single and widower) for European males as a function of `age` from the nonparametric multinomial logit fit using the P-spline VGAM approach. From Fig. 6.5, the percentage of single starts to decrease sharply from 90% to 10% between the late teens and mid-30s, and then is roughly horizontal between 40 – 60. In contrast, the percentage of married/partnered increases steeply from 20% to over 80% between the early-20s and the mid-30s. Overall, approximately 80 – 90% of NZ males with age ranging from early-30s to early-70s were married/partnered. The proportion of divorced/separated hits the highest level of approximately 0.1 at the age of approximately 50. The percentage of widowed is almost zeros and then starts increasing to approximately 10% from the age over 70s.

6.3 Proportional-odds model fits to body mass index (BMI) data

When the responses are "ordinal" (consisting of ordered categories), the regression models can incorporate this ordering through the logit transformations of the response probabilities. The most frequently used models for an ordered response are the proportional and non-proportional odds models. We consider an ordered multinomial response variable y with categorical outcomes denoted by $1, 2, \ldots, K$, and $\boldsymbol{x}$ denote a vector of covariates. Yee and Wild (1996) described this model in terms of modeling each of the binary events $y \geq j$ versus $y < j$ using a logistic model:

$$\Pr(y \geq j|\boldsymbol{x}) \;=\; \frac{\exp\{\eta_j(\boldsymbol{x})\}}{1+\exp\{\eta_j(\boldsymbol{x})\}}, \qquad j \;=\; 2, \ldots, K. \tag{6.2}$$

The probabilities for observing a particular response category j are obtained by

$$\Pr(y \;=\; j) \;=\; \Pr(y \geq j) - \Pr(y \geq j+1).$$

The proportional-odds model of McCullagh (1980) assumes that the effect of the explanatory variables on the odds ratio is identical across the K-cut points. So that $\eta_j(\boldsymbol{x})$ in (6.2) is constrained to be in the form

$$\eta_j(\boldsymbol{x}) \;=\; \alpha_j + \eta(\boldsymbol{x}).$$

An unconstrained model that imposes no such assumptions is represented by

$$\eta_j(\boldsymbol{x}) \;=\; \alpha_j + \eta_j(\boldsymbol{x}), \tag{6.3}$$

where the effect of the explanatory variables on the odds ratio are now allowed to vary arbitrarily across the cutpoints of y. This equation is referred to the non-proportional odds model (Armstrong and Sloan, 1989). The additive version of the proportional-odds model can be obtained by using an additive predictor (Hastie and Tibshirani, 1990)

$$\eta_j(\boldsymbol{x}) \;=\; \sum_{k=1}^{p} f_k(x_k).$$

The body mass index (BMI) data from the `xs.nz` data frame in the VGAMdata package is used to illustrate our approach with ordinal responses. The response outcome of interest is BMI $y =$ BMI, transformed into three categories as follows: $< 18.5 - 24.9$: underweight or normal weight ($y = 1$); $25 - 29.9$: overweight ($y = 2$); ≥ 30: obese ($y = 3$). Note that we combine the underweight category with the normal weight category as the frequency counts in underweight category are very small. The considered explanatory variable is age ranging between $18 - 88$ years (`age`). For homogeneity, we confine our analysis to a subset of 2600 European women and missing values are removed. Again, P-spline VGAMs are performed using the `psvgam()` function and VGAMs are performed using the VGAM package.

We are interested in how $y =$ BMI-group varies as a function of age. Since the response variable is ordinal, we will use the proportional-odds model. The model is then

$$\operatorname{logit} \Pr\left(y \geq j | \text{age}\right) = f_{(j)}\left(\text{age}\right), \quad j = 2, 3.$$

We fit the nonparametric proportional-odds model using P-spline VGAMs. We smooth with B-splines of degree 3, a second order penalty and 10 equally-spaced B-spline knots with smoothing parameters being chosen automatically by minimizing the UBRE score. The resulting call to `psvagm()` is given by

```
fitps.pom <- psvgam(ordBMI ~ ps(age, 10), family = propodds,
                          data = women.bmi)
```

Since the proportional-odds model imposes the parallelism assumption, one should first investigate whether the proportional-odds assumption is reasonable. It turns out that the deviance for the nonparametric proportional-odds model is 4716.40, while that for the nonparametric non-proportional odds model is 4716.47. A comparison yields 0.07 on 2.54 degrees of freedom suggesting that the proportional-odds assumption is not unreasonable.

Table 6.2: EDF estimates for each smooth term obtained from (i) default VGAMs (`fit.pom`) and (ii) P-spline VGAMs (`fitps.pom`).

Model	$\widehat{f}$
(i) `s(age)`	4
(ii) `ps(age,10)` ($\widehat{\lambda}$)	2.6 (1.68)

To find out whether the VGAM approach leads to different results, we fit a nonparametric proportional-odds model using VGAMs as well. We smooth using cubic smoothing splines with default degrees of freedom ($\text{df}_{(j)} = 4$). The resulting call to `vgam()` is given by

```
fit.pom <- vgam(ordBMI ~ s(age), family = propodds, data = women.bmi)
```

Figs. 6.6 (a) – (b) show the fitted functions $\widehat{f}$ that P-spline VGAMs (`fitps.pom`) and default VGAMs (`fit.pom`) respectively yield. Table 6.2 shows the EDF estimates for $\widehat{f}$ from the two models and the optimal smoothing parameters $\widehat{\lambda}$ obtained from P-spline VGAMs (`fitps.pom`). The estimated function $\widehat{f}$ shown in Fig. 6.6 (a) measures overweight or obesity for European women and its EDF suggests the presence of non-linearity. It may be seen that the fitted function $\widehat{f}$ is generally increasing with age until about 60, and reaches the maximum at the age of mid-60s, and then starts to decrease.

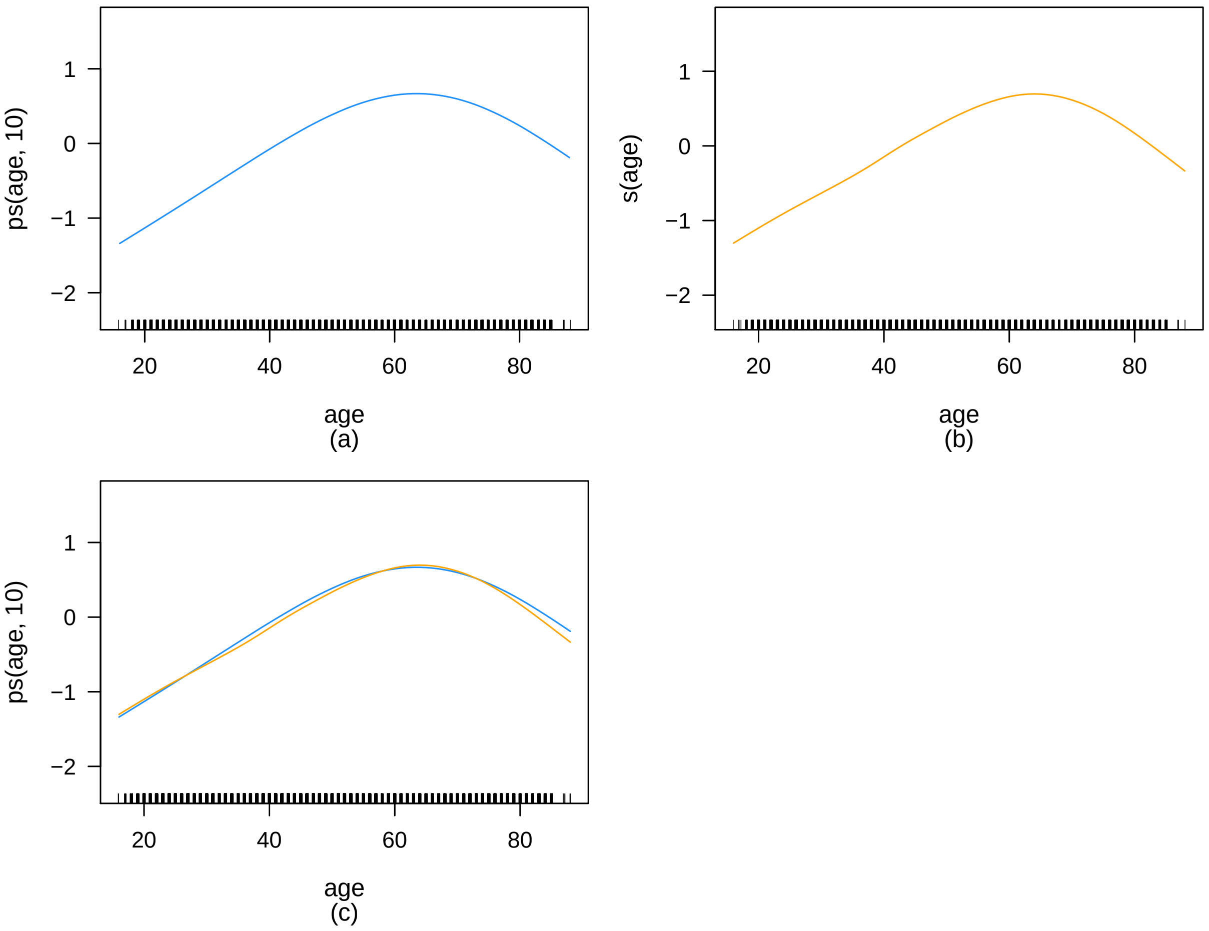


Figure 6.6: Estimate functions $\widehat{f}$ using P-spline VGAMs (a) and default VGAMs (b) fitted to the body mass index data. The fitted functions are overlaid in (c) with P-spline VGAMs (`fitps.pom`) and default VGAMs (`fit.pom`) given respectively by the blue and orange lines.

Fig. 6.6 (c) shows that the estimated curves from the default VGAM method do not differ much from the estimated smooths obtained using the method proposed. Fig. 6.7 shows that the confidence limits obtained using P-spline VGAMs are generally very similar to those of default VGAMs. This could suggest that the value of the degree of smoothness given by the default VGAM method is an appropriate choice of degree of smoothness for the given data.

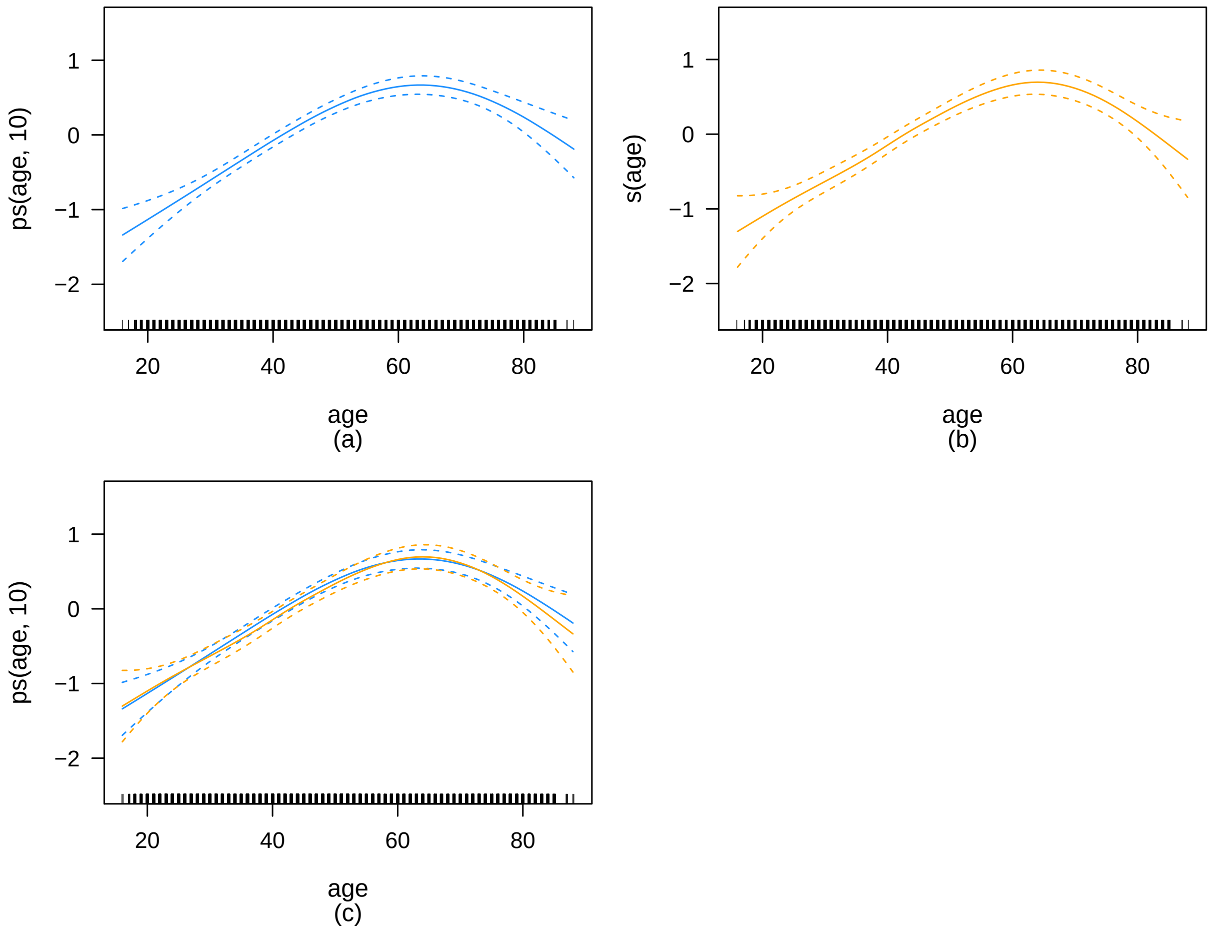


Figure 6.7: Estimate functions $\widehat{f}$ and their 2-standard-error bands using P-spline VGAMs (a) and default VGAMs (b) fitted to the body mass index data. The fitted functions and their 2-standard-error bands are overlaid in (c) with P-spline VGAMs (`fitps.pom`) and default VGAMs (`fit.pom`) are respectively given by the blue and orange lines.

Table 6.3: The deviance and the residual "degrees of freedom" of the two fits to the body mass index data.

Model	Dev	df^{err}	ΔDev	Δdf^{err}
(i) `ps(age,10)`	4716.47	5223.40		
(ii) `s(age)`	4713.53	5222.02	2.94	1.38

Although the models are not nested, we will use the informal deviance tests described in Section 4.6 as a rough guide for comparing models. Table 6.3 summarizes the results from the two models. For each model, we show the deviance of the fit (Dev) and the residual "degrees of freedom" (df^{err}). The change in deviance between each model fit (ΔDev) and the difference in the number of parameters between each model fit (Δdf^{err}) are also shown. The approximate chi-square test statistic under the null hypothesis that the nonparametric proportional-odds model obtained using P-spline VGAMs (`fitps.pom`) is suitable yields a p value of $\Pr\left(\chi^2_{1.38} > 2.94\right) = 0.14$, indicating that the simpler nonparametric proportional-odds model obtained using the method proposed appears quite reasonable compared to the default VGAM.

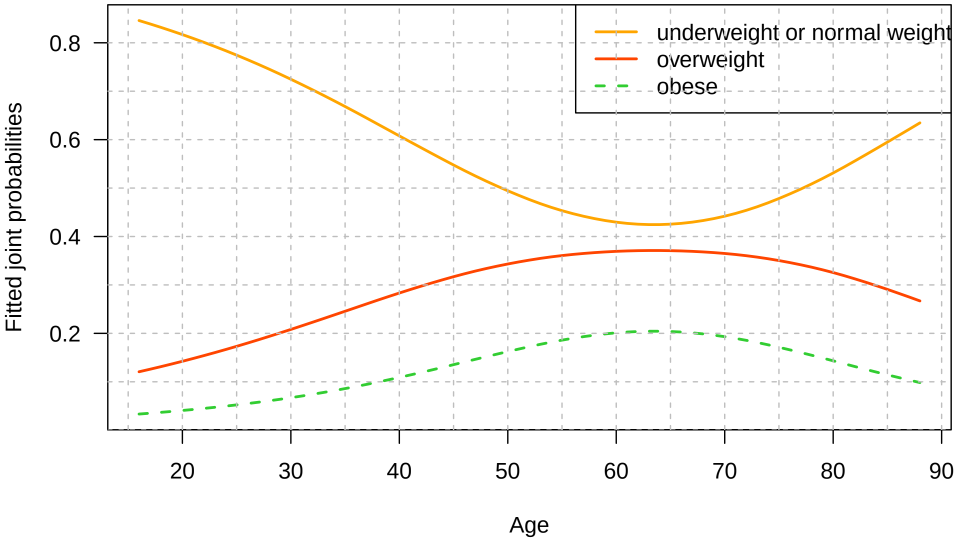


Figure 6.8: Estimated probabilities for each BMI-group of a subset of European women using the P-spline VGAM approach.

The fitted probabilities for individual categories of BMI obtained using the P-spline VGAM approach are shown in Fig. 6.8. The probability trends for being overweight and obese for European women are the same but the probability of being overweight is almost twice as large as that of being obese of the same age. The plot shows that the probability of being overweight and obese for European women increases with age, and reaches a maximum level at the age of mid-60s, and then starts to decrease. Sometimes data like this decreases past a maximum point. This may be a selection bias because people with high BMIs are more susceptible to certain diseases such as cardiovascular disease, diabetes and cancer, and therefore die younger.

6.4 Bivariate logistic fits to cat and dog data

We will compare P-spline VGAM and VGAM fits using the example of cat and dog pet ownership from the `xs.nz` data. These data were presented by Yee (2015b, section 4.4.3) as an application of VGAM fitting. The response outcome of interest is a European woman $Y_j,\ j = 1, 2,$ classified as having a household cat $(Y_1 = 1)$ and having a household dog $(Y_2 = 1)$. A single covariate considered, namely age (`age`). For homogeneity, we confine our analysis to a subset of 2569 European women and remove missing values. P-spline VGAMs are performed using the `psvgam()` function and VGAMs are performed using the VGAM package.

