Model Selection in Generalised Structured Additive Regression Models

Christiane Belitz

München 2007

Model Selection in Generalised Structured Additive Regression Models

Christiane Belitz

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Christiane Belitz
aus München

München, den 27.8.2007

Erstgutachter: Prof. Dr. Stefan Lang
Zweitgutachter: Prof. Dr. Ludwig Fahrmeir
Rigorosum: 12.11.2007

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Institut für Statistik der Ludwig–Maximilians–Universität München. Sie wurde zum Teil durch Mittel des Sonderforschungsbereiches 386 "Statistische Analyse diskreter Strukturen" gefördert.

Mein Dank gilt in erster Linie meinen beiden Betreuern Stefan Lang und Ludwig Fahrmeir, die mich wesentlich bei der Fertigstellung der Arbeit unterstützt haben und immer ein offenes Ohr für fachliche Probleme hatten. Insbesondere die intensive Zusammenarbeit und Betreuung durch meinen Doktorvater Stefan Lang während des gesamten Promotionszeitraumes hat die Dissertation maßgeblich und im positiven Sinne beeinflusst. Desweiteren war es mir eine Freude im Team von Ludwig Fahrmeir mitzuarbeiten, der mir durch die Bereitstellung der Stelle am Institut die Promotion erst ermöglicht hat.

Allen Mitarbeitern und Mitarbeiterinnen des Instituts für Statistik möchte ich für die sehr angenehme und freundschaftliche Atmosphäre danken. Dabei gebührt mein besonderer Dank meinen Freunden und Kollegen Rüdiger Krause, Andrea Hennerfeind, Andreas Brezger, Thomas Kneib, Susanne Heim, Michael Höhle, Florian Leitenstorfer, Manuela Hummel und Ursula Gerhardinger für die moralische Unterstützung sowie die vielen wertvollen fachlichen Diskussionen.

München, Dezember 2007 *Christiane Belitz*

Zusammenfassung

In den vergangenen Jahren hat die Komplexität von Datensätzen immer weiter zugenommen, wodurch flexiblere Analyseverfahren erforderlich wurden. Ein solches flexibles Verfahren ist die Regressionsanalyse basierend auf einem strukturiert additiven Prädiktor. Dieser ermöglicht eine geeignete Modellierung von unterschiedlichen Informationsarten, z.B. mittels glatter Funktionen für räumliche Information, nichtlinearer Funktionen für stetige Kovariablen oder mittels Effekten für die Modellierung gruppenspezifischer Heterogenität. In dieser Arbeit geben wir einen Überblick über viele wichtige Funktionen. Außerdem setzen wir einen Schwerpunkt auf Interaktionseffekte und führen eine Möglichkeit zur einfachen Modellierung einer komplexen Interaktion zweier stetiger Kovariablen ein.
Ein zentraler Aspekt dieser Arbeit ist das Thema der Variablenselektion und Glättungsparameterbestimmung in strukturiert additiven Regressionsmodellen. Zu diesem Zweck führen wir einen effizienten Algorithmus ein, der gleichzeitig relevante Kovariablen auswählt sowie den Glattheitsgrad ihrer Effekte bestimmt. Mit diesem Algorithmus ist es sogar möglich, komplexe Situationen mit vielen Kovariablen und Beobachtungen zu bewältigen. Dabei basiert die Bewertung von verschiedenen Modellen auf Gütekriterien wie z.B. dem AIC, BIC oder GCV. Die methodische Entwicklung wurde stark durch Fallstudien aus unterschiedlichen Bereichen motiviert. Als Beispiele analysieren wir zwei verschiedene Datensätze bezüglich der Einflussfaktoren auf Unterernährung in Indien sowie auf die Tarifberechnung von Versicherungen. Außerdem untersuchen wir das Verhalten unseres Selektionsalgorithmus anhand mehrerer ausführlicher Simulationsstudien.