We will investigate how having a household cat and a household dog, together with their interactions, and how this is related to people's ages. We then fit a nonparametric bivariate logistic model to the two binary responses Y_j. The model is as follows (cf. Yee (2015b)):

$$\begin{aligned}
\eta_1 &= \operatorname{logit} P\,(Y_1 = 1|x_2) = \beta_{(1)1} + f_{(1)2}(x_2),\\
\eta_2 &= \operatorname{logit} P\,(Y_2 = 1|x_2) = \beta_{(2)1} + f_{(2)2}(x_2),\\
\eta_3 &= \log\,\psi(x_2) = \beta_{(3)1} + f_{(3)2}(x_2),
\end{aligned} \tag{6.4}$$

where $\psi(x_2) = \dfrac{\text{odds}\,(Y_1 = 1|Y_2 = 1, x_2)}{\text{odds}\,(Y_1 = 1|Y_2 = 0, x_2)}$ is the odds ratio. The model (6.4) taken here is a "non-exchangeable" error structure which the marginal probabilities are different and the odds ratio is modeled as a function of all the explanatory variables. We will fit the model above using the the P-spline VGAM approach and the VGLM/VGAM approach. The models and the corresponding calls are as follows:

(i) default VGAMs with 4 degrees of freedom for each smooth term

```
fit.cd0 <- vgam(cbind(cat, dog) ~ s(age),
                family = binom2.or(zero = NULL),
                data = women.eth0.catdog)
```

(ii) VGAMs with 4, 4, and 2 degrees of freedom for $f_{(j)2}$, $j = 1, 2, 3$ (Yee, 2015b)

```
fit.cd1 <- vgam(cbind(cat, dog) ~ s(age, df = c(4, 4, 2)),
                family = binom2.or(zero = NULL),
                data = women.eth0.catdog)
```

(iii) a parametric model (linear B-splines with 40 and 50 knots in `age` for European women who have a household cat (Y_1) and a household dog (Y_2) respectively, and quadratic in `age` for the odds ratio) using VGLMs (Yee, 2015b)

```
Hlist <- list("(Intercept)" = diag(3),
              "bs(age, degree = 1, knot = 40)" = rbind(1, 0, 0),
              "bs(age, degree = 1, knot = 50)" = rbind(0, 1, 0),
              "poly(age, 2)" = rbind(0, 0, 1))
fit.cd3 <- vglm(cbind(cat, dog) ~ bs(age, degree = 1, knot = 40) +
                bs(age, degree = 1, knot = 50) + poly(age, 2),
                family = binom2.or(zero = NULL),
                data = women.eth0.catdog, constraints = Hlist)
```

[Note: `vglm()` allocates the first `age` term to η_1, the second to η_2 and the third to η_3.]

(iv) P-spline VGAMs using penalized B-splines of degree 3, together with a second order penalty, and 10 equally-spaced B-spline knots with the smoothing parameters being selected automatically through minimization of the UBRE score

```
fitps.cd1 <- psvgam(cbind(cat, dog) ~ ps(age, ps.interval = 10),
                    family = binom2.or(zero = NULL),
                    data = women.eth0.catdog)
```

Table 6.4: EDF estimates for each smooth term obtained from (i) default VGAMs (`fit.cd0`), (ii) VGAMs ($\mathrm{df}_{(j)} = (4, 4, 2)$) (`fit.cd1`) and (iii) P-spline VGAMs (`fitps.cd1`).

Model	$\widehat{f}_{(1)2}(x_2)$	$\widehat{f}_{(2)2}(x_2)$	$\widehat{f}_{(3)2}(x_2)$
(i) `s(age)`	4.0	3.8	3.8
(ii) `s(age,df=c(4,4,2))`	4.0	3.8	1.9
(iii) `ps(age,10)` ($\widehat{\lambda}_{(j)2}$)	2.51 (2.08)	2.63 (0.94)	1.97 (3.02)

Fig. 6.9 shows the three fitted functions $\widehat{f}_{(j)2}(x_2)$, $j = 1, 2, 3$ that each of the four models yield. Table 6.4 shows the EDF estimates for each smooth term from the three models and the optimal smoothing parameters ($\widehat{\lambda}_{(j)2}$) obtained from P-spline VGAMs (`fit.ps`). The EDF estimates obtained using P-spline VGAMs suggest that $\widehat{f}_{(1)2}(x_2)$, $\widehat{f}_{(2)}(x_2)$ and $\widehat{f}_{(3)}(x_2)$ are nonlinear.

We now interpret each plot obtained from the method proposed with interest. Plots in Fig. 6.9 ((d): left and middle panels) correspond to European women who have a household cat and a household dog respectively. Both plots indicate that the probability of having a cat and a dog increases with age, reaches a maximum of the age of about 40 (cat) and 45 (dog), and then starts to decrease. To interpret the plot in Fig. 6.9 ((d): right panel), we will use the definition of the term ψ in equation (6.4) interpreted as the odds ratio of $Y_1 = 1$ for a European woman with a covariate $(Y_2 = 1,\ x_2)$ relative to a European woman with a covariate $(Y_2 = 0,\ x_2)$. We therefore describe the plot in Fig. 6.9 ((d): right panel) as follows. Over the age about 50, European women who have a cat are increasingly more likely to have a dog than those who do not have a cat.

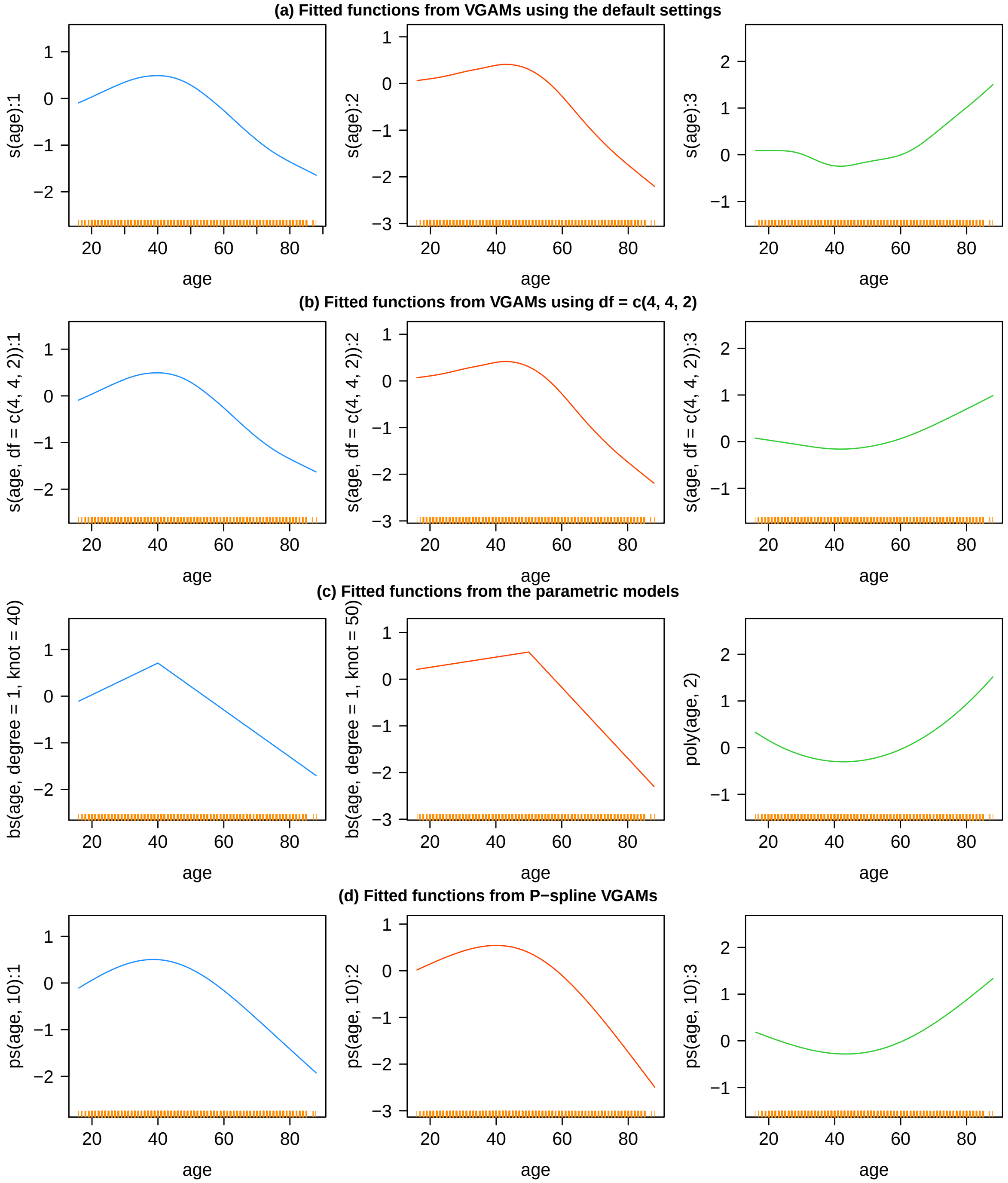


Figure 6.9: Fitted functions $\widehat{f}_{(j)2}(x_2)$, $j = 1, 2, 3$ (see equation (6.4)) using (a) default VGAMs, (b) VGAMs ($\text{df}_{(j)} = (4, 4, 2)$), (c) VGLMs using polynomial terms and (d) P-spline VGAMs fitted to a subset of European women with household cat and dog pet ownership data.

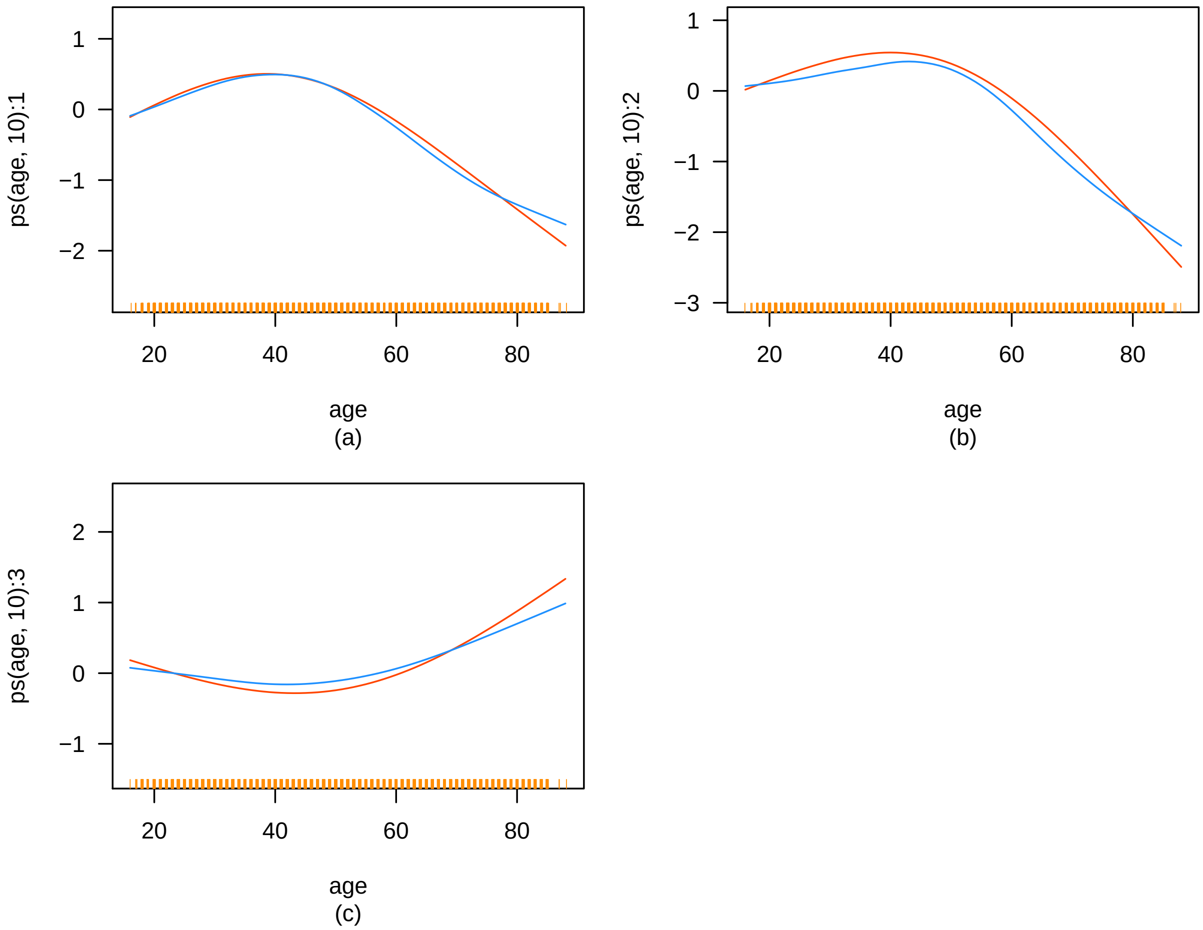


Figure 6.10: (a) – (c) Fitted functions $\widehat{f}_{(j)2}(x_2)$, $j = 1, 2, 3$. VGAMs ($\text{df}_{(j)} = (4, 4, 2)$) (`fit.cd1`) and P-spline VGAMs (`fitps.cd1`) are overlaid and respectively given by the blue and red lines.

Figs. 6.9 (a) – (d) show the estimated smooth functions $f_{(j)2}(x_2)$ for η_j, $j = 1, 2, 3$ obtained respectively using default VGAMs (`fit.cd0`), hand-tuned VGAMs ($\text{df}_{(j)} = (4, 4, 2)$) (`fit.cd1`) and VGLMs (`fit.cd3`), and P-spline VGAMs (`fitps.cd1`). Again, Fig. 6.9 ((a): right panel) shows that a value of the degrees of freedom given by the default VGAM method results in excess wiggliness in the estimated smooth function $\widehat{f}_{(3)2}(x_2)$. Yee (2015b) tuned this spline parameter to obtain a better fit (VGAMs ($\text{df}_{(j)} = (4, 4, 2)$) (`fit.cd1`)) as shown Fig. 6.9 (b). He then simplified the nonparametric version of VGAMs (`fit.cd1`) by substituting smooth terms with parametric terms (VGLMs (`fit.cd3`)) resulting in Fig. 6.9 (c). P-spline VGAMs (Fig. 6.9 (d)), by contrast, automatically yield a closer fit to the hand-tuned fit than the default VGAM method.

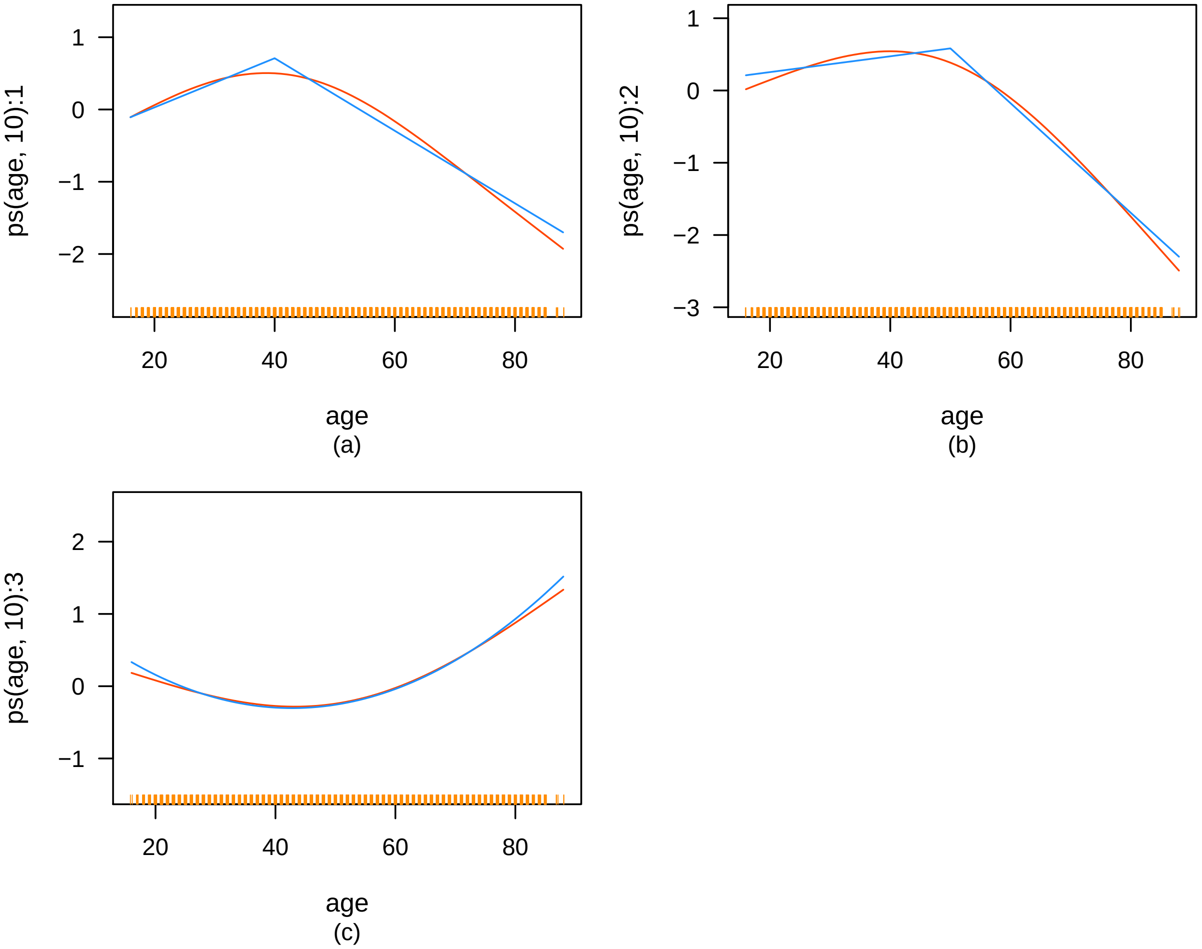
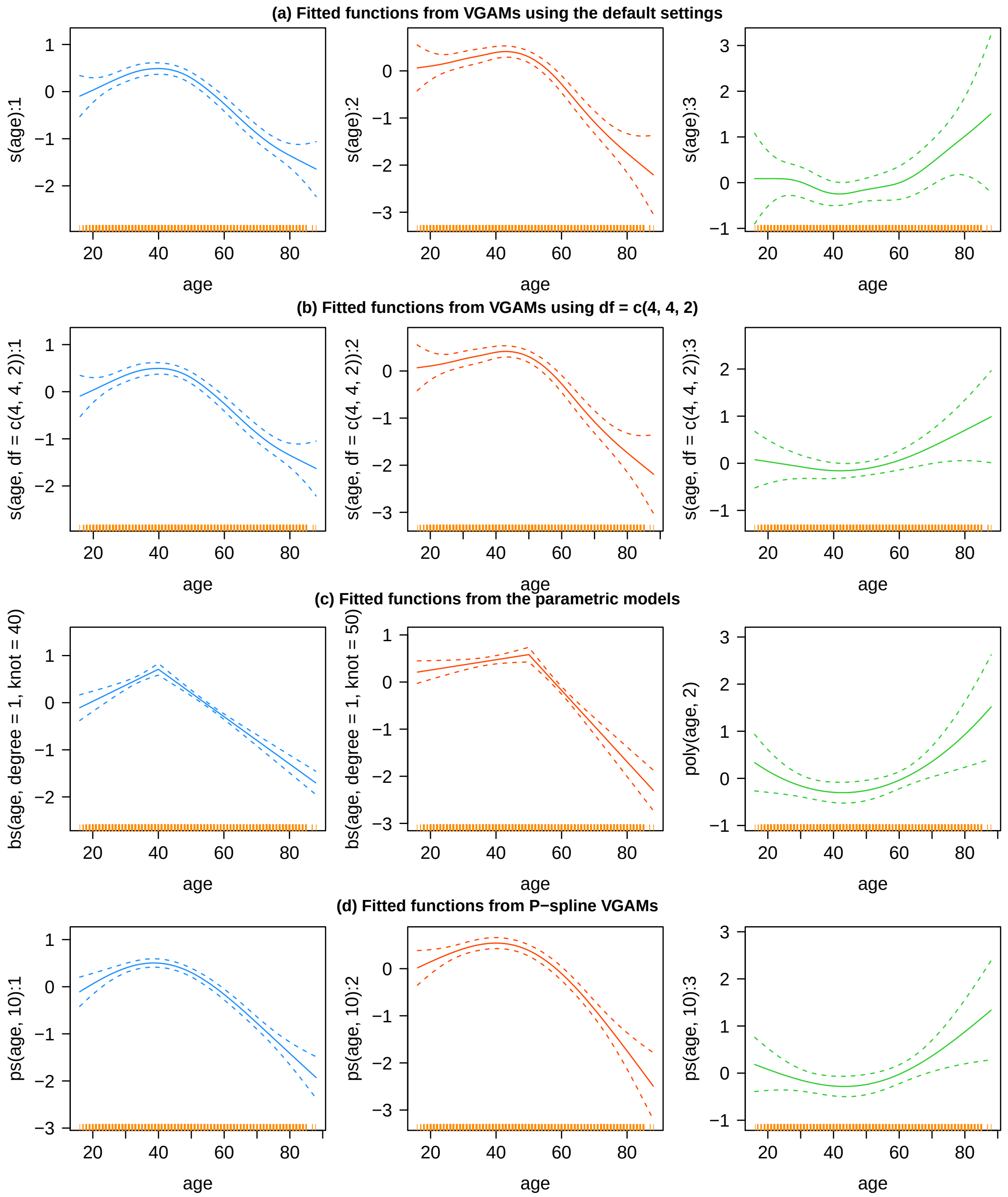


Figure 6.11: (a) – (c) Fitted functions $\widehat{f}_{(j)2}(x_2)$, $j = 1, 2, 3$. VGLMs (`fit.cd3`) and P-spline VGAMs (`fitps.cd1`) are overlaid and respectively given by the blue and red lines.

Overlaying plots of the fitted functions for $f_{(j)2}(x_2)$, $j = 1, 2, 3$ show that the fitted curves obtained using the method proposed are quite similar to the reasonable fitted models (`fit.cd1`) (Fig. 6.10) and very similar to the hand-tuned parametric polynomial version (`fit.cd3`) (Fig. 6.11) given by Yee (2015b) as might be expected. These results again suggest that P-VGAMs can be effectively used to automatically find parametric components of VGAMs.

Fig. 6.12 shows the fitted functions for $f_{(j)2}(x_2)$, $j = 1, 2, 3$ and their 2-standard-error bands obtained using the four models. With the exception of the model from the parametric models with their shape "elbows", the confidence intervals of the models are generally very similar (Figs. 6.12 (b) – (d)).



(a) Fitted functions from VGAMs using the default settings

(b) Fitted functions from VGAMs using df = c(4, 4, 2)

(c) Fitted functions from the parametric models

(d) Fitted functions from P−spline VGAMs

Figure 6.12: Fitted functions $\widehat{f}_{(j)2}(x_2)$, $j = 1, 2, 3$ (see equation (6.4)) and their 2-standard-error bands using (a) default VGAMs, (b) VGAMs ($\text{df}_{(j)} = (4, 4, 2)$), (c) VGLMs using polynomial terms and (d) P-spline VGAMs fitted to a subset of European women with household cat and dog pet ownership data.

Table 6.5: The deviance and the residual "degrees of freedom" of the two fits to a subset of European women with household cat and dog pet ownership data.

Model	Dev	df^{err}	ΔDev	Δdf^{err}
(i) `ps(age,10)`	6167.48	7696.89		
(ii) `s(age,c(4,4,2))`	6160.23	7694.35	7.25	2.54

Table 6.5 summarizes the deviance and the residual "degrees of freedom" of the models obtained using P-spline VGAMs (`fitps.cd1`) and VGAMs ($\mathrm{df}_{(j)} = (4, 4, 2)$) (`fit.cd1`). If we use the approximate chi-square test as a rough guide (these models are not nested), we get a p value of $\Pr\left(\chi^2_{2.54} > 7.25\right) = 0.05$, which might suggest a degree of underfitting. If we increase the number of knots to 15 (following the increased number of knots in the Yee's (2015b) parametric model (`fit.cd3`)), then the fits become almost indistinguishable.

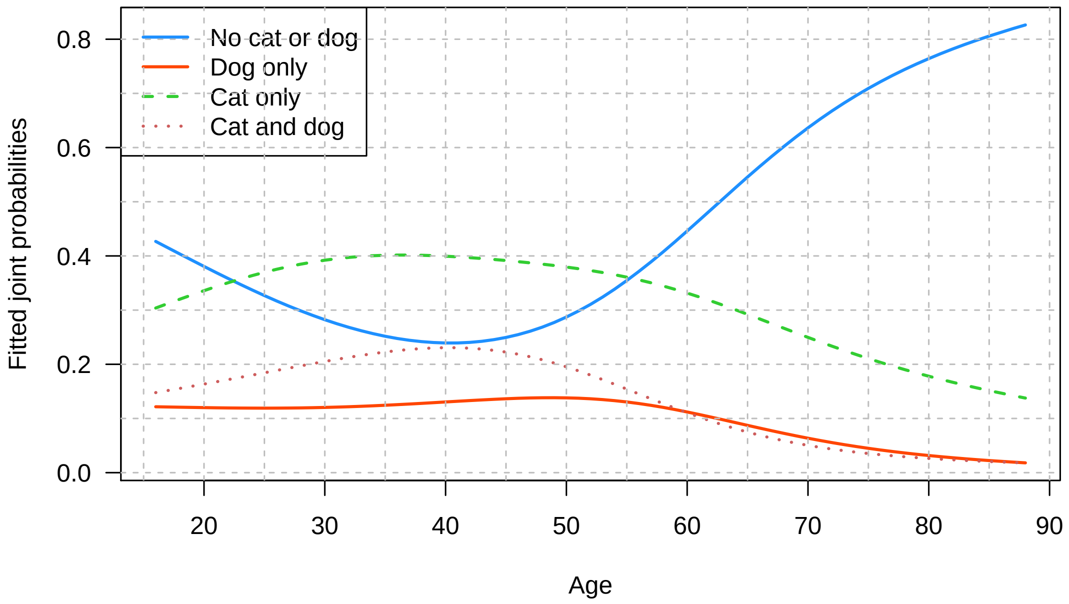


Figure 6.13: Estimated probabilities for all four combinations (both cats and dogs, cats only, dogs only, and no cats or dogs) of a subset of European women from the `xs.nz` data frame using the P-spline VGAM approach.

Fig. 6.13 shows the fitted probabilities for all four combinations (both cats and dogs, cats only, dogs only, and no cats or dogs) obtained using the method proposed. The probability of having cats only at home for European women is higher than that of having dogs only for European women of the same age. The proportion of having cats alone generally increases with age, reaches a maximum of 0.4 at the age between the mid-30s and the mid-40s, and then starts to decrease. The probability trends for having both types of pets and having cats only for European women are very similar in shape but the the probability of having cats only is almost twice as large as that of having both types of pets of the same age. The proportion having a dog only stays fairly stable over the age between 20 to 60 and then starts to decrease. The proportion of having no pets drops from the ages of 15 to 40 and then starts to climb quite steeply.

6.5 Bivariate logistic fits to pregnancy and birth data

We will use the `birth` data in the `catdata` package to illustrate the proposed method in a dataset with two binary responses and more than one covariate. These data were presented by Boulesteix (2006) as an application of the method used for the selected chi-square statistics for ordinal variables. The data of 620 labours contain several variables related to the birth process. The responses of interest are whether instruments have been used during labour $(Y_1 = 1)$ for each labour and whether birth has been induced ($Y_2 = 1$). The predictors are age at labour (`Age`) and weight of the baby at birth (`Weight`, in grams). We have removed cases with missing values. P-spline VGAMs are fitted using the `psvgam()` function and VGAMs are fitted using the VGAM package.

We are interested in examining how the probability of requiring the instruments during labour and/or birth and induced labour varies as a function of labour's age (x_2) and baby's weight (x_3). The nonparametric bivariate logistic model for these two predictors is given by

$$\begin{aligned}
\eta_1 &= \operatorname{logit} P\,(Y_1 = 1|\boldsymbol{x}) = \beta_{(1)1} + f_{(1)2}(x_2) + f_{(1)3}(x_3),\\
\eta_2 &= \operatorname{logit} P\,(Y_2 = 1|\boldsymbol{x}) = \beta_{(2)1} + f_{(2)2}(x_2) + f_{(2)3}(x_3),\\
\eta_3 &= \log\,\psi(\boldsymbol{x}) = \beta_{(3)1} + f_{(3)2}(x_2) + f_{(3)3}(x_3),
\end{aligned} \tag{6.5}$$

where $\psi(\boldsymbol{x})$ is the odds ratio. We note that (6.5) has a "non-exchangeable" structure in which no constraints are imposed on η_1, η_2 and η_3. As in previous examples, we fit the nonparametric bivariate logistic model in (6.5) using the P-spline VGAMs approach and the VGAM method. The models and the corresponding calls are as follows :

(i) default VGAMs with the default 4 degrees of freedom for each smooth term

```r
fit.bi1 <- vgam(cbind(Instrument, Induced) ~ s(Age) + s(Weight),
                family = binom2.or(zero = NULL), data = birth.b)
```

(ii) P-spline VGAMs using penalized B-splines of degree 3, together with a second order penalty, and 8 equally-spaced B-spline knots with the smoothing parameters being selected automatically through minimization of the UBRE score

```r
fitps.bi1 <- psvgam(cbind(Instrument, Induced) ~ ps(Age, 8) +
                    ps(Weight, 8), family = binom2.or(zero = NULL),
                    data = birth.b)
```

Table 6.6: EDF estimates for each smooth term obtained from (i) default VGAMs (`fit.bi1`) and (ii) P-spline VGAMs (`fitps.bi1`).

Model	$\widehat{f}_{(1)2}(x_2)$	$\widehat{f}_{(1)3}(x_3)$	$\widehat{f}_{(2)2}(x_2)$	$\widehat{f}_{(2)3}(x_3)$	$\widehat{f}_{(3)2}(x_2)$	$\widehat{f}_{(3)3}(x_3)$
(i) `s(Age)+s(Weight)`	3.7	3.9	4.1	3.6	3.8	3.3
(ii) `ps(Age,8)+ps(Weight,8)`	2.5	2.6	2.2	2.6	2.0	1.0
$(\widehat{\lambda}_{(j)k})$	(0.53)	(0.46)	(3.52)	(0.22)	(0.74)	(1.7×10^7)

Figs. 6.14 and 6.15 show the fitted functions $\widehat{f}_{(j)k}(x_k)$, $j = 1, 2, 3$ and $k = 2, 3$ that P-spline VGAMs (`fitps.bi1`) and default VGAMs (`fit.bi1`) yield. Table 6.6 shows the EDF estimates for each smooth term from the two models and the optimal smoothing parameters $(\widehat{\lambda}_{(j)k})$ obtained from P-spline VGAMs (`fit.ps`). The EDF estimates obtained using P-spline VGAMs suggest that the $\widehat{f}_{(j)k}(x_k)$ are nonlinear with the exception of $\widehat{f}_{(3)3}(x_3)$. For $\widehat{f}_{(3)3}(x_3)$, the EDF estimate is approximately 1 suggesting linearity.

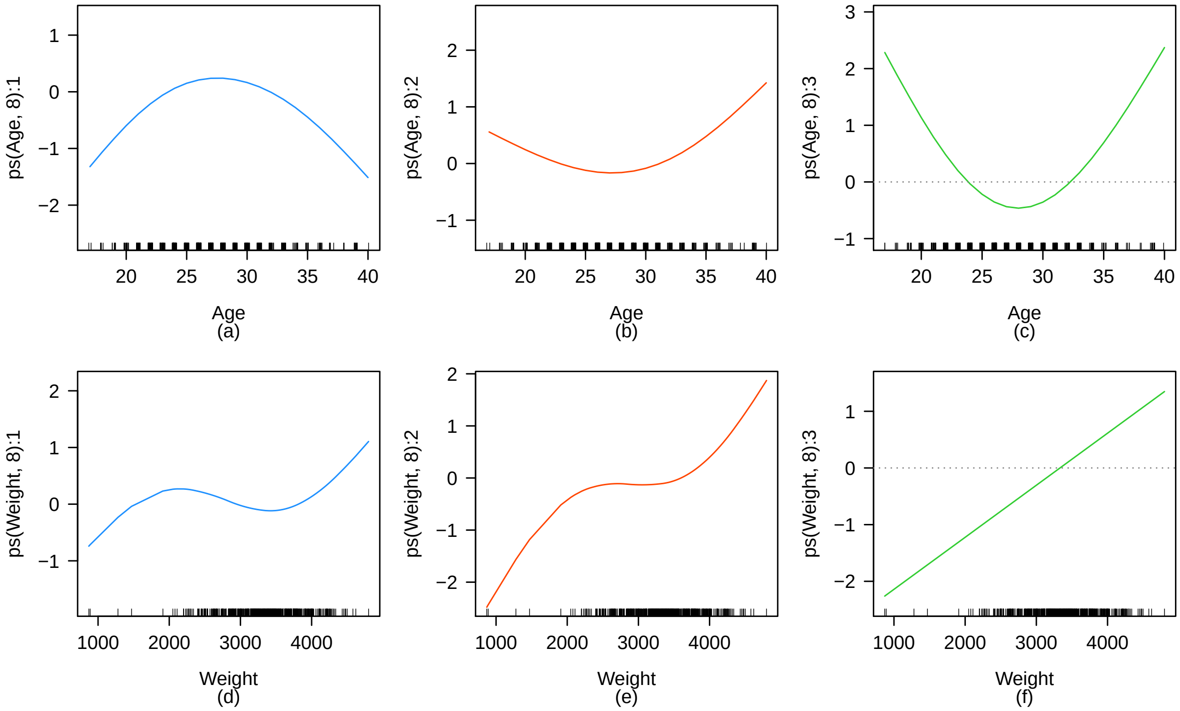


Figure 6.14: Fitted functions $\widehat{f}_{(1)2}(\text{age})$, $\widehat{f}_{(2)2}(\text{age})$, $\widehat{f}_{(3)2}(\text{age})$, $\widehat{f}_{(1)3}(\text{weight})$, $\widehat{f}_{(2)3}(\text{weight})$ and $\widehat{f}_{(3)3}(\text{weight})$ (see equation (6.5)) using P-spline VGAMs (`fitps.bi1`) fitted to the pregnancy and birth data.

We now interpret each plot. Each plot in Figs. 6.14 (a), (b), (d) and (e) can be described in the same way as an ordinary logistic regression for GAMs. Plots (a) and (d) correspond to the need of instruments during labour while (b) and (e) correspond to labour induction. The fitted curve $\widehat{f}_{(2)2}(x_2)$ (Fig. 6.14 (a)) suggests that a need of instruments during labour increases from the age of early-20s to mid-20s, reaches a maximum at the age of about 26, and then starts to decrease. Plot (b) indicates that the use of labour induction generally decreases with labour's age, reaches a minimum at an age of 28, and then starts to increase. Plots (d) and (e) indicate that a need for the instruments during labour and induced labour increases when the baby's weight is greater than approximately 3500 grams. The down trend of low weights corresponds to a region with very little data.

To interpret the plot in Figs. 6.14 (c) and (f), we recall from (6.4) that

$$\psi(\boldsymbol{x}) \;=\; \frac{\text{odds}\,(Y_1 \;=\; 1|\,Y_2 \;=\; 1,\,\boldsymbol{x})}{\text{odds}\,(Y_1 \;=\; 1|\,Y_2 \;=\; 0,\,\boldsymbol{x})}.$$

Here, ψ is the conditional odds ratio given $\boldsymbol{x}$ for the use of instruments $(Y_1 \;=\; 1)$ comparing induced births to non-induced births (Y_2). The fitted curve for the log-odds ratio versus age in Fig. 6.14 (c) suggests that the odds of instrument use is higher for induced births than non-induced births under the age of about 24 and over about 33, but lower for induced births than non-induced births for ages of between 28 and 33. The fitted curve for the log-odds ratio versus a baby's weight in Fig. 6.14 (f) suggests that when the baby's weight is above 3500 grams, the odds of requiring the use of instrument for those who required induced birth is higher than the odds of requiring the use of instrument for those who did not require induced birth. When the baby's weight is below 3500 grams, the odds of requiring the use of instruments is lower for induced labours than non-induced labours. The trend is linear on a log-odds-ratio scale.

Fig. 6.15 shows the estimated functions for $f_{(j)k}(x_k)$, $j \,=\, 1,2,3$, $k \,=\, 2,3$ obtained using VGAMs (`fit.bi1`). The differences are more pronounced for functions $f_{(1)2}(x_2)$, $f_{(2)2}(x_2)$, $f_{(3)2}(x_2)$, and $f_{(3)3}(x_3)$ (Figs. 6.15 (a), (b), (c), and (f)) when employing default VGAMs. These plots again indicate that a value of degree of smoothness given by the default VGAM method leads to an excessive wiggliness of the estimated smooth functions (overfitting). The P-spline VGAM approach appears to be correcting this automatically.