Abstract

In recent years data sets have become increasingly more complex requiring more flexible instruments for their analysis. Such a flexible instrument is regression analysis based on a structured additive predictor which allows an appropriate modelling for different types of information, e.g. by using smooth functions for spatial information, nonlinear functions for continuous covariates or by using effects for the modelling of cluster–specific heterogeneity. In this thesis, we review many important effects. Moreover, we place an emphasis on interaction terms and introduce a possibility for the simple modelling of a complex interaction between two continuous covariates.
Mainly, this thesis is concerned with the topic of variable and smoothing parameter selection within structured additive regression models. For this purpose, we introduce an efficient algorithm that simultaneously selects relevant covariates and the degree of smoothness for their effects. This algorithm is even capable of handling complex situations with many covariates and observations. Thereby, the validation of different models is based on goodness of fit criteria, like e.g. AIC, BIC or GCV. The methodological development was strongly motivated by case studies from different areas. As examples, we analyse two different data sets regarding determinants of undernutrition in India and of rate making for insurance companies. Furthermore, we examine the performance or our selection approach in several extensive simulation studies.

Contents

Chapter 1

Introduction

The issues addressed in this thesis arise in the course of practical applications in many different areas like e.g. marketing, insurance, development economics, ecology and many more. The introduction will explain the central issues on the basis of an example from an insurance company and give an outline of the thesis.
The example confronts us with the following problem: during one year, a Belgian insurance company selling car insurance policies gets claim reports from some of their policyholders together with the costs which have arisen by these claims. Additionally, the company has certain information about their policyholders: gender, age, address, type and age of the car, etc. Based on this data our objective is to calculate (at least relatively) fair premiums: Policyholders who produce high costs for the company due to many and/or expensive claims are supposed to pay higher fees than the rest. Hence, we need to detect characteristics of policyholders who produce high costs and characteristics of policyholders with low costs. Therefore, the relation between each variable of interest, i.e. number and costs of claims, and the influencing variables, i.e. characteristics of the policyholders, has to be analysed. With a simple descriptive analysis, it is possible to study the relation between the response variable and one (or possibly two) independent variables at a time. Figure 1.1, for instance, shows average response values for the Belgian districts and, separately for men and women, average response values for the grouped policyholder's age. Both response variables vary over the Belgian districts: the highest average logarithmic claim size is observed in the extreme south of Belgium whereas the same area has the lowest average claim frequencies. High average claim frequencies can be observed in the area around Brussels. The policyholder's age also shows variation in both response variables: the average logarithmic claim size is especially high for young and old drivers whereas the average claim frequency decreases with age. With policyholder's age, the average values of both response variables differ between the sexes but show a similar trend for each sex.