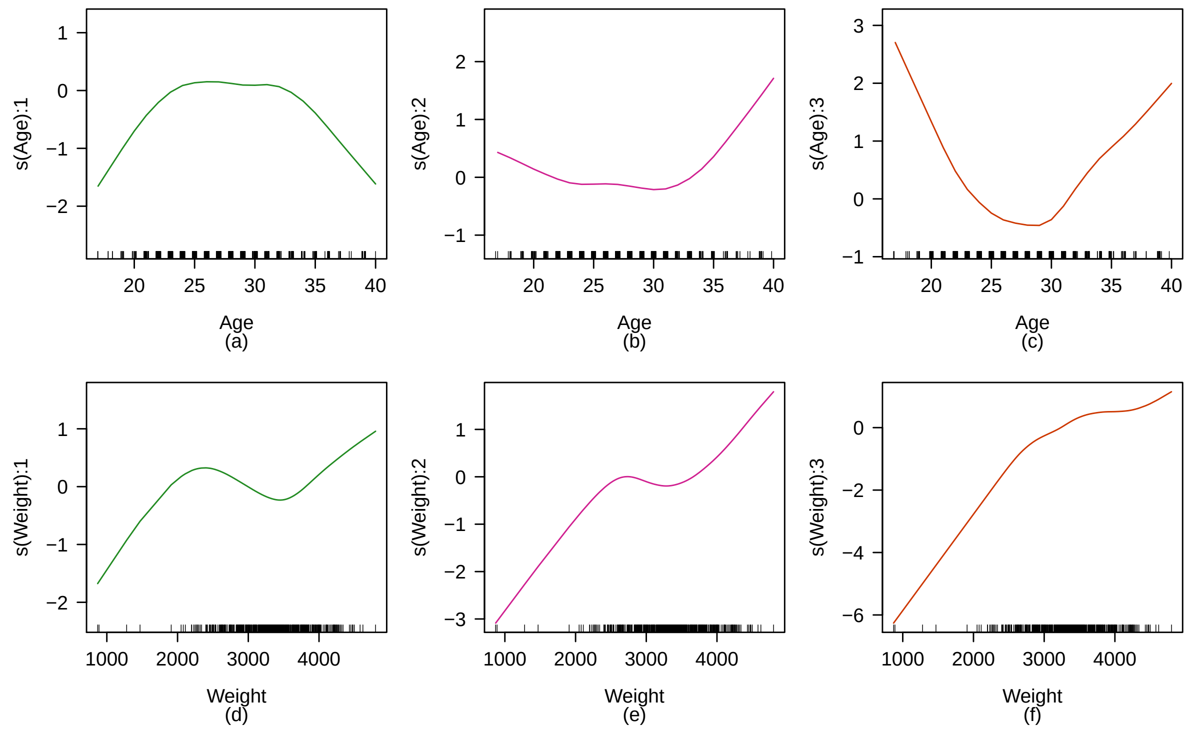


Figure 6.15: Fitted functions $\widehat{f}_{(1)2}(\text{age})$, $\widehat{f}_{(2)2}(\text{age})$, $\widehat{f}_{(3)2}(\text{age})$, $\widehat{f}_{(1)3}(\text{weight})$, $\widehat{f}_{(2)3}(\text{weight})$ and $\widehat{f}_{(3)3}(\text{weight})$ (see equation (6.5)) using default VGAMs (`fit.bi1`) fitted to the pregnancy and birth data.

Figs. 6.16 and 6.17 show the fitted curves and their 2-standard-error bands obtained respectively using P-spline VGAMs and default VGAMs. The standard error bands for default VGAMs, e.g., for $f_{(3)2}(x_2)$, $f_{(1)3}(x_3)$, $f_{(2)3}(x_3)$ and $f_{(3)3}(x_3)$ (Figs. 6.17 (c) – (f)) are generally wider than those of using the P-spline VGAM method (Figs. 6.16 (c) – (f)) as might be expected because the degrees of freedom obtained from P-spline VGAMs are lower.

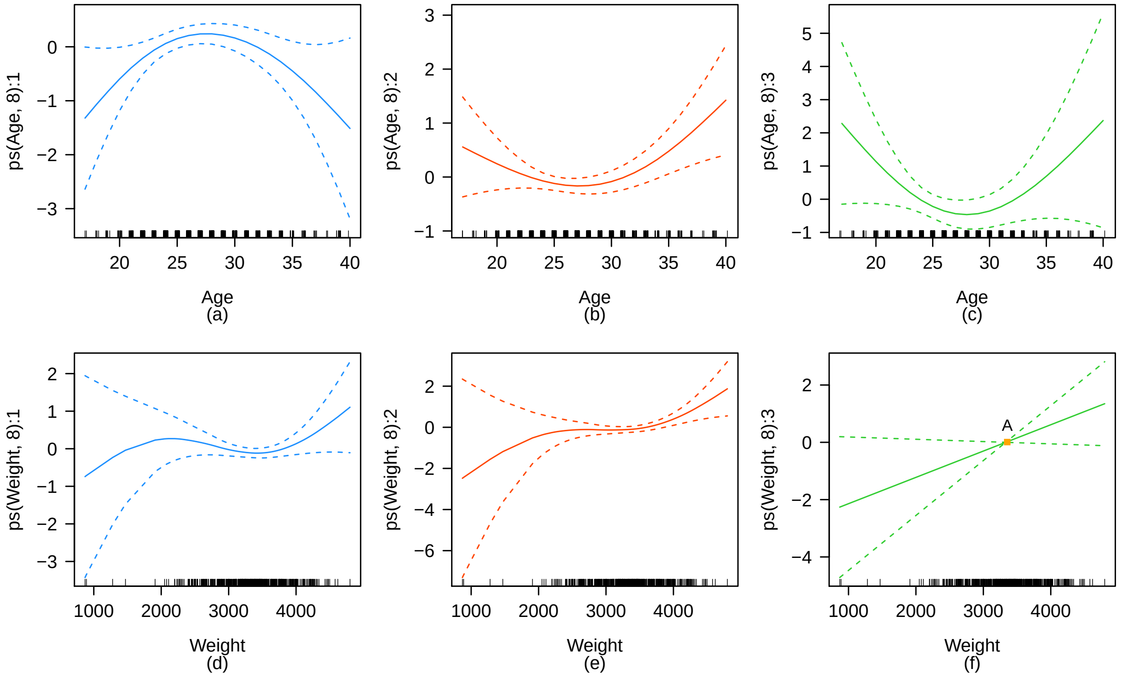


Figure 6.16: Fitted functions $\widehat{f}_{(1)2}(\text{age})$, $\widehat{f}_{(2)2}(\text{age})$, $\widehat{f}_{(3)2}(\text{age})$, $\widehat{f}_{(1)3}(\text{weight})$, $\widehat{f}_{(2)3}(\text{weight})$ and $\widehat{f}_{(3)3}(\text{weight})$ (see equation (6.5)) and their 2-standard-error bands using P-spline VGAMs (`fitps.bi1`).

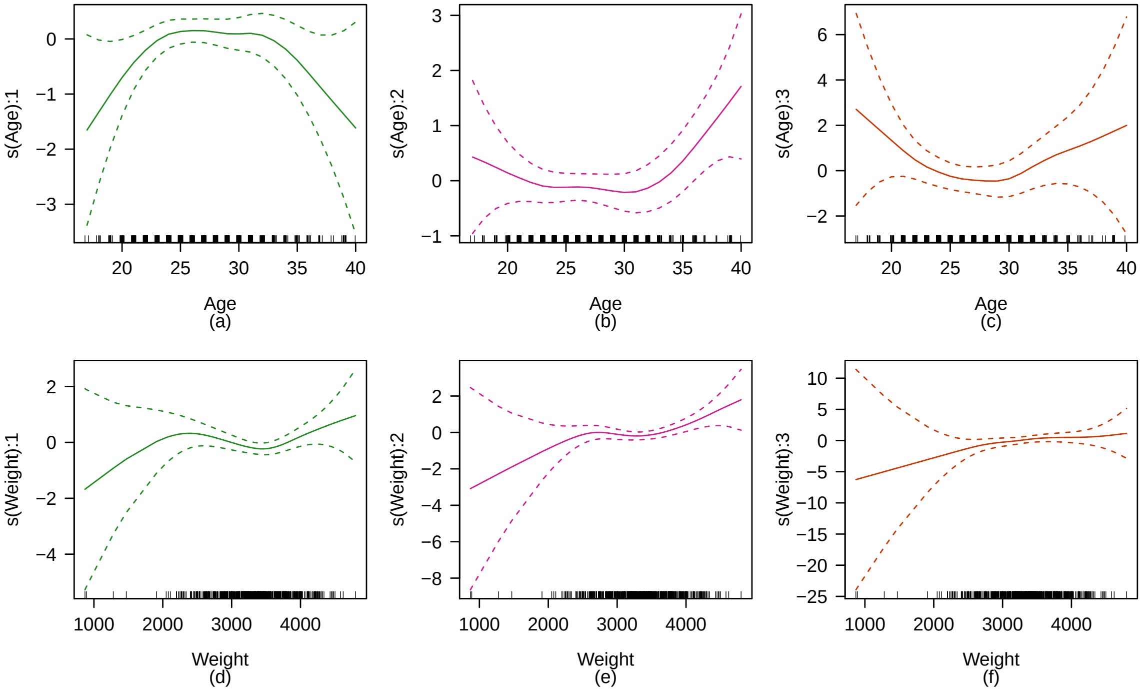


Figure 6.17: Fitted functions $\widehat{f}_{(1)2}(\text{age})$, $\widehat{f}_{(2)2}(\text{age})$, $\widehat{f}_{(3)2}(\text{age})$, $\widehat{f}_{(1)3}(\text{weight})$, $\widehat{f}_{(2)3}(\text{weight})$ and $\widehat{f}_{(3)3}(\text{weight})$ (see equation (6.5)) and their 2-standard-error bands using default VGAMs (`fit.bi1`).

Table 6.7: The deviance and the residual “degrees of freedom” of the two fits to the birth data.

Model	*Dev*	df^{err}	ΔDev	Δdf^{err}
(i) `ps(Age,8)+ps(Weight,8)`	1280.25	1844.07		
(ii) `s(Age)+s(Weight)`	1268.20	1834.58	12.05	9.49

The point labeled A in Fig. 6.16 (f) for η_3, which has value 0 and standard error zero, occurs at the mean value of the variable `Weight`, which is where the component function has been centered.

Table 6.7 summarizes the deviance and the residual “degrees of freedom” of the models obtained using P-spline VGAMs (`fitps.bi1`) and default VGAMs (`fit.bi1`). The approximate chi-square test statistic yields a p value of $\Pr\left(\chi^2_{9.49} > 12.05\right) = 0.24$, indicating that the nonparametric bivariate logistic model obtained using the method proposed appears quite reasonable as compared to the default VGAM.

6.6 The LMS method fits to body mass index (BMI) data

In this section, we will compare P-spline VGAM and VGAM fits to the BMI data from `xs.nz` in the VGAMdata package. These data were presented by Yee and Mackenzie (2002) and Yee (2002, 2004) as an application of quantile regression. The response of interest is BMI (`BMI`) and the explanatory variable is age (`age`). We confine our analysis to a subset of 2600 European women and missing values are removed. P-spline VGAMs are performed using the `psvgam()` function and VGAMs are performed using the VGAM package.

Of interest is how BMI is affected by age. To investigate this, we fit the LMS-normal method to the data set. The model is formulated as

$$\boldsymbol{\eta}\,(x) \;=\; \left(\lambda(x), \mu(x), \log\left(\sigma\left(x\right)\right)\right)^T . \tag{6.6}$$

We fit the model above using P-spline VGAMs and VGAMs. The models and the corresponding calls are as follows:

(i) default VGAMs with 4 degrees of freedom for $\lambda(x)$, $\mu(x)$, and $\log(\sigma(x))$

```
fit.lms0 <- vgam(BMI ~ s(age), family = lms.bcn(zero = NULL),
                 data = bmi.dat)
```

(ii) VGAMs with 2, 4, and 2 degrees of freedom for $\lambda(x)$, $\mu(x)$, and $\log(\sigma(x))$ (Yee and Mackenzie, 2002)

```
fit.lms1 <- vgam(BMI ~ s(age, df = c(2, 4, 2)),
                 family = lms.bcn(zero = NULL),
                 data = bmi.dat)
```

(iii) P-spline VGAMs using penalized B-splines of degree 3, together with a second-order penalty, and 10 equally-spaced B-spline knots with the smoothing parameters being selected automatically through minimization of the UBRE score

```
fitps.lms <- psvgam(BMI ~ ps(age, 10), lms.bcn(zero = NULL),
                    data = bmi.dat)
```

Table 6.8: EDF estimates for each function obtained from (i) default VGAMs (`fit.lms0`), (ii) VGAMs ($\mathrm{df}_{(j)} = (2, 4, 2)$) (`fit.lms1`) and (iii) P-spline VGAMs (`fitps.lms`).

Model	$\widehat{\lambda}(x)$	$\widehat{\mu}(x)$	$\log(\widehat{\sigma}(x))$
(i) `s(age)`	3.9	3.9	4.0
(ii) `s(age,df=c(2,4,2))`	1.9	3.9	2.0
(iii) `ps(age,10)` ($\widehat{\lambda}_{(j)2}$)	1.0 (1.5×10^9)	2.8 (0.37)	2.0 (1.6×10^2)

Figs. 6.18 (a) – (c) show the fitted functions $\widehat{\eta}_1 = \widehat{\lambda}(x)$, $\widehat{\eta}_2 = \widehat{\mu}(x)$ and $\widehat{\eta}_3 = \log(\widehat{\sigma}(x))$ that each of the three models yield. Table 6.8 shows the EDF estimates for each function from the three models and the optimal smoothing parameters ($\widehat{\lambda}_{(j)2}$) obtained from P-spline VGAMs (`fitps.lms`). The EDF estimates obtained using P-spline VGAMs suggest that $\widehat{\lambda}(x)$ is linear, while $\widehat{\mu}(x)$ and $\log(\widehat{\sigma}(x))$ are nonlinear.

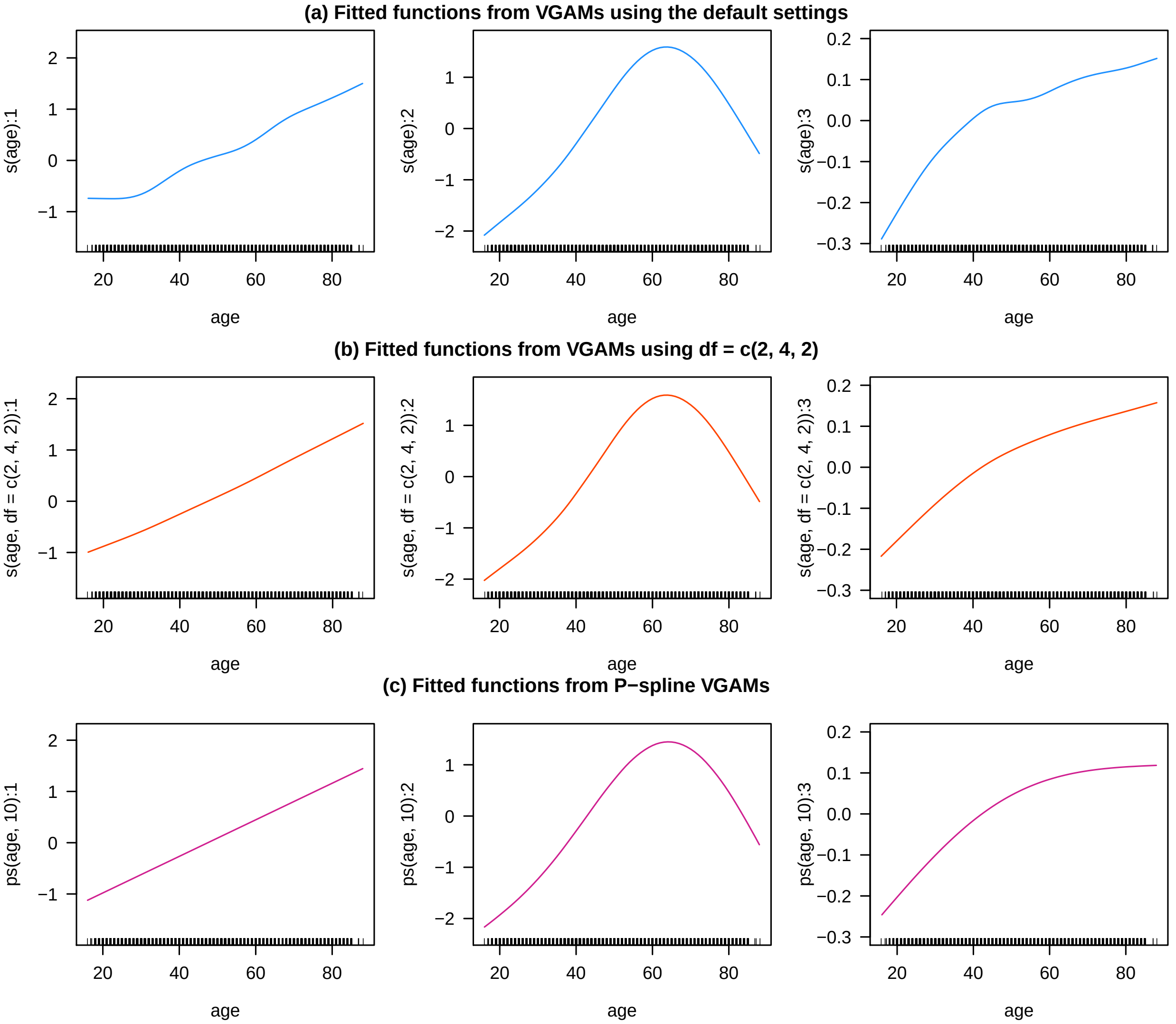


Figure 6.18: Fitted functions $\widehat{\eta}_1 = \widehat{\lambda}(x)$, $\widehat{\eta}_2 = \widehat{\mu}(x)$ and $\widehat{\eta}_3 = \log(\widehat{\sigma}(x))$ (see equation (6.6)) using (a) default VGAMs, (b) VGAMs ($\text{df}_{(j)} = (2, 4, 2)$) and (c) P-spline VGAMs fitted to the BMI data.

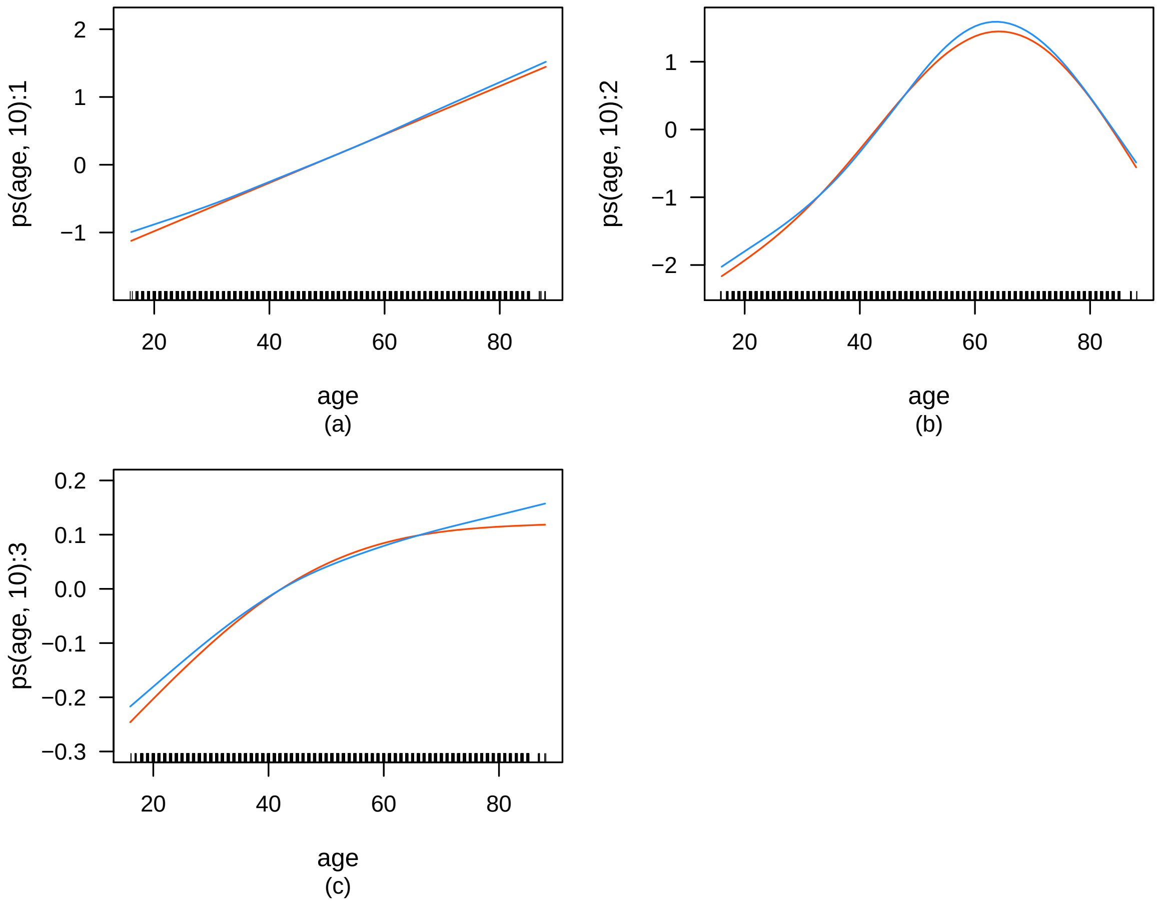
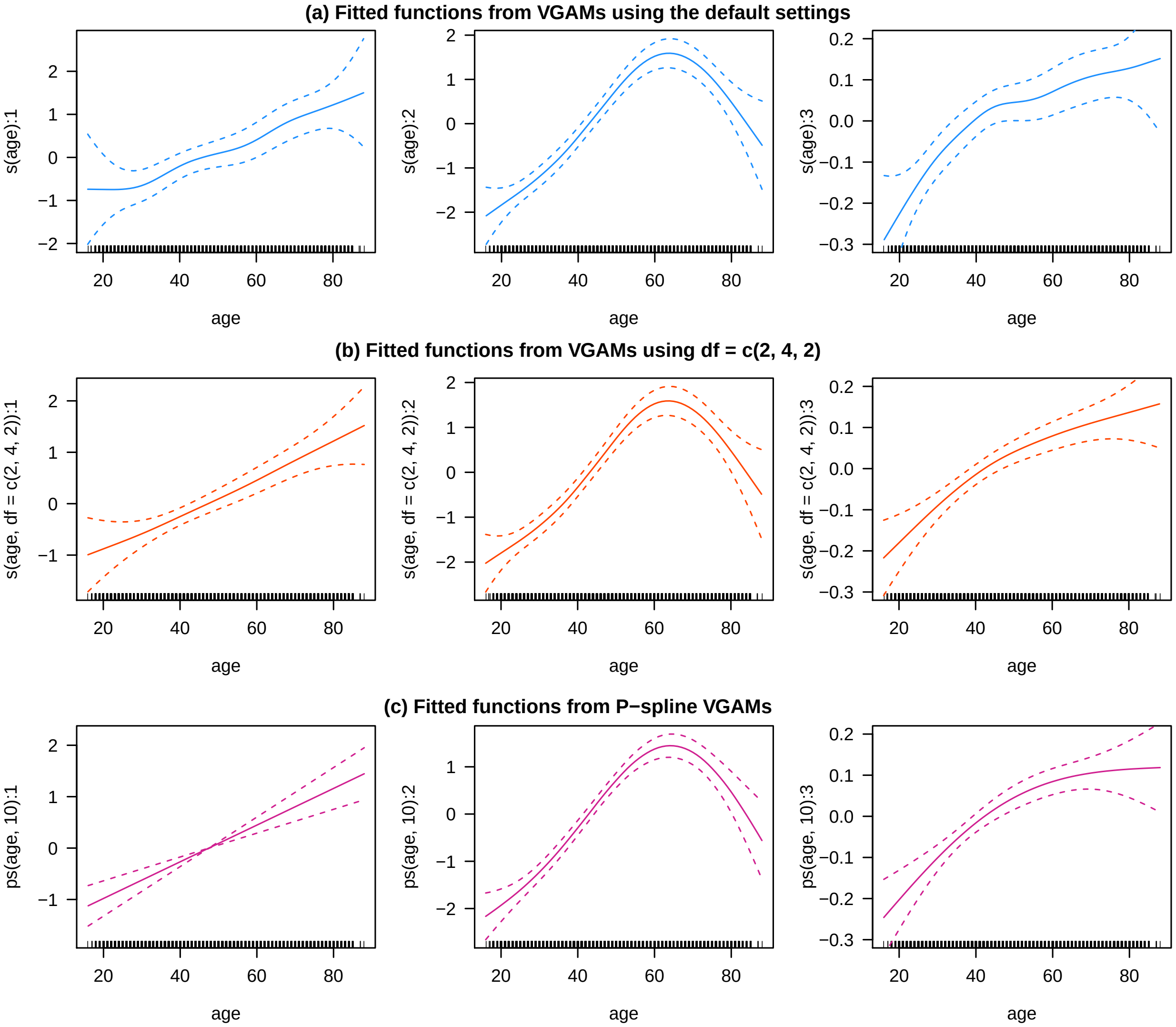


Figure 6.19: (a) - - (c) Fitted functions $\widehat{\eta}_1 = \widehat{\lambda}(x)$, $\widehat{\eta}_2 = \widehat{\mu}(x)$ and $\widehat{\eta}_3 = \log(\widehat{\sigma}(x))$. VGAMs ($\text{df}_{(j)} = (2, 4, 2)$) (`fit.lms1`) and P-spline VGAMs (`fitps.lms`) are overlaid and respectively given by the blue and red lines.

Fig. 6.18 ((a): left and right panels) shows that the fitted functions $\widehat{\lambda}(x)$ and $\log(\widehat{\sigma}(x))$ obtained from default VGAMs are more wiggly than they should be. Yee and Mackenzie (2002)) reduced the degrees of freedom for these functions to obtain a better fit (VGAMs ($\text{df}_{(j)} = (2, 4, 2)$ (`fit.lms1`) as shown in Fig. 6.18 (b). Overlaying plots of the fitted functions (Fig. 6.19) show that the method proposed automatically yields a fit close to the hand-tuned fit. Plots of the fitted functions and their 2-standard-error bands show that the confidence limits obtained from the method proposed (Fig. 6.20 (c)) are narrower than those of default VGAMs (Fig. 6.20 (a)).



(a) Fitted functions from VGAMs using the default settings

(b) Fitted functions from VGAMs using df = c(2, 4, 2)

(c) Fitted functions from P−spline VGAMs

Figure 6.20: Fitted functions $\widehat{\eta}_1 = \widehat{\lambda}(x)$, $\widehat{\eta}_2 = \widehat{\mu}(x)$ and $\widehat{\eta}_3 = \log(\widehat{\sigma}(x))$ (see equation (6.6)) and their 2-standard-error bands using (a) default VGAMs, (b) VGAMs ($\text{df}_{(j)} = (2,4,2)$) and (c) P-spline VGAMs fitted to the BMI data.

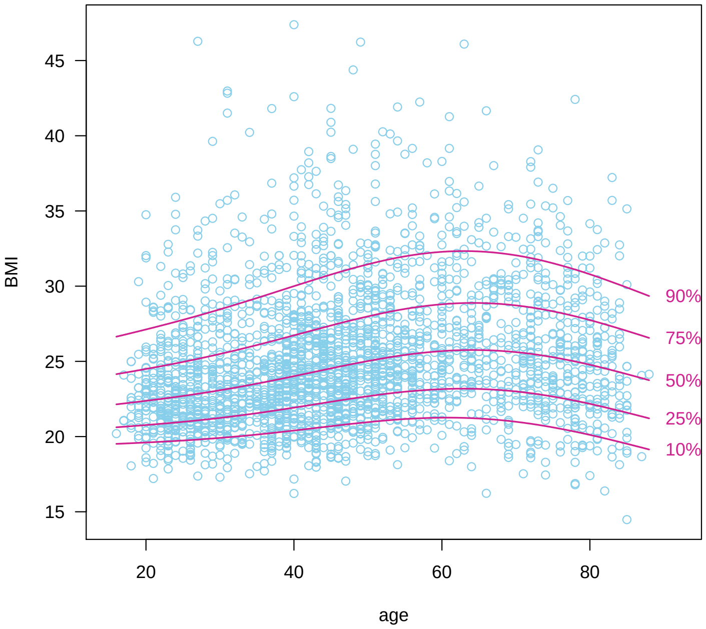


Figure 6.21: Quantile regression fits to dataset `xs.nz` in `VGAMdata` using the LMS method. The solid lines represent the estimated smooth quantiles obtained using a P-spline VGAM.

Fig. 6.21 shows the fitted quantile-curve that the P-spline VGAM yields. For fixed ages, the distribution of BMI in Fig. 6.21 is clearly negatively skewed. The plot shows that the median BMI of European women generally increases with age until their mid-60s and then the BMI decreases.

To find out whether the quantile curves for European men differ from European women, we then fit P-spline VGAMs (`fitps.lms`) to both male and female data. The fitted quantile-curve is given by Fig. 6.22. Not surprising, the plot indicates that the median BMI of European men is greater than that of European women of the same age.

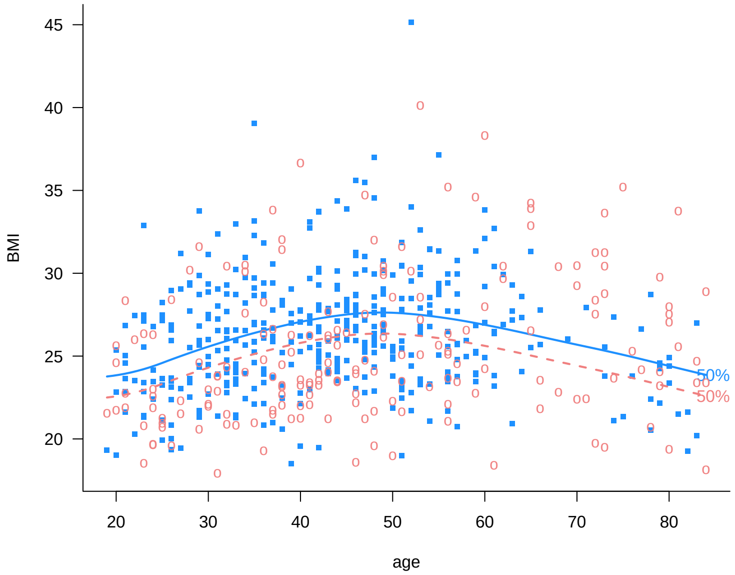


Figure 6.22: The fitted quantile-curve for European men and women using P-spline VGAMs. The solid line (blue) and squares are for men and the dashed line (pink) and circles are for women.

6.7 Conclusions

In this chapter, we have illustrated the new methods developed in this study and discussed their major advantages through the application to data from a cross-sectional workforce study combined with a health survey from New Zealand during the 1990s, and data from a survey study of the pregnancy and birth process during 1990 – 2004.

Our results show that P-spline VGAMs integrated with the automatic smoothness estimation performed well in real dataset for many multivariate response types and models in which their model structures involve constraints on the model terms and those that did not. The new method yields a more reasonable fit-curve as compared with default VGAMs based on backfitting. Overall, the nonparametric models yielded by the new methods approximate nonparametric models fitted using the VGAM approach with hand tuning. Our penalized likelihood framework together with automated smoothness estimation has a major advantage over VGAMs in eliminating the need for hand tuning.

Chapter 7

Conclusions and future work

The main purpose of this research study was to develop an alternative estimation procedure for the VGAM class based on the penalized likelihood approach of Eilers and Marx (1996), Marx and Eilers (1998) and Wood (2006b) for GAM modeling, and to integrate an automatic procedure for determining the degrees of smoothing for smooth terms from the data into the VGAM framework building on the GAM work of Wood (2006b). We discussed theoretical and practical aspects of GAMs based on penalized regression splines and summarized also VGLMs/VGAMs emphasizing elements relevant to this study. We then concentrated on developing new efficient methods based on penalized regression splines for estimating parameter coefficients for the full range of VGAM models, implementing an efficient computational method for automatic smoothing parameter selection into the VGAM framework, implementing the methods in R, investigating and comparing the practical performance of the method proposed to the default VGAMs method via simulations, and illustrating the new approach with multivariate response types and models.

More particularly, Chapter 4 showed how P-spline VGAMs can be represented using penalized regression splines and how they can be estimated, once a basis for the smooth functions has been chosen together with associated measures of function wiggliness. Given a set of B-splines as a basis, P-spline VGAMs are simply VGLMs, with an associated set of penalties. Model estimation was then developed using a penalized version of IRLS. The very important feature of constraint matrices, which allow the linear/additive predictors to share relationships with each other, etc. was able to be catered for in a natural way. The effective degrees of freedom (EDF) used for measuring the flexibility of the fitted model for P-spline VGAMs were defined and further tools that are useful for applied modeling with the purposed method such as comparing models were provided.

Chapter 5 developed automatic smoothing parameter selection for P-spline VGAMs. This was done by adapting the unbiased risk estimator (UBRE) methods of Wood (2004) and Marra and Radice (2011). Smoothing parameter estimation by the UBRE was included in the P-IRLS scheme by applying automatic UBRE optimization to the weighted least squares problem produced at each stage of the iterative least squares method. The degrees of freedom for each smooth term in the model were chosen simultaneously as part of model fitting. A simulation study showed that the automatic smoothness estimation performed well in terms of estimating smooth components for model structures involving constraints on the model terms and those that did not. Overall, the method proposed performs significantly better than the default VGAM method in terms of predictive ability.

Chapter 6 applied the proposed method to two real data sets with several multivariate response types and models and compared them to default VGAMs and VGAMs that were hand tuned to improve fitting performance. In all cases the proposed method performed in the desired way by automatically choosing smoothness that closely approximated the hand-tuned VGAMs chosen by Yee and co-authors. This was true for model structures that involved constraints on the model terms and those that did not.

The proposed framework can be employed for all simple exponential family distributions and most multivariate response types and models such as categorical response (multinomial logit model, proportional and non-proportional odds model), quantile regression (LMS method, e.g., Box-Cox to normal, Yeo-Johnson to normal distributions), expectile regression (asymmetric least squares, e.g., for normal, Poisson, binomial, exponential), Gumbel, bivariate binary responses (bivariate logistic model and bivariate probit model), zero-inflated Poisson, and multivariate regression, etc. However, numerical problems tend to occur for some models and distributions such as LMS method - Box-Cox to gamma. As Yee (2015a) stated, some models are harder to fit than others because of inherent numerical difficulties associated with them.

There are limitations in the method as presently implemented as following:

P-spline VGAMs proposed in this research may suffer occasionally from convergence problems such as sample sizes smaller than about 500 for the bivariate probit model with fairly high correlation between two responses. This is not surprising as Marra and Radice (2011) have already reported these problems for their recursive semiparametric bivariate probit model. If convergence problems occur, Yee (2015a) suggested assigning an initial value for the correlation ρ and monitoring convergence (e.g., set the argument `trace = TRUE`).

In our experience, convergence failure may occur occasionally with high-dimensional settings for the number of knots and/or the order of the penalty. In practice, choosing the number of knots between 5 and 15, and the order of the penalty between 1 and 3, is usually adequate. As Yee and Mackenzie (2002) stated, numerical problems are common, especially for $M > 1$ models, and in the VGLM framework, IRLS computation may fail on very large data sets since its design matrix requires too much memory. As explained in Section 4.2.2, both storage and time costs for fitting P-spline VGAMs increase rapidly with respect to M, followed by p, S_k, and then n. Storage can be reduced by reducing the number of parameters through imposing constraints on the functions and reducing the dimensions of number of knots (S_k). However, further work is required to reduce storage-use inefficiencies such as storing entire block diagonal matrices with specialized algorithms and data structures that take advantage of the sparse structure of the matrix.

Convergence failure may also sometimes occur due to an infinite cycling between the two steps of estimations, one for estimating $\boldsymbol{\beta}^*$ given smoothing parameters and another for estimating $\boldsymbol{\lambda}$ given $\boldsymbol{\beta}^*$ (cf. Wood (2006b) and Marra and Radice (2013)).

We have highlighted some further work needed to remedy limitations of the method as currently implemented. There are other interesting areas for further research including the following:

A fitted P-spline VGAM object is returned by the `psvgam` function and of the class "`psvgam`" inheriting from the classes "`vglm`" and "`vgam`". Method functions such as `summary`, `deviance`, `residuals`, `fitted`, `predict` and `plotvgam` therefore exist for the class "`psvgam`". But the fitting method for P-spline VGAMs is different to VGLMs and VGAMs, so that the elements relating to specific features of penalized regression splines, e.g., the B-spline basis and wiggliness penalty, and smoothing parameter estimation are not inherited and thus need to be written. Therefore, the method functions should be written to extract elements such as the penalty matrix for the models, the dimension of the B-spline basis used to represent the smooth, the order of the penalty, the estimated smoothing parameters of the smooth components, the minimized smoothing parameter selection score: UBRE, effective residual degrees of freedom of the model and number of iterations performed for the smoothing parameter estimation (relating to the `magic` part) of the fitting procedure. We note that the method functions for extracting the estimated degrees of freedom for each smooth term in a P-spline VGAM fit (`edfpsvlm()`) and the array of the elements from the leading diagonal of the influence matrix (`hatvaluespsvlm()`) have been written and are described in A.3.