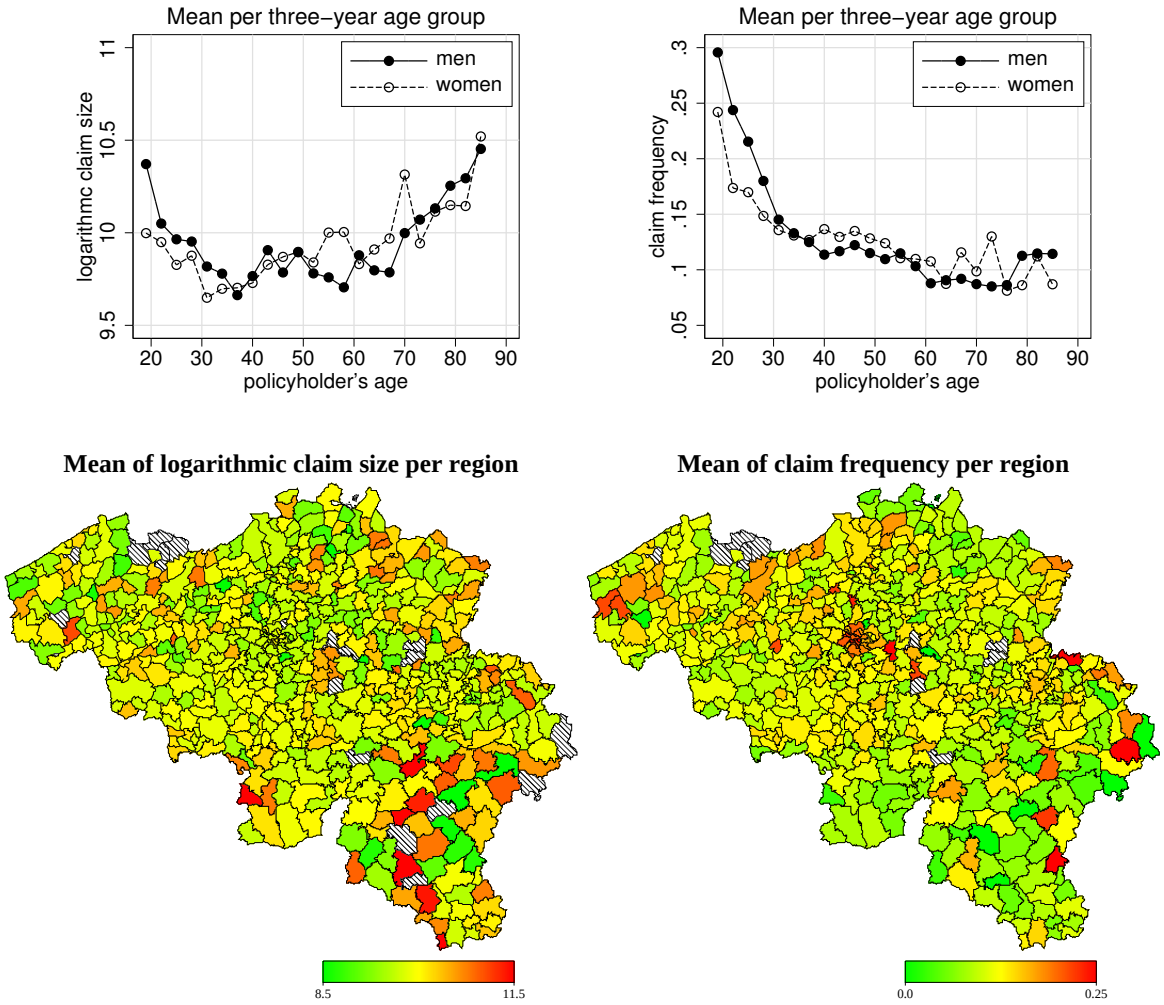

Figure 1.1: Average response variables logarithmic claim size (left column) and claim frequency (right column) each calculated over three successive years of age separately for men and women (upper row) and over the Belgian districts (bottom row).

Instead of considering the effect of only one influencing factor at a time as in figure 1.1, our objective is to obtain a model for each response variable which considers all influencing variables simultaneously. An adequate instrument is a special, very flexible form of regression analysis which is explained in **chapter 2** and which assumes the following relation between the expectation of the response variable y (either logarithmic claim size or claim frequency in the example) and influencing variables $x_1, \ldots, x_q$:

$$E(y|x_1, \ldots, x_q) = h(\gamma_0 + f_1(x_1) + \ldots + f_q(x_q)),$$

where the functions f_j and the parameter γ_0 are estimated from the data at hand but the so called response function h is fixed. The choice for each function f_j depends on the type of

variable x_j and on assumptions about the function's smoothness. For many different types of covariates, chapter 2 describes functions which adequately model the respective effect. For example, it is possible to estimate a smooth spatial effect for the districts of Belgium which assumes that neighbouring regions behave similar (which is assumed due to a similar traffic density or similar socio–demographic factors). The effect of the policyholder's age can be modelled by a smooth nonlinear function because figure 1.1 indicates a nonlinear relationship between age and each of the two response variables. Nonlinear functions for continuous covariates can even deal with effects whose functional form is unknown. It is also possible to estimate two separate nonlinear age effects for men and women. Moreover, we introduce a special kind of function (which we call ANOVA type decomposition) for the simple modelling of a complex nonlinear interaction effect between two continuous covariates.