In practice, when using penalized regression splines, users have to choose the number of knots that will be used in the model building process. Chapter 6 showed an example where the changing the number of knots gave a better fit. It would be advantageous to incorporate the ability to have a numerical procedure, e.g. minimizing AIC, to choose the number of knots.

Another extension would be to consider the inference parts of P-spline VGAMs such as confidence interval construction. It would be useful to be able to investigate how well the confidence intervals reliably represented the uncertainty of smooth terms for model parameters. Gu (2002), Ruppert et al. (2003) and Wood (2006a) generalized the well-known Bayesian confidence intervals originally introduced by Wahba (1983) or Silverman (1985) in the univariate spline model context to Gaussian non-Gaussian settings and GAM components. Marra and Radice (2011) used these results and constructed Bayesian confidence intervals for the components of their recursive semiparametric bivariate probit model. Marra and Wood (2012) showed by simulation and extension of Nychka's 1988 method that the Wahba/Silverman type Bayesian intervals for the smooth component functions of GAMs represented using any penalized regression spline approach have generally close to nominal 'across the function' frequentist coverage probabilities. The usual componentwise extension of Wahba/Silverman type intervals as discussed, for instance, by Gu (2002), Ruppert et al. (2003), Wood (2006a), Marra and Wood (2012) could be extended to find better confidence intervals for the P-spline VGAM family.

A trust region algorithm (see, e.g., Nocedal and Wright (1999, section 4.2), Marra et al. (2013b), Radice et al. (2015)) might be applied to make the maximization of the P-spline likelihood function more reliable. At present the `trust` package by Geyer (2014) implements this approach. This algorithm evaluates an eigen-decomposition of Fisher information matrix at each iteration (see Marra et al., 2013b).

In this study, we reformulated VGAMs based on only one type of penalized regression splines (P-spline smoothers). A variety of alternative smoothers based on splines are available. This spline bases also have fairly convenient mathematical properties and good numerical stability. The `mgcv` package by Wood offers several other types of smoothers: thin plate regression splines, thin plate regression splines with shrinkage-to-zero, cubic regression splines, cubic regression splines with shrinkage-to-zero, cyclic cubic regression splines, cyclic P-splines, additive smooths of 1 or 2 variables, simple random effect terms, Markov random field smoothers for smoothing over discrete districts and tensor product smooths. In his package, thin plate regression splines are given as the default smooth for `s` terms within `gam` model formulas.

Thin plate regression splines are constructed by first constructing the basis and penalty for a full thin plate spline and then the space of the wiggly components of this basis is truncated in an optimal manner, to obtain a low rank smoother (see Wood, 2003). The major advantages of thin plate regression splines are that they avoid the knot-replacement problems that can substantially influence modeling with penalized regression and they provide a sensible manner of modeling interaction terms in GAMs. Another interesting smoother is cubic regression splines. For the cubic regression splines, the values at adjacent knots are connected through sections of cubic polynomial under the conditions that the spline must be continuous to the second derivative at the knots. A smooth curve is obtained, which is a natural cubic spline combining through the values at the knots. The extra conditions that the spline should have zero second derivative at the two end knots are imposed. This prevents unstable end effects on the spline. Full details about cubic regression splines are given in Wood (Sections 4.1.2 and 4.1.3, 2006b).

To illustrate the use of such smoothers for GAM modeling, we used the diabetic retinopathy data from Bender and Grouven (1998). The response outcome of interest is the `presence` (`1`) or the `absence` (`0`) of diabetic retinopathy. The available regressors are diabetes duration in years (`DIAB`), glycosylated hemoglobin measured in percent (`GH`), and diastolic blood pressure in mm Hg (`BP`). The data are available in the data frame `retinopathy` from `catdata`. There are 613 observations, resulting in 225 presences and 388 absences. We are interested in investigating the relationship between diabetic retinopathy and the three predictor variables. A logistic additive model is used to describe the conditional probability of diabetic retinopathy given the predictor variables. We then fitted the additive logistic model to the three predictors using the function `gam()` in the `mgcv` package. The models fitted used, respectively: thin plate regression splines, cubic regression splines and P-splines. The partial contributions of each predictor to the conditional probability of diabetic retinopathy with 95% Bayesian intervals obtained from the three models are shown in Figs. 7.1 (a) – (c). The results are very similar except for the middle panel of Fig. 7.1 (c), where the P-splines are displayed more extreme end behavior.

Other penalized regression splines can also be, in principle, used to construct smooth functions of the explanatory variables of VGAMs. As with P-spline VGAMs, it is possible to set up a model

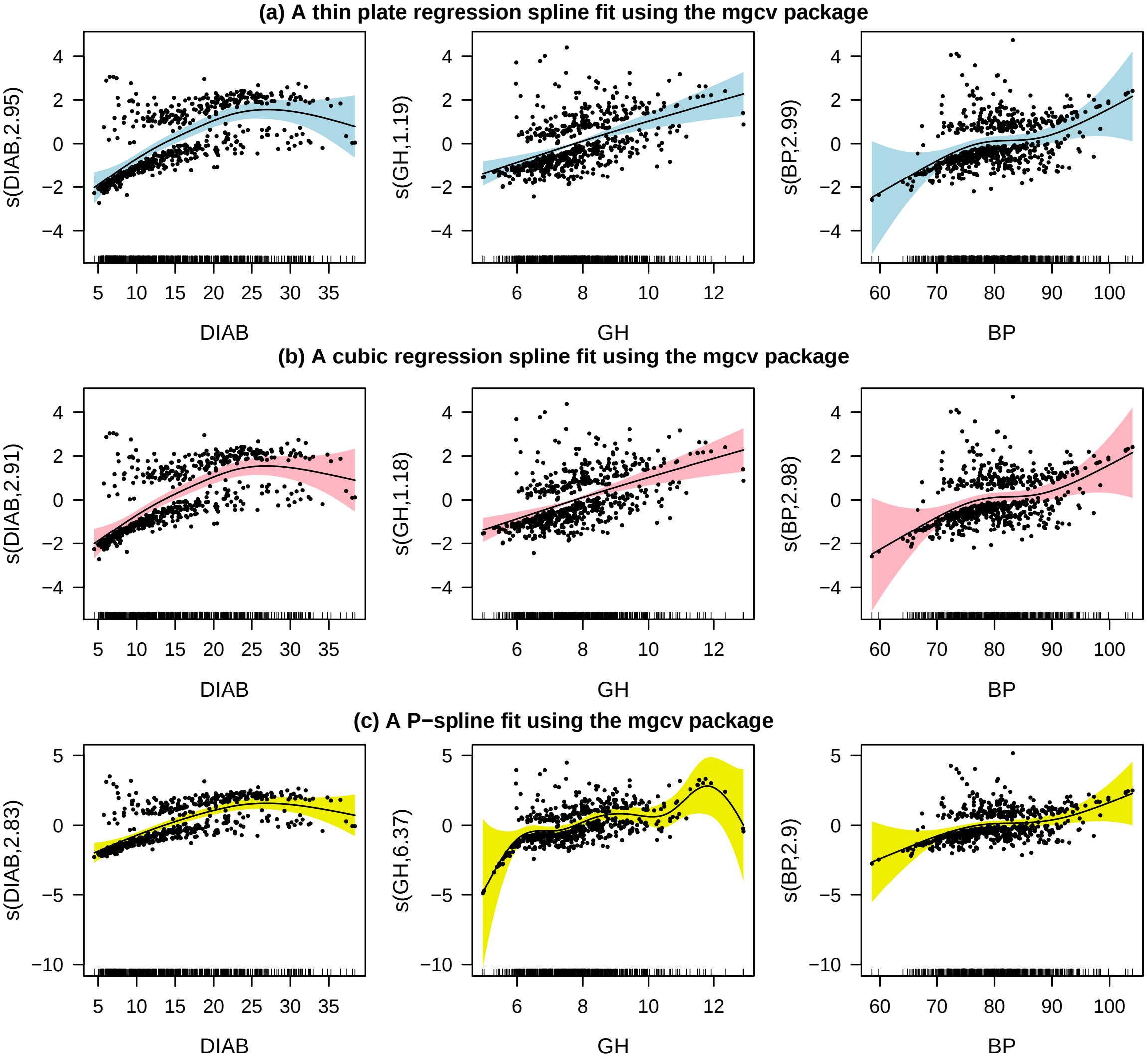


Figure 7.1: (a) - (c): The estimated smooth terms of a thin plate regression spline, a cubic regression spline fit and a P-spline fit with partial residuals and 95% Bayesian intervals using the `mgcv` package.

matrix and wiggliness penalty matrix for each smooth function using these splines. The model matrix for the whole model as well as the penalty for the model would be in the same basic form as for $\mathbf{X}_{\text{VAM}}$ and $\mathbf{P}^*_\lambda$ (cf. equations (4.43) and (4.46)), and the parameter vector would be given by $\boldsymbol{\beta}^*$. Given the model matrix and penalties, the coefficients and smoothing parameters of these penalized regression splines could be obtained as P-spline VGAM components using the methods of Sections 4.2 and 5.3. However, it is more difficult to construct the basis and an associated penalty. P-splines were used in our approach because they are easier to construct being that low-rank smoothers using B-spline basis functions, their estimation uses a lower dimensional system of equations, which is less expensive computationally, they are easy to set up and use, and provide great flexibility in that users can combine any order of B-spline basis with any order of the penalty (see Section 2.3). However, P-splines allow only for equally-spaced knots, so that the problems arise if uneven knot spacing is required, and their penalties are difficult to interpret in terms of the properties of the fitted smooth as compared to the derivative penalties, e.g., $\int \{f_k^{'}(x_k)\}^2 \, \mathrm{d}x_k$ or $\int \{f_k^{''}(x_k)\}^2 \, \mathrm{d}x_k$. Wood (2007) stated that, in practice, splines with derivative penalties (as used with thin plate regression or cubic regression splines) perform slightly better than P-splines with discrete penalties. It would therefore be advantageous to extend the work done here to other forms of smoother.

Recently, Donat and Marra (2015) proposed estimation methods for fitting a semi-parametric bivariate polychotomous ordinal regression. They dealt with a bivariate polychotomous random variable defined as $\mathbf{Y} = (Y_1, Y_2)^T$, where Y_j, $j = 1, 2$, is “ordinal” (measured on the ordinal scale). They represented the additive non-parametric effects of the explanatory variables using penalized regression splines. They obtained model formulation, which can be specified as an instance of the class of a penalized GLM, and therefore estimation and inference can be done by a natural extension of GLMs. It would be interesting to reformulate this problem in terms of penalized regression-spline VGAMs.

Finally, it would also be useful to explore a mixed model approach to estimation and inference with P-spline VGAMs. In the penalized spline context, several authors such as Eilers (1999), Wand (2003), Ruppert et al. (2003), Wand and Ormerod (2008) and Wood (2011), to name a few recast the model formulation into a mixed model formulation. In this approach, the penalized regression smoothers are written as components of a mixed model while treating their smoothing parameters as variance component parameters to be estimated by Likelihood, REML or PQL methods. These ideas should be able to be extended to VGAMs.

Appendix A

Implementation details

This appendix covers the use of the P-spline VGAM functions and briefly discusses some of the implementation details for fitting P-spline VGAMs. The functions for fitting P-spline VGAMs have been coded in R. The main functions are mainly developed from P-spline GAMs and VGLMs/VGAMs. The primary function `psvgam()` and the support functions called `psvglm.fit()` and `psvlm.wfit()` were adapted from `vgam()`, `vglm()`, `vglm.fit()` and `vlm.wfit()` in the VGAM package. The `psvgam()` function is very much like the `vglm()` function. The main differences are that the `psvgam` model formula can include smooth terms, `ps()`, adapted from Marx and Eilers's (1998) `ps()` function, and a numerical procedure for controlling automatic smoothness selection of Wood (2004) has been incorporated into model fitting. New functions, `Pen.vps()` and `Xps2Xmagic()`, were developed respectively to construct the penalty matrix for P-spline VGAMs and the necessary quantities required for smoothing parameter estimation of Wood (2004). The extractor functions for the "hat values" and the "effective degrees of freedom" of a fitted P-spline VGAM were also developed. The functions presently implemented for fitting P-spline VGAMs are too long to include in print form here. They are available from the author and are also contained at https://www.stat.auckland.ac.nz/∼csom017/P-spline VGAMs.

A.1 R Functions and objects for fitting P-spline VGAMs

We now discuss aspects of the use of the R functions developed for fitting and understanding P-spline VGAMs. Readers familiar with VGLMs/VGAMs will soon discover that the tools are much the same. In this research, the `psvgam()` function is used to fit P-spline VGAMs, where the additive predictors of the models can be flexibly specified using parametric and regression spline components. The underlying representation and estimation of the models are based on the penalized regression spline approach. The degrees of smoothness of model terms is estimated as part of model fitting.

A.1.1 Fitting the models

Here, we discuss the use of `psvgam`. Details of this function are given in A.2.1. A call to the function `psvgam()` in its simplest form looks like

```
psvgam(formula, family, method = "psvglm.fit")
```

The argument `formula` provides a symbolic description of the model to be fitted while the argument `family` specifies the distribution and link to use in fitting. The argument `method` specifies the method to be used in fitting the model, where the default method `psvglm.fit` uses penalized iteratively reweighted least squares (P-IRLS) with integrated automatic multiple smoothing parameter selection.

Let us revisit the non-exchangeable bivariate logistic model from Section 6.4. We fitted the model using 5 equally spaced B-spline knots as in the expression (cf. equation (6.4)):

```
fitps.cd1 <- psvgam(cbind(cat, dog) ~ ps(age, ps.interval = 5),
                     binom2.or(zero = NULL),
                     data = women.eth0.catdog)
```

In the expression above, the `psvgam()` function fits the coefficients of the nonparametric bivariate logistic model via the `binom2.or` family (a bivariate odds-ratio model) using penalized maximum likelihood.

The model formula in the call

```
cbind(cat, dog) ~ ps(age, ps.interval = 5)
```

specifies a single P-spline smooth term in `age` specified using `ps()`, where the number of knots is given by `ps.interval = 5`. In the model formula, users can mix smooth terms with linear terms or factors as in (cf. the bivariate logistic fits to pregnancy and birth data in Section 6.5)

```
cbind(Instrument, Induced) ~ ps(Age, ps.interval = 8) + Weight
```

or

```
cbind(Instrument, Induced) ~ ps(Age, ps.interval = 8) + poly(Weight, 5)
```

The implementation of `psvgam()` does not currently accommodate interaction between smooth terms, although in principle this is possible. A term using `ps()` such as `ps(Age, ps.interval = 8)` does not perform any smoothing itself. It exists to setup a model using P-spline based smooths. Details about `ps()` are given in A.2.2.

Let us revisit the additive fit `fitps.cd1`. The summary of the fit shows that the convergence is obtained in 11 iterations. We set `trace = TRUE` in order to produce the output for each iteration. The results for first three iterations with the estimates of the smoothing parameters are shown as following:

```
1  > fitps.cd1 <- psvgam(cbind(cat, dog) ~ ps(age, 5),
2  +                     binom2.or(zero = NULL),
3  +                     data = women.eth0.catdog,
4  +                     crit = "coef", trace = TRUE)
5  VGLM    linear loop  1 :  coefficients =
6   0.013234722, -0.917866874,  0.381471390,  0.312677101,  0.225757142,
7  -0.432622906,  0.526406214,  0.415658125, -0.505238556,  0.293005851,
8   0.249689839, -0.175046401, -0.380256752, -0.259556686,  0.652708805,
9  -1.316084793, -0.953401040,  1.887418832, -2.372722142, -1.733072365,
10  3.286363877
11 ps(age, 5)1 ps(age, 5)2 ps(age, 5)3
12  0.07325548  0.05077608  0.07791104
13 VGLM    linear loop  2 :  coefficients =
14 -0.0056273174, -1.0625477262,  0.3776413387,  0.3324734601,
15  0.3365296603, -0.1996268452,  0.5588976001,  0.6522986468,
16 -0.1599340898,  0.3192457188,  0.4278190220,  0.0013913776,
17 -0.4045040311, -0.3670054811,  0.3422767453, -1.4475013923,
18 -1.5327268984,  0.8553482219, -2.6440324184, -2.8909273076,
19  1.4252250502
20 ps(age, 5)1 ps(age, 5)2 ps(age, 5)3
21  0.09346089  0.04510848  0.16136868
22 VGLM    linear loop  3 :  coefficients =
23 -0.0079569429, -1.1001849949,  0.3053019195,  0.3357160526,
24  0.3703824671, -0.1280641056,  0.5664698670,  0.7233779425,
25 -0.1364310970,  0.3232899338,  0.4799003814, -0.0374558829,
26 -0.4071983949, -0.4032012834,  0.2005268489, -1.4538418282,
27 -1.7065145834,  0.5582655539, -2.6519616299, -3.2276686303,
28  0.9627038709
29 ps(age, 5)1 ps(age, 5)2 ps(age, 5)3
30  0.08480635  0.03402325  0.18100675
```

At each P-IRLS iteration a penalized weighted least squares problem is solved, and the smoothing parameters of that problem are estimated by the UBRE using the `magic()` function. The estimates of the smoothing parameters for $\widehat{f}_{(j)2}(x_2)$, $j = 1, 2, 3$ (see equation (6.4)) at each iteration are displayed in lines 12, 21 and 30. For example, `ps(age, 5)1` in line 11 indicates the estimates of smoothing parameters for $\widehat{f}_{(1)2}(x_2)$, at the 1^{st} iteration, which approximate to 0.07325548.

There are several generic functions in VGAM for extracting single components from the fitted `psvgam()` object such as `coef()`, `deviance()`, `residuals()`, etc. and these work seamlessly with `psvgam` output.

```
> coef(fitps.cd1, matrix = TRUE)
              logit(mu1) logit(mu2) log(oratio)
(Intercept) -0.008375495 -1.1049905  0.29645370
ps(age, 5)2  0.336810890  0.3752026 -0.12222140
ps(age, 5)3  0.572013304  0.7431335 -0.17106830
ps(age, 5)4  0.328046465  0.4976813 -0.08224011
ps(age, 5)5 -0.406477936 -0.4047798  0.16839995
ps(age, 5)6 -1.456850235 -1.7328450  0.54524015
ps(age, 5)7 -2.658805107 -3.2813490  0.97937050
```

Similarly, a deviance value can be obtained for the model:

```
> deviance(fitps.cd1)
[1] 6178.076
```

A.1.2 Plotting the fitted models

We can check the result by plotting the fitted model object. The component functions of a `psvgam()` object can be plotted using `plotvgam` as follows.

```
mycol <- c("dodger blue", "orange red", "limegreen")
mymain <- c("(a)", "(b)", "(c)")
par(mfrow = c(1, 3), mar =  c(5, 4, 1, 1) + 0.1, las = 1)
for (ii in 1:3) {
    plotvgam(fitps.cd1, which.cf = ii, se = TRUE, scale = 4,
          lcol = mycol[ii], scol = mycol[ii], rcol = "dark orange",
          sub = mymain[ii], cex.lab = 1.5,
          cex.axis = 1.5, cex.sub = 1.5)}
```

The resulting plot is displayed in Fig A.1. The plots show the estimated effects as solid curves with their 2-standard-error bands shown as dashed lines. The coincidence of the confidence limits and the estimated solid line at the point where the line passes through zero on the y-axis, is a result of applying the identifiability constraints to the smooth terms. The rug plots represented at the bottom of each plot show the values of the covariates of each smooth. Full details of the more useful arguments for `plotvgam` can be found in Yee (Section 8.4.4, 2015b).

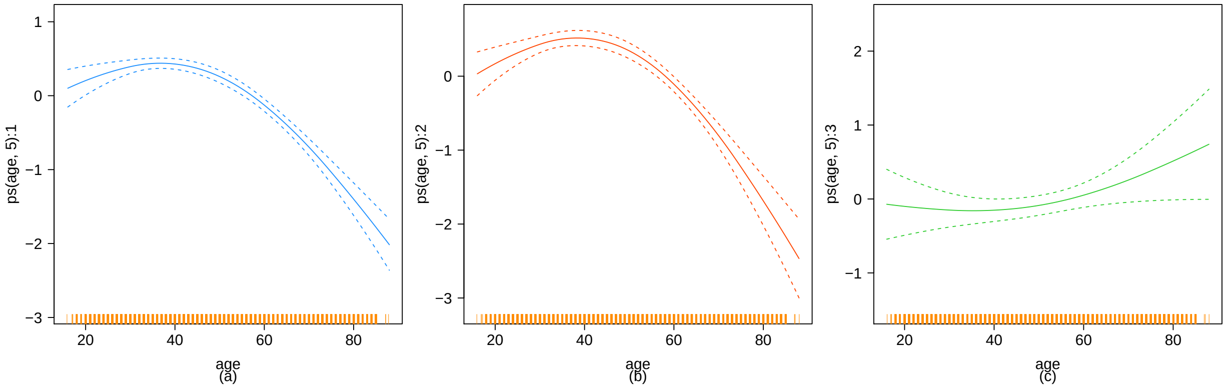


Figure A.1: Fitted functions $\widehat{f}_{(j)2}(x_2)$, $j = 1, 2, 3$ (see equation (6.4)) using P-spline VGAMs fitted to a subset of European women with household cat and dog pet ownership data.

A.1.3 Prediction

The `predict()` method function for `vgam` enables a `psvgam` fitted model object to be used for prediction at new values of the model covariates. We revisit the `fitps.cd1` to show this feature. For the predictions on the scale of the linear predictor ($\widehat{\eta}_{ij}$), we use:

```
> predict(fitps.cd1, type = "link")[1:5,]
   logit(mu1) logit(mu2) log(oratio)
5   0.1462018 -1.0016191   0.2099265
10  0.4202110 -0.5916896   0.1445043
13  0.3618802 -0.6430008   0.1679122
20  0.4257794 -0.5887917   0.1417692
24  0.2771795 -0.7320893   0.1993012
```

For the prediction on the response scale ($\widehat{\mu}_{ij}$), we use:

```
> predict(fitps.cd1, type = "response")[1:5,]
          00        01        10        11
5  0.3492079 0.1143066 0.3821689 0.1543166
10 0.2631173 0.1333490 0.3806354 0.2228983
13 0.2781841 0.1323204 0.3772474 0.2122481
20 0.2618460 0.1332886 0.3812419 0.2236235
24 0.3017915 0.1293538 0.3734720 0.1953826
```

Similarly, the `fitted()` method function extracts the fitted values. We note that the terms `00, 01, 10, 11` from the results above correspond to the joint probability p_{00}, p_{10}, p_{10} and p_{11} respectively (cf. Section 3.3.2).

A.1.4 P-spline VGAMs with constraints

As with VGLMs/VGAMs, there are two ways of fitting P-spline VGAMs with constraints. The first is to use family function-specific arguments such as `parallel`, `exchangeable` and `zero`. For example, analogous to the non-exchangeable bivariate logistic model `fitps.cd1`, the exchangeable version of the bivariate logistic model with an intercept-only log odds-ratio can be created by the call

```
fitps.cd2 <- psvgam(cbind(cat, dog) ~ ps(age, ps.interval = 5),
                    family = binom2.or(zero = 3, exchangeable = TRUE),
                    data = birth.b)
```

Here, the `zero` argument specifies which linear predictors are to be modeled with an intercept term only (here is η_3). The second way of fitting P-spline VGAMs with constraints is to use the `constraints` argument. A list of constraint matrices per term of a `psvgam()` object can be extracted using `constraints()` (cf. equation (4.42)):

```
> constraints(fitps.cd2, type = "term")
$‘(Intercept)‘
     [,1] [,2]
[1,]    1    0
[2,]    1    0
[3,]    0    1

$‘ps(age, 5)‘
     [,1]
[1,]    1
[2,]    1
[3,]    0
```

The full detailed documentation of VGLMs/VGAMs with constraints can be found in Yee (Section 3.3.1, 2015b).

A.1.5 Degrees of freedom

In this research, we developed an extractor function for the hat values and the effective degrees of freedom (EDF) of a fitted `psvgam()` object (cf. the EDF for P-spline VGAMs in Section 4.4). The function `edfpsvlm()` extracts the EDF associated with each penalty in a P-spline VGAM fit:

```
> edfpsvlm(fitps.cd1)
ps(age, 5):1 ps(age, 5):2 ps(age, 5):3
    1.935848     2.047130     1.347955
```

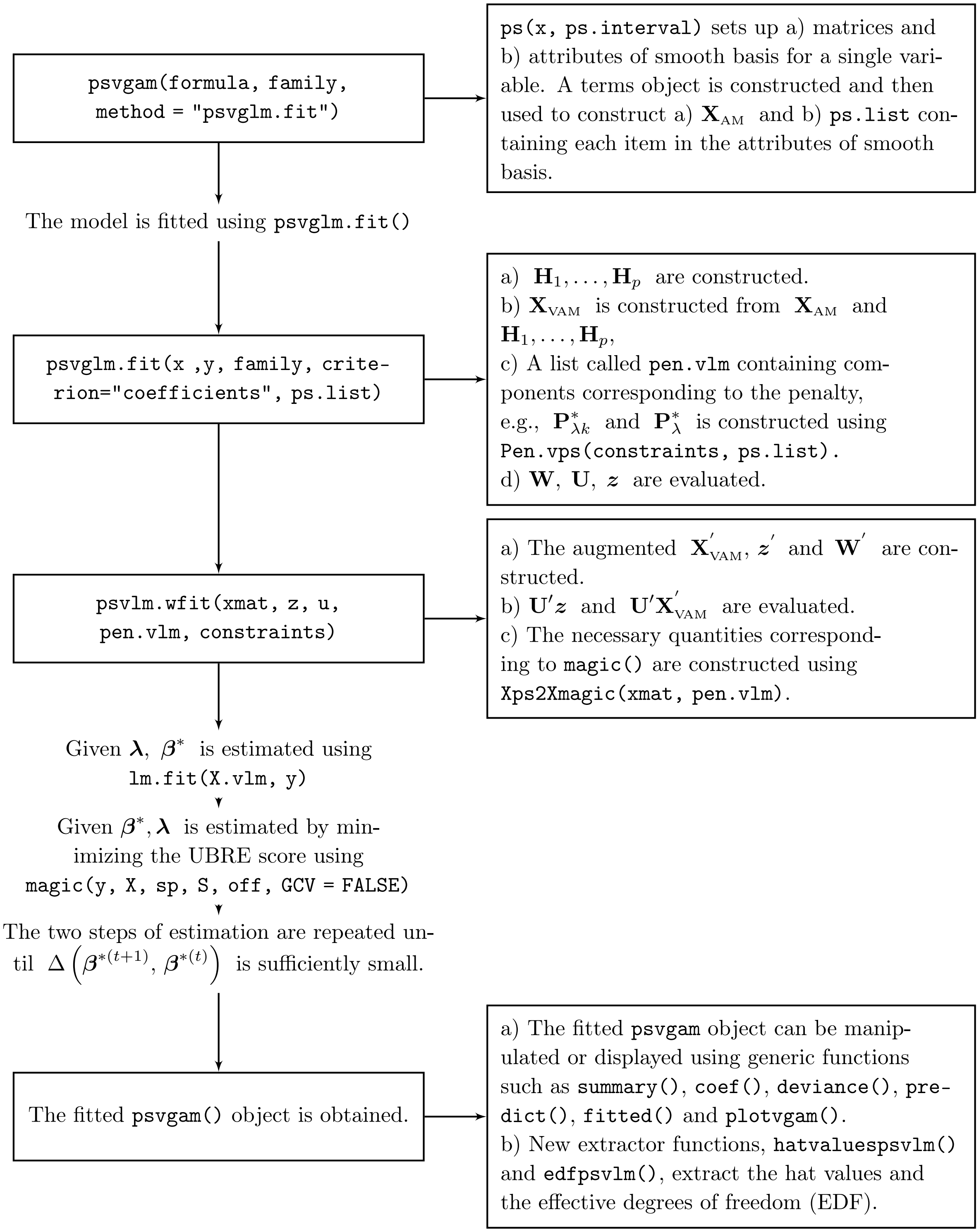
The results above show the EDF estimates for each smooth term, e.g., the EDF estimates for $f_{(1)2}(x_2) \approx 1.94$, and the EDF estimates for $f_{(2)}(x_2) \approx 2.05$ (cf. equation (6.4)).

A.2 Implementation details

Here, we provide a high-level overview of the functions and how they fit together. Standard R help files are provided in pdf form at https://www.stat.auckland.ac.nz/∼csom017/P-spline VGAMs.

The beginning of `psvgam()` is almost identical to that of `vgam()` or even `vglm()`. This starts with the `terms()` function. The `terms()` function takes a formula and the **"ps"** marked as special in the `specials` argument, and constructs a terms object. The terms object can then be used to construct a model matrix ($\mathbf{X}_{\text{AM}}$, cf. equation (4.10)). As we have seen, the `ps()` function exists to help to set up matrices and attributes of smooth basis using P-spline based smooths. Any matrices evaluated by `ps()` become included in the model matrix $\mathbf{X}_{\text{AM}}$. A list containing each item in the attributes of smooth basis is obtained.

This information is passed to the `psvglm.fit()` function to perform penalized iteratively reweighted least squares (P-IRLS). The constraint matrices $\mathbf{H}_1, \ldots, \mathbf{H}_p$ are obtained and $\mathbf{X}_{\text{VAM}}$ (cf. equation (4.43)) is then constructed from $\mathbf{X}_{\text{AM}}$ and $\mathbf{H}_1, \ldots, \mathbf{H}_p$. A list containing components corresponding to the penalty, e.g., $\mathbf{P}^*_{\lambda k}$ and $\mathbf{P}^*_{\lambda}$ (cf. equation (4.46)) is then obtained using the `Pen.vps()` function. The working weighted $\mathbf{W}$, a Cholesky decomposition of $\mathbf{W}$ and the adjusted dependent variable $\boldsymbol{z}$ are evaluated. This information is passed to the `psvlm.wfit()` function to perform the two steps of estimation. An internal function of `psvlm.wfit()`, called `Xps2Xmagic()` then constructs the necessary quantities required for the computational procedure of Wood (2004) (using the `magic()` function from the `mgcv` package). The P-IRLS algorithm consisting of the two steps cycles around: (i) given $\boldsymbol{\lambda}$, the estimates of $\boldsymbol{\beta}^*$ is obtained by solving a generalized least squares problem using the data augmentation on the adjusted dependent variable ($\boldsymbol{z}'$), regressors ($\mathbf{X}'_{\text{VAM}}$), and weights ($\mathbf{W}'$) (using the `lm.fit()` function), and (ii) given $\boldsymbol{\beta}^*$, $\boldsymbol{\lambda}$ is estimated by minimizing the UBRE score using the `magic()` function. The two steps are repeated until the relative change in parameter estimates sufficiently small. The entire procedure for fitting P-spline VGAMs is summarized in the Flowchart Fig. A.2. We give more details of each function used in A.2.1 – A.2.6.



`psvgam(formula, family, method = "psvglm.fit")`

`ps(x, ps.interval)` sets up a) matrices and b) attributes of smooth basis for a single variable. A terms object is constructed and then used to construct a) $\mathbf{X}_{\text{AM}}$ and b) `ps.list` containing each item in the attributes of smooth basis.

The model is fitted using `psvglm.fit()`

`psvglm.fit(x ,y, family, criterion="coefficients", ps.list)`

a) $\mathbf{H}_1, \ldots, \mathbf{H}_p$ are constructed.
b) $\mathbf{X}_{\text{VAM}}$ is constructed from $\mathbf{X}_{\text{AM}}$ and $\mathbf{H}_1, \ldots, \mathbf{H}_p$,
c) A list called `pen.vlm` containing components corresponding to the penalty, e.g., $\mathbf{P}^*_{\lambda k}$ and $\mathbf{P}^*_{\lambda}$ is constructed using `Pen.vps(constraints, ps.list)`.
d) $\mathbf{W}$, $\mathbf{U}$, $\boldsymbol{z}$ are evaluated.

`psvlm.wfit(xmat, z, u, pen.vlm, constraints)`

a) The augmented $\mathbf{X}^{'}_{\text{VAM}}$, $\boldsymbol{z}^{'}$ and $\mathbf{W}^{'}$ are constructed.
b) $\mathbf{U}'\boldsymbol{z}$ and $\mathbf{U}'\mathbf{X}^{'}_{\text{VAM}}$ are evaluated.
c) The necessary quantities corresponding to `magic()` are constructed using `Xps2Xmagic(xmat, pen.vlm)`.

Given $\boldsymbol{\lambda}$, $\boldsymbol{\beta}^*$ is estimated using `lm.fit(X.vlm, y)`

Given $\boldsymbol{\beta}^*, \boldsymbol{\lambda}$ is estimated by minimizing the UBRE score using `magic(y, X, sp, S, off, GCV = FALSE)`

The two steps of estimation are repeated until $\Delta\left(\boldsymbol{\beta}^{*(t+1)}, \boldsymbol{\beta}^{*(t)}\right)$ is sufficiently small.

The fitted `psvgam()` object is obtained.

a) The fitted `psvgam` object can be manipulated or displayed using generic functions such as `summary()`, `coef()`, `deviance()`, `predict()`, `fitted()` and `plotvgam()`.
b) New extractor functions, `hatvaluespsvlm()` and `edfpsvlm()`, extract the hat values and the effective degrees of freedom (EDF).

Figure A.2: The procedure for fitting P-spline VGAMs

A.2.1 The psvgam() function

The psvgam() is used to fit a P-spline VGAM. The additive predictors of the models can be specified using parametric terms and regression spline components (P-spline smoothers by the ps() function). The numerical routines used for fitting are an adaptation of IRLS algorithm from the VGAM package in combination with an adaptation of the smoothness estimation fitting procedure from the mgcv package.

In general, the psvgam() function works by first calling the terms() function. The terms() function takes a formula and the ‘‘ps’’ marked as special in the specials argument. The ps() function is called to set up a set of B-splines and its associated ‘discrete’ penalty for each smooth term in the model formula. The model matrix $\mathbf{X}_{\text{AM}}$ and a set of penalty matrices for the smooth terms are then obtained by psvgam. This information is passed to the psvglm.fit() function. The function then solves the penalized maximum likelihood with integrated automatic multiple smoothing parameter estimation. The call sequence of the psvgam() function is of the form of

```
psvgam <- function(formula, family, data, weights, subset, na.action,
                   etastart, mustart, coefstart,
                   control = vglm.control(maxit = 50,...),
                   offset, method = "psvglm.fit", ...)
```

The psvgam() function has the same arguments as Yee (2015a)’s vgam() function, except that, (i) in place of vgam formulae, we used ps() in the definition of (vector) smooth terms, and (ii) we employed the method psvglm.fit for the default method in place of vglm.fit. As with the vglm() function, psvgam() is accompanied by vglm.control() which provides default values for algorithmic variables used to control the numerical options for fitting P-spline VGAMs, e.g. maxit, the maximum number of IRLS iterations and epsilon, the tolerance in the convergence criterion between two successive iteration (see vglm.control() for full details). The new arguments are defined as follows:

Arguments

`formula` a symbolic description of the model to be fit. The RHS of the formula is applied to each linear/additive predictor, and usually includes at least one `ps` term. Different variables in each additive predictor can be chosen by specifying constraint matrices.

`method` the method to be used in fitting the model. The default method `psvglm.fit` uses P-IRLS with smoothing parameters associated with each penalty chosen by minimizing the UBRE score using the `magic()` function.

The function returns an object of the class "`psvgam`", which has the same slots as the "`vglm`" class (see `vglm-class` for further information).

A.2.2 The `ps()` function

The `ps()` function is used in definition of (vector) smooth terms within `psvgam` model formulae. This function is a modification of Marx and Eilers's (1998) `ps()` function. The major modification is to include an argument to specify smoothing parameters, which can be either scalar or vector. The call sequence of this function is of the form of

```
ps <- function(x, ps.intervals = NULL, lambda = 0, degree = 2,
               order = 2, ridge.adj = 1e-005, ridge.inv = 1e-004)
```

Following Marx and Eilers (1998), the `ps()` function has arguments as follows:

Arguments

`ps.interval` the number of equally-spaced B-spline intervals (the number of knots).

`lambda` a (vector) smoothing parameter.

`degree` the degree of B-splines.

`order` order of the penalty.

`ridge.adj` small positive numbers to stabilize linear dependencies among B-spline bases.

`ridge.inv` same as for `ridge.adj`.

The function returns a matrix with attributes of the number of knots, the degree of the B-splines, order of the penalty, a set of penalty matrices for the smooth terms, and (vector) smoothing parameters, that are used by `psvgam`.