The choice of the response function h depends on the distribution assumed for the response y and is chosen such that the estimated expectations lie in the correct domain. For many frequently used distributions, possible choices for h are given in the second chapter. Moreover, we describe how the estimation of functions and regression parameters is performed. In the second chapter we assume that all influencing variables $x_1, \ldots, x_q$ which are used in the regression model have an influence on the response y. In **chapter 3** we want to dismiss this assumption out of the following reasons: the assumption implies, that before we estimate the regression model we have to carefully choose the covariates entering the model from all available variables. Thereby, the goal is to consider all important factors but to limit the size of the model. For this selection by hand, a descriptive analysis like in figure 1.1 can provide useful clues. However, descriptive plots often do not clearly show whether certain covariates are actually important. For instance, the policyholder's age clearly has an effect on both response variables. But based on figure 1.1 one cannot definitely decide, whether an interaction between age and gender is necessary for the logarithmic claim size. Moreover, only one variable (or one interaction) at a time can be examined. Hence, the variation visible in a descriptive plot could also be due to other more important covariates whose behaviour differs over the range of the examined variable. The result of such dependencies may be that the less important covariate loses its influence on the response if all covariates are considered in a common model. For instance, regional differences as visible in figure 1.1 are probably to a large extent due to differences in traffic density and allowed speed: In urban areas there is high density of traffic at low allowed speed while this is the opposite in rural areas. Hence, if the two factors traffic density and allowed speed were available (what is not the case) and included in the regression model, the spatial effect may vanish. Hence, after a descriptive analysis, we do not definitely know which covariates or terms should be included in the model.

Furthermore, nonlinear functions f_j include an additional parameter which governs the

smoothness of the respective function. The methods in chapter 2 can only deal with a fixed smoothing parameter so that the degree of smoothness must be known beforehand. For the spatial function, this implies that we know how similar neighbouring regions actually are: completely alike, sharing some common characteristics or completely different? However, these facts are usually unknown.
In conclusion, when analysing a data set, we have to deal with some or all of the following questions:

- Which terms (covariates) are to be included in the model?
- Which degree of smoothness is appropriate for a nonlinear function?
- Does a nonlinear effect vary over the range of another variable?
- Is there a complex interaction between two continuous variables?
- Does the data contain spatial heterogeneity?
- Does the data contain heterogeneity between groups or clusters?

These questions are addressed in the third chapter. We introduce selection algorithms that are designed to answer these questions by automatically selecting a good model from a large set of possible models. Thereby, the evaluation of competing models is based on goodness of fit criteria. An emphasis is placed on the practicability of the selection algorithms even for complex models with many available covariates.
Consider our starting example again: For the logarithmic claim size (*logs*) the question has arisen if an interaction term between the policyholder's age (*ageph*) and gender (s) is necessary. Hence, we specify the largest possible model by

$$logs = \gamma_0 + f_1(ageph) + g_1(ageph) \cdot s + f_{spat}(dist) + g_{spat}(dist) \cdot s + \gamma_s\, s + \ldots + \varepsilon,$$

where the effect of the policyholder's age and the spatial effect over the Belgian districts (*dist*) may vary between the sexes. Our automatic selection algorithm chooses the model

$$logs = \gamma_0 + f_1(ageph) + g_1(ageph) \cdot s + f_{spat}(dist) + \gamma_s\, s + \ldots + \varepsilon,$$

where only the interaction effect of the policyholder's age and gender is selected but not the interaction between the spatial effect and gender.
Chapter 4 extends the contents of the preceding chapters to the special case of multinomial logit models. Here, the response variable is categorical and can have more than two possible outcomes. Hence, this chapter deals with a special kind of multivariate response, in contrast to chapters 2 and 3 which deal with univariate response variables.
Chapter 5 addresses the subject of credible intervals for regression parameters and nonlinear functions. Confidence bands of nonlinear functions are an important optical tool that

help to detect areas of the function with a larger uncertainty. Moreover, we consider the issue of model selection uncertainty: The selected model depends on the available data and would probably be different for a new data sample. Hence, we are interested to examine the stability of the selected model.
We implemented the selection algorithms described in chapters 2–5 in the programming language C++ within the software package *BayesX*. *BayesX* is available free of charge via internet from

`http://www.stat.uni-muenchen.de/~bayesx`

Chapter 6 explains how a data analysis based on this methodology can be performed using *BayesX*.
We tested our selection algorithm in excessive simulation studies and compared it to competing approaches. The results are presented in **chapter 7**.
In **chapter 8** we analyse two real data sets using the methodology of chapters 2–5. First (in section 8.1) we continue the car insurance application and select a model both for the logarithmic claim size and for the number of claims. Thereby, a focus is placed on interaction effects with regard to the policyholder's gender. Additionally, for each response variable we use the methodology from chapter 5 to examine model selection uncertainty, i.e. the stability of the selected model.
The second application described in section 8.2 examines child undernutrition in India. Here, the response variable is the nutritional condition of a child compared to the average nutritional status of children from a well–nourished reference population. We analyse chronic undernutrition which is indicated by an insufficient height for age also called stunting. Again, we focus on interaction effects with regard to the children's gender.
The **appendix** refers to selected topics of chapters 2 and 3 and explains these topics in greater detail.
Finally, we want to mention that based on the methodology from chapter 3 we published two papers in proceedings volumes: Steiner, Belitz & Lang (2006) and Belitz & Lang (2007).

Chapter 2

Univariate Structured Additive Regression Models

This chapter gives an introduction to regression models based on a structured additive predictor (STAR models). These regression models are very general and can deal with different types of dependent variables and also with different kinds of covariates. In the first section 2.1 of this chapter, we give a short introduction in regression models including the generalisation to STAR models. How to adequately approximate different covariate effects is the subject of section 2.2. The last section 2.3 deals with parameter estimation in the class of STAR models.

2.1 Introduction

The objective of regression analysis is to measure the influence of some variables x_j, $j = 1, \dots q$, the so–called covariates, on a further variable y called response or independent variable. The model most widely used is the classical linear model. This model requires a Gaussian distributed (or under less strict assumptions at least continuous) response variable. The relation between the conditional mean of the response and the covariates is assumed to be

$$\mathrm{E}(y|x_1, \dots, x_q) = \gamma_0 + \gamma_1 x_1 + \dots + \gamma_q x_q = \gamma' x =: \eta. \tag{2.1}$$

Through their value and algebraic sign, the regression coefficients $\gamma_1, \dots, \gamma_q$ determine the direction and the strength of influence of their respective covariate. The parameter γ_0 is called constant term or intercept. Parameter η is referred to as linear predictor because formula (2.1) is linear in the regression coefficients and the relation between covariates and

expectation is also linear.
If the response variable is no longer Gaussian distributed but belongs to an univariate exponential family, the generalised linear model can be used. Here, it is assumed that the linear predictor η and the conditional expectation are linked through a response function h, i.e.

$$\mathrm{E}(y|x_1,\ldots,x_q) = h(\gamma_0 + \gamma_1 x_1 + \ldots + \gamma_q x_q) = h(\gamma' x) = h(\eta). \tag{2.2}$$

Usually, function h is chosen such that the values of η are transformed to the domain of the expected value. For Gaussian distributed responses, the expectation can adopt all real values. Hence, a transformation is not necessary and the identity function can be chosen for h, i.e. $h = id$. Examples for non–Gaussian response variables and appropriate choices for function h are given in section 2.3.3 of this chapter. In a similar way it is also possible to deal with multicategorical response variables, see chapter 4.
In this thesis, we replace the linear predictor

$$\eta := \gamma_0 + \gamma_1 x_1 + \ldots + \gamma_q x_q = \gamma' x \tag{2.3}$$

by a semiparametric structured additive predictor (compare Fahrmeir, Kneib & Lang (2004)) of the form

$$\eta := \gamma_0 + f_1(x_1) + \ldots + f_q(x_q) + \gamma_1 u_1 + \ldots + \gamma_p u_p = f_1(x_1) + \ldots + f_q(x_q) + \gamma' u. \tag{2.4}$$