A.2.3 The `psvglm.fit()` function

The `psvglm.fit()` function is an internal function of the `psvgam()` function. It is a modification of the `vglm.fit()` function. The major modification is that instead of solving a weighted least squares problem at each IRLS step, a weighted, penalized least squares problem is solved at each IRLS step with smoothing parameters associated with each penalty selected by the UBRE score using the `magic()` function from the `mgcv` package (see the `magic()` function for further information of stable multiple-smoothing-parameter estimation by GCV or UBRE).

The `psvglm.fit()` function works by first constructing a list of constraint matrices, $\mathbf{H}_1, \ldots, \mathbf{H}_p$, and then the "vector additive model" model matrix $\mathbf{X}_{\text{VAM}}$ is constructed. The penalty matrix for the P-spline VGAM model is obtained using an internal function of the `psvglm.fit()` function, namely the `Pen.vps()` function. This information is passed to the `psvlm.wfit()` function to perform the two steps of estimation, one for estimating $\boldsymbol{\beta}^*$ given smoothing parameters and another for estimating $\boldsymbol{\lambda}$ given $\boldsymbol{\beta}^*$. The call sequence of the `psvglm.fit()` function is of the form

```
psvglm.fit <- function (x, y, w, X.vlm.arg, Xm2, Ym2, family,
                        control = vglm.control(),
                        criterion = "coefficients",
                        qr.arg, constraints, extra, Terms,
                        smooth.labels, ps.list = ps.list, ...)
```

The arguments for `psvglm.fit()` are defined in the same way as those of `vglm.fit()` and we have added the argument called `ps.list`, which contains the components corresponding to P-spline smoothers such as the number of knots to be used for basis construction, a vector of smoothing parameters, a set of penalty matrices for the smooth terms.

Arguments

`ps.list` a list containing the components corresponding to P-spline smoothers such as a vector of the number of knots, the degree of B-splines, order of the penalty, non-negative regularization parameters and small positive numbers to stabilize linear dependencies among B-spline bases, and a set of penalty matrices for the smooth terms.

The function returns a list of fit information.

A.2.4 The `psvlm.wfit()` function

The `psvlm.wfit()` is an internal function of the `psvglm.fit()` function. This is a modification of the `vlm.wfit()` function. The function works by constructing the augmented adjusted dependent variable ($\boldsymbol{z}'$), regressors ($\mathbf{X}'_{\text{VAM}}$) and weights ($\mathbf{W}'$) (cf. equation (4.31)). Then, the necessary quantities required for the `magic()` function such as the model matrix, the array of smoothing parameters, a list of penalty matrices in the form of `S[[k]]`, where `k` indicates the kth penalty matrix and an array indicating the element `1,1` of `S[[k]]`, are obtained using the `Xps2Xmagic()` function. The P-IRLS algorithm consisting of iterating the two steps to convergence: (i) given $\boldsymbol{\lambda}$, the estimates of $\boldsymbol{\beta}^*$ is obtained by solving a generalized least squares problem using the data augmentation on the adjusted dependent variable, regressors, and weights (using the `lm.fit()` function), and (ii) given $\boldsymbol{\beta}^*$, $\boldsymbol{\lambda}$ is estimated by minimizing the UBRE score using the `magic()` function. The call sequence of the `psvlm.wfit()` function is of the form

```
psvlm.wfit <- function (xmat , zmat , Blist , wz , U, pen.vlm ,
                        B.list , ps.list , ...)
```

The arguments for `psvlm.wfit()` are defined in the same way as those of `vlm.wfit()`. We have added new arguments, namely, `pen.vlm`, `B.list` and `ps.list` which are defined as the following:

Arguments

`pen.vlm` a list containing contains the components corresponding to the penalty of P-spline VGAMs such as the diagonal blocks, $\mathbf{P}^*_{\lambda 1}, \ldots, \mathbf{P}^*_{\lambda p}$, and $\mathbf{P}^*_{\lambda}$.

`B.list` a list containing the vector $\mathbf{0}_{\vartheta}$ and the $\vartheta \times \vartheta$ identity matrix $\mathbf{I}_{\vartheta}$ used for the data augmentation (cf. equation (4.31)).

`ps.list` same as for `psvglm.fit`.

The function returns a list of fit information.

A.2.5 The `Pen.vps()` function

This is an internal function of `psvglm.fit()`. This function is used to construct the penalty matrix $\mathbf{P}^*_{\lambda}$ for P-spline VGAMs. The `Pen.vps()` function works by first obtaining the components such as constraint matrices and the components corresponding to the P-spline smoothers. The diagonal blocks, $\mathbf{P}^*_{\lambda 1}, \ldots, \mathbf{P}^*_{\lambda p}$, of $\mathbf{P}^*_{\lambda}$ is constructed (cf. equation (4.46)). A vector containing a set of the smoothing parameters for each smooth component function in $\boldsymbol{f}_k(x_k) = \left(f_{(1)k}(x_k), \ldots, f_{(\mathrm{R}_k)k}(x_k)\right)^T$ is constructed. The smoothing parameter vector is then given to an associated diagonal block. The penalty matrix $\mathbf{P}^*_{\lambda}$ is constructed. The diagonal blocks $\mathbf{P}^*_{\lambda 1}, \ldots, \mathbf{P}^*_{\lambda p}$ are constructed in the order that the smooth terms appear in the model formula.

If a parametric term such as linear terms or factors is included as the kth term in the model formula, then the $\mathbf{P}^*_{\lambda k}$ term for a strictly parametric term is set to a zero matrix as such terms are not penalized. The call sequence of the `Pen.vps()` function is of the form

```
Pen.vps <- function(constraints, ps.list)
```

Arguments

`constraints` same as for `psvglm.fit`.

`ps.list` same as for `psvglm.fit`.

The function returns a penalty matrix for P-spline VGAMs with attributes that are used by `psvglm.fit`.

A.2.6 The `Xps2Xmagic()` function

This is an internal function of `psvlm.wfit()`. This function is used to construct the necessary quantities required for the `magic()` function in `mgcv`. The `Xps2Xmagic()` function works by obtaining the components such as constraint matrices, the model matrix $\mathbf{X}_{\text{VAM}}$, the diagonal blocks, $\mathbf{P}^*_{\lambda 1}, \ldots, \mathbf{P}^*_{\lambda p}$, of $\mathbf{P}^*_{\lambda}$ and a list containing the components corresponding to the P-spline smoothers. The major inputs required for the `magic()` function such as the array of smoothing parameters, a list of penalty matrices in the form of `S[[k]]`, where `k` indicates the kth penalty matrix and an array indicating the element `1,1` of `S[[k]]`, are then constructed. The call sequence of the `Xps2Xmagic()` function is of the form

```
Xps2Xmagic <- function(xmat, constraints, ps.list, pen.vlm)
```

Arguments

`xmat`	the model matrix for P-spline VGAMs.
`constraints`	same as for `psvglm.fit`.
`ps.list`	same as for `psvglm.fit`.
`pen.vlm`	same as for `psvlm.wfit`.

The function returns a list of the major components required for the `magic()` function.

A.3 Extractor functions

We have provided extractor functions for the “hat values” and the “effective degrees of freedom”, of a fitted P-spline VGAM object. These extractor functions are called `hatvaluespsvlm()` and `edfpsvlm()` respectively.

`hatvaluespsvlm()` function

`hatvaluespsvlm()` is a modification of `hatvaluesvlm()`. This suite of functions can be used to compute some of the regression (leave-one-out deletion) diagnostics for the P-spline VGAM class. The invocation `hatvaluespsvlm(psvgamObject)` returns a $n \times M$ matrix of the diagonal elements of the hat (projection) matrix of a `psvgam` object, computed from, a weighted, penalized least squares problem instead of a weighted least squares problem. Following Yee (2015a), the QR decomposition of the object is reconstructed, and then straightforward calculations are performed.

`edfpsvlm()` function

`edfpsvlm` extracts the effective degrees of freedom (EDF) associated with each penalty in a P-spline VGAM fit. This function returns a vector of EDFs, named with labels identifying which penalty each EDF relates to.

Bibliography

RS Anderssen and Peter Bloomfield. A time series approach to numerical differentiation. *Technometrics*, 16(1):69–75, 1974.

Ben G Armstrong and Margaret Sloan. Ordinal regression models for epidemiologic data. *American Journal of Epidemiology*, 129(1):191–204, 1989.

JR Ashford and RR Sowden. Multi-variate probit analysis. *Biometrics*, pages 535–546, 1970.

D Bell, J Walker, G O'Connor, J Orrel, and R Tibshirani. Spinal deformation following multilevel thoracic and lumbar laminectomy in children. *Submitted for publication*, 1989.

Ralf Bender and Ulrich Grouven. Using binary logistic regression models for ordinal data with non-proportional odds. *Journal of clinical epidemiology*, 51(10):809–816, 1998.

Anne-Laure Boulesteix. Maximally selected chi-square statistics for ordinal variables. *Biometrical Journal*, 48(3):451–462, 2006.

Norman E Breslow and David G Clayton. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88(421):9–25, 1993.

John M Chambers and Trevor J Hastie. *Statisticals models in S*. Chapman & Hall, London, 1993.

Timothy J Cole and Pamela J Green. Smoothing reference centile curves: the LMS method and penalized likelihood. *Statistics in medicine*, 11(10):1305–1319, 1992.

TJ Cole. Fitting smoothed centile curves to reference data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, pages 385–418, 1988.

Carl De Boor. A practical guide to splines. *Mathematics of Computation*, 1978.

Paul Dierckx. *Curve and surface fitting with splines*. Oxford University Press, 1995.

Francesco Donat and Giampiero Marra. Semi-parametric bivariate polychotomous ordinal regression. *Statistics and Computing*, pages 1–17, 2015.

Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.

Paul HC Eilers and Brian D Marx. Flexible smoothing with B-splines and penalties. *Statistical science*, pages 89–102, 1996.

PHC Eilers. Discussion on âthe analysis of designed experiments and longitudinal data by using smoothing splinesâ(by ap verbyla, br cullis, mg kenward and sj welham). *Appl. Statist*, 48: 307–308, 1999.

Jeffrey Fessler et al. Nonparametric fixed-interval smoothing with vector splines. *Signal Processing, IEEE Transactions on*, 39(4):852–859, 1991.

David A Freedman and Jasjeet S Sekhon. Endogeneity in probit response models. *Political Analysis*, 18(2):138–150, 2010.

Alan Genz, Frank Bretz, Tetsuhisa Miwa, Xuefei Mi, Friedrich Leisch, Fabian Scheipl, and Torsten Hothorn. mvtnorm: Multivariate normal and t distributions. *R package version 0.9-2, URL http://CRAN. R-project. org/package= mvtnorm*, 2008.

Charles J Geyer. Trust region optimization. 2014.

Philip E Gill, Walter Murray, and Margaret H Wright. *Practical optimization*, volume 5. Academic press London, 1981.

Peter J Green and Bernard W Silverman. *Nonparametric regression and generalized linear models: a roughness penalty approach*. CRC Press, 1993.

William H Greene. *Econometric analysis*. New York: Prentice Hall, 2007.

Chong Gu. Cross-validating non-Gaussian data. *Journal of Computational and Graphical Statistics*, 1(2):169–179, 1992.

Chong Gu. *Smoothing spline ANOVA models*. Springer, 2002.

Chong Gu and Grace Wahba. Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method. *SIAM Journal on Scientific and Statistical Computing*, 12(2):383–398, 1991.

Trevor J Hastie. Pseudosplines. *Journal of the Royal Statistical Society: Series B (Methodological)*, pages 379–396, 1996.

Trevor J Hastie and Robert J Tibshirani. Generalized additive models. *Statistical science*, pages 297–310, 1986.

Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*. Chapman and Hall, London, 1990.

Oliver Linton and Jens Perch Nielsen. A kernel method of estimating structured nonparametric regression based on marginal integration. *Biometrika*, pages 93–100, 1995.

S MacMahon, R Norton, R Jackson, MJ Mackie, A Cheng, S Vander Hoorn, A Milne, and A McCulloch. Fletcher challenge-university of auckland heart & health study: design and baseline findings. *The New Zealand medical journal*, 108(1013):499–502, 1995.

Giampiero Marra. On p-values for semiparametric bivariate probit models. *Statistical Methodology*, 10(1):23–28, 2013.

Giampiero Marra and R Radice. Semiparbivprobit: semiparametric bivariate probit modelling. *R package version*, 3:2–9, 2013.

Giampiero Marra and Rosalba Radice. Penalised regression splines: theory and application to medical research. *Statistical Methods in Medical Research*, 19(2):107–125, 2010.

Giampiero Marra and Rosalba Radice. Estimation of a semiparametric recursive bivariate probit model in the presence of endogeneity. *Canadian Journal of Statistics*, 39(2):259–279, 2011.

Giampiero Marra and Simon N Wood. Coverage properties of confidence intervals for generalized additive model components. *Scandinavian Journal of Statistics*, 39(1):53–74, 2012.

Giampiero Marra, Georgios Papageorgiou, and Rosalba Radice. Estimation of a semiparametric recursive bivariate probit model with nonparametric mixing. *Australian & New Zealand Journal of Statistics*, 55(3):321–342, 2013a.

Giampiero Marra, Rosalba Radice, et al. A penalized likelihood estimation approach to semiparametric sample selection binary response modeling. *Electronic Journal of Statistics*, 7: 1432–1455, 2013b.

Brian D Marx and Paul HC Eilers. Direct generalized additive modeling with penalized likelihood. *Computational Statistics & Data Analysis*, 28(2):193–209, 1998.

Peter McCullagh. Regression models for ordinal data. *Journal of the royal statistical society: Series B (Methodological)*, pages 109–142, 1980.

Peter McCullagh and John A Nelder. *Generalized linear models.* London: Chapman & Hall, second edition, 1989.

JA Nelder and RWM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, pages 370–384, 1972.

Marc Nerlove and S James Press. *Univariate and multivariate log-linear and logistic models*, volume 1306. Rand Corporation Santa Monica, Calif, 1973.

Jorge Nocedal and Stephen J Wright. Numerical optimization. *Springerverlang, USA*, 1999.

Douglas Nychka. Bayesian confidence intervals for smoothing splines. *Journal of the American Statistical Association*, 83(404):1134–1143, 1988.

Finbarr O'Sullivan. A statistical perspective on ill-posed inverse problems. *Statistical science*, pages 502–518, 1986.

Juni Palmgren. *Regression models for bivariate binary responses*, 1989.

Roger D Peng and Leah J Welty. The nmmapsdata package. *R news*, 4(2):10–14, 2004.

Rosalba Radice, Giampiero Marra, and Małgorzata Wojtyś. Copula regression spline models for binary outcomes. *Statistics and Computing*, pages 1–15, 2015.

Christian H Reinsch. Smoothing by spline functions. *Numerische mathematik*, 10(3):177–183, 1967.

David Ruppert, Matt P Wand, and Raymond J Carroll. *Semiparametric regression*. Number 12. Cambridge university press, 2003.

George A. F. Seber and C. J. Wild. *Nonlinear Regression*. Wiley-Interscience, 9 2003. ISBN 9780471471356.

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

Gerhard Tutz and Harald Binder. Generalized additive modeling with implicit variable selection by likelihood-based boosting. *Biometrics*, 62(4):961–971, 2006.

William N Venables and Brian D Ripley. *Modern applied statistics with S*. Springer, New York, USA, fourth edition, 2002.

Grace Wahba. Bayesian" confidence intervals" for the cross-validated smoothing spline. *Journal of the Royal Statistical Society: Series B (Methodological)*, pages 133–150, 1983.

Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

Grace Wahba and P Craven. Smoothing noisy data with spline functions. estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:377–404, 1978.

Grace Wahba, Yuedong Wang, Chong Gu, Ronald Klein, Barbara Klein, et al. Smoothing spline ANOVA for exponential families, with application to the Wisconsin Epidemiological Study of Diabetic Retinopathy: the 1994 neyman memorial lecture. *The Annals of Statistics*, 23(6): 1865–1895, 1995.

Matt P Wand. A comparison of regression spline smoothing procedures. *Computational Statistics*, 15(4):443–462, 2000.

Matt P Wand. Smoothing and mixed models. *Computational statistics*, 18(2):223–249, 2003.

MP Wand and JT Ormerod. On semiparametric regression with o'sullivan penalized splines. *Australian & New Zealand Journal of Statistics*, 50(2):179–198, 2008.

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

Simon N Wood. Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):95–114, 2003.

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

Simon N Wood. On confidence intervals for generalized additive models based on penalized regression splines. *Australian & New Zealand Journal of Statistics*, 48(4):445–464, 2006a.

Simon N Wood. *Generalized additive models: An introduction with R*. Chapman & Hall/CRC, Boca Raton, FL, USA, 2006b.

Simon N Wood. The mgcv package. *www. r-project. org*, 2007.

Simon N Wood. Fast stable direct fitting and smoothness selection for generalized additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3): 495–518, 2008.

Simon N Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(1):3–36, 2011.

Simon N Wood. P-splines with derivative based penalties and tensor product smoothing of unevenly distributed data. *Statistics and Computing*, pages 1–5, 2016.

Eileen M Wright and Patrick Royston. A comparison of statistical methods for age-related reference intervals. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 160(1):47–69, 1997.

Thomas W Yee. An implementation for regression quantile estimation. In *Compstat*, pages 3–14. Springer, 2002.

Thomas W Yee. Quantile regression via vector generalized additive models. *Statistics in Medicine*, 23:2295–2315, 2004.

Thomas W Yee. The VGAM package. *R News*, 8:28–39, October 2008. URL http://CRAN.R-project.org/doc/Rnews/.

Thomas W Yee. The VGAM package for categorical data analysis. *Journal of Statistical Software*, 32:1–34, 2010. URL http://www.jstatsoft.org/v32/i10/.

Thomas W Yee. Package VGAM. 2015a.

Thomas W Yee. *Vector Generalized Linear and Additive Models: With an Implementation in R*. Springer, New York, NY, USA, 2015b.

Thomas W Yee and Trevor J Hastie. Reduced-rank vector generalized linear models. *Statistical Modelling*, 3:15–41, 2003.

Thomas W Yee and Monique Mackenzie. Vector generalized additive models in plant ecology. *Ecological Modelling*, 157(2):141–156, 2002.

Thomas W Yee and Chris J Wild. Vector splines and the vector additive model. Technical Report STAT04, Department of Statistics, University of Auckland, Auckland, 1994.

Thomas W Yee and Chris J Wild. Vector generalized additive models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58:481–493, 1996.