The reason for using a semiparametric predictor lies in the strong assumptions made by the linear predictor. The linear predictor assumes: (i) a linear influence of the covariates on the predictor or even on the response in the Gaussian case; (ii) independence of the observations. In many situations, however, the assumptions are not adequate and we are confronted with one or more of the following problems:

- The effect of some of the continuous covariates may be of a (unknown) nonlinear form.
- The observations can be spatially correlated.
- The observations can be temporally correlated.
- There can be unobserved heterogeneity among individuals or units that is not accounted for by the available covariates.
- There may be a complex interaction between two continuous variables.

The structured additive predictor (2.4) overcomes the difficulties by replacing the linear effects $\gamma_j x_j$ by functions $f_j(x_j)$. The functions f_j can be of different type according to

the different types possible for the covariates x_j. For instance, the predictor is able to model nonlinear effects of continuous variables or time scales and it can deal with spatial or unit–specific information. The estimation of complex interactions between two covariates is also possible. Possibilities for appropriate functions f_j will be given in section 2.2 of this chapter. The predictor can be semiparametric, i.e. include a parametric part like $\gamma' u$ in formula (2.4), so that some covariates, especially categorical variables, can still be modelled by linear effects. Note that covariates which are modelled linearly are denoted by u_j in order to distinguish them from other covariates. The parametric part $\gamma' u$ also contains the intercept term γ_0.
Structured additive regression models cover a wide range of different models. Some special cases that are well known in the literature are: additive and generalised additive models (Hastie & Tibshirani (1990), Rigby & Stasinopoulos (2005) or Wood (2006a)), generalised additive mixed models (Ruppert, Wand & Carroll (2003)), geoadditve models (Fahrmeir & Lang (2001a) or Kammann & Wand (2003)), varying coefficient models (Hastie & Tibshirani (1993)), geographically weighted regression (Fotheringham, Brunsdon & Charlton (2002)) and ANOVA type interaction models (Chen (1993)).

2.2 Model components

As already mentioned in the last section, we deal with different kinds of independent variables in the context of STAR models. For every type of covariate, there exist one or more possibilities to construct a function which adequately represents the available information. These possibilities with their specific features are described in this section.
It turns out that all nonlinear functions described in this section can be written in a general form. This allows an equal treatment of all nonlinear functions when estimating regression coefficients and selecting relevant covariates (compare chapter 3 for this topic). That means, for inference and selection algorithms we only need to distinguish between two cases: linear effects and nonlinear functions.
In this thesis we follow mainly a frequentist approach based on a penalised likelihood. Since some of the nonlinear functions originally were derived under a Bayesian point of view, we discuss also Bayesian interpretations and the equivalence between penalised likelihood and empirical Bayesian estimation.
The common features of all nonlinear functions $f(x)$ are listed below:

- First of all, the vector of function evaluations $\mathbf{f} = (f_1, \ldots, f_n)'$ for n observations can be written as a linear combination of a $n \times p$ design matrix $\mathbf{X}$ and a vector of

regression coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$, i.e.

$$\mathbf{f} = \mathbf{X}\boldsymbol{\beta}. \tag{2.5}$$

That means, all functions f are linear in their regression coefficients. Because of the additive structure of the predictor this property still holds for the entire predictor even if the predictor contains several functions.

- In a Bayesian framework each function f is provided with a prior distribution. The prior distribution depends on the type of the respective covariate x and on assumptions about the smoothness of the function f. This leads to different priors for the different types of functions which are described in the following sections in detail. Generally, the prior assumptions about f can be expressed by applying a prior distribution to the regression coefficients $\boldsymbol{\beta}$. The distribution is either a proper or improper Gaussian distribution of the form

$$p(\boldsymbol{\beta}) \propto \exp\left(-\frac{1}{2\tau^2}\boldsymbol{\beta}'\mathbf{P}\boldsymbol{\beta}\right), \tag{2.6}$$

with a variance parameter τ^2 and a precision matrix $\mathbf{P}$. The prior distributions of different function types are characterised by their individual precision matrix which contains information about the function type and assumptions about the smoothness of the function. If matrix $\mathbf{P}$ is rank–deficient the prior distribution is improper, otherwise it is proper.
There is a close relationship between the Bayesian and the penalised likelihood approach: Suppose, the predictor only contains function f, i.e. $\eta = \mathbf{f} = \mathbf{X}\boldsymbol{\beta}$. In this case the likelihood function $L(y|\boldsymbol{\beta})$ and the log–likelihood function $l(y|\boldsymbol{\beta})$ only contain the parameter vector $\boldsymbol{\beta}$ and no other parameters. Then, the posterior distribution $p(\boldsymbol{\beta}|y)$ with response vector $y = (y_1, \ldots, y_n)$ is given by

$$p(\boldsymbol{\beta}|y) \propto L(y|\boldsymbol{\beta}) \cdot p(\boldsymbol{\beta}). \tag{2.7}$$

The mode of this distribution may be calculated from the logarithmic posterior distribution

$$\begin{aligned} \log(p(\boldsymbol{\beta}|y)) &\propto l(y|\boldsymbol{\beta}) + \log(p(\boldsymbol{\beta})) \\ &\propto l(y|\boldsymbol{\beta}) - \frac{1}{2\tau^2}\boldsymbol{\beta}'\mathbf{P}\boldsymbol{\beta}. \end{aligned} \tag{2.8}$$

Formula (2.8) is equivalent to a penalised log–likelihood where the precision matrix $\mathbf{P}$ is used as penalty matrix. Hence, the penalised maximum likelihood estimate and the mode of the posterior distribution are identical. The logarithmic kernel of the

prior $p(\boldsymbol{\beta})$ corresponds to the penalty term of the penalised log–likelihood.
In the context of penalised likelihood, instead of variance parameter τ^2 usually a smoothing parameter is used to control the smoothness of the function. This smoothing parameter is defined as $\lambda := \phi/\tau^2$ (see Green & Silverman (1994)), where ϕ is the scale parameter of the response variable's distribution, i.e. $\phi = \sigma^2$ and $\lambda := \frac{\sigma^2}{\tau^2}$ for the special case of a Gaussian distributed response. The formula of the penalised log–likelihood, which is to be maximised for the calculation of estimates for $\boldsymbol{\beta}$, is then defined by

$$l_{pen}(y|\boldsymbol{\beta}) = \phi \cdot l(y|\boldsymbol{\beta}) - \frac{1}{2}\lambda\,\boldsymbol{\beta}'\mathbf{P}\boldsymbol{\beta}. \tag{2.9}$$

An estimation algorithm for the regression coefficients is described in section 2.3.3 of this chapter.
In the case of a Gaussian response, maximisation of formula (2.9) is equivalent to minimising the penalised residual sum of squares

$$\text{RSS}_{pen} = (y - \mathbf{X}\boldsymbol{\beta})'\mathbf{W}(y - \mathbf{X}\boldsymbol{\beta}) + \lambda \cdot \boldsymbol{\beta}'\mathbf{P}\boldsymbol{\beta},$$

with $n \times n$ diagonal matrix $\mathbf{W}$ containing weights for all observations. An algorithm for estimating the coefficients $\boldsymbol{\beta}$ in the Gaussian case is presented in section 2.3.2. The estimator for $\boldsymbol{\beta}$ is here given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda\mathbf{P})^{-1}\mathbf{X}'\mathbf{W}y. \tag{2.10}$$

- All prior distributions (2.6) include a variance parameter τ^2 that influences the form of the estimated effect. In this chapter we consider the parameter τ^2 or equivalently the smoothing parameter λ for each nonlinear function as fixed. How to determine an appropriate value for smoothing parameters is the subject of chapter 3.

The following subsections will give detailed information concerning the derivation and specific features of different types of functions.

2.2.1 Linear effects

As mentioned in the last section, a structured additive predictor contains often a parametric part including variables u_j, $j = 1, \ldots, q$ which are to be modelled linearly. Moreover, at least in this thesis, the predictor always contains an intercept term γ_0. For the vector of regression parameters $\boldsymbol{\gamma} = (\gamma_0, \ldots, \gamma_q)'$ for all linear effects including the intercept term, we use no penalisation. In this case, we get maximum likelihood estimates for the coefficients.

For a Gaussian response, the maximum likelihood estimates (or equivalently the least squares estimates) are given by

$$\hat{\boldsymbol{\gamma}} = (\mathbf{U}'\mathbf{W}\mathbf{U})^{-1}\mathbf{U}'\mathbf{W}y,$$

where $\mathbf{U}$ is the $n \times (q+1)$ design matrix including all observations for all respective covariates. Additionally, $\mathbf{U}$ contains a column containing merely the value one for the estimation of the intercept term.
Equivalent to the maximum likelihood approach is to assume independent, diffuse priors $p(\gamma_j) \propto const$, $j = 1, \ldots, q$ for a Bayesian perspective here. In this case, the mode of the posterior distribution is equal to the maximum likelihood estimates.

2.2.2 Categorical Variables

In order to estimate the effect of a categorical variable u with $k \geq 2$ categories, the variable is represented by $k-1$ dummy- or effect variables. We will describe both representations in this section because both can be used with our selection algorithms. In both cases, one of the categories has to be specified as reference category. Without restriction, we number the categories as $1, \ldots, k$ and use the last category k as reference.

2.2.2.1 Dummy Coding

Dummy variables u_j, $j = 1, \ldots, k-1$ are defined as

$$u_j = \begin{cases} 1 & , \text{ if } u = j \\ 0 & , \text{ otherwise.} \end{cases} \tag{2.11}$$

The reference category is indicated by entries of zero in all dummy variables. The effect of the categorical variable is a linear combination of all dummy variables, i.e.

$$\gamma_1 \cdot u_1 + \ldots + \gamma_{k-1} \cdot u_{k-1}$$

and is added to the parametric part of the predictor. That means, all dummies are fixed effects and the parameters γ_j are independent with a diffuse prior each as was described in the last section. The effect of the reference category k is incorporated in the intercept γ_0. The parameters γ_j represent the difference between category j and the reference category. The reason for using only $k-1$ parameters is to get an identifiable model, i.e. to get unique solutions for the parameter estimates. In this thesis, we consider only models containing an intercept term. In this case and when using all possible dummy variables, a constant value

can be added to the intercept and subtracted from all other parameters without changing the predictor, i.e.

$$\gamma_0 + \gamma_1 \cdot u_1 + \ldots + \gamma_k \cdot u_k = (\gamma_0 + c) + (\gamma_1 - c) \cdot u_1 + \ldots + (\gamma_k - c) \cdot u_k.$$

By using only $k-1$ dummies, i.e. by setting $\gamma_k = 0$, this problem is solved and we get unique solutions for the parameter estimates.

2.2.2.2 Effect Coding

Effect coding works similar but the variables u_j are now defined by

$$u_j = \begin{cases} 1 & , \text{ if } u = j \\ -1 & , \text{ if } u = k \\ 0 & , \text{ otherwise} \end{cases} \tag{2.12}$$

for $j = 1, \ldots, k-1$. This leads to a different interpretation of the regression coefficients. A parameter for the reference category can be calculated by

$$\gamma_k = -\sum_{j=1}^{k-1} \gamma_j.$$

The intercept represents the average of all categories and parameter γ_j the difference between this mean and category j.

2.2.3 Continuous covariates

In this section, we consider the simple model $\eta_i = f(x_i)$, $i = 1, \ldots, n$, where function f is supposed to be a smooth function of a continuous variable or time scale x. To approximate these nonlinear functions, there are different approaches in the literature, either depending on local likelihood approaches (see e.g. Fan & Gijbels (1984) and Loader (1999)) or on an expansion in basis functions. In this thesis we will consider the latter case.

2.2.3.1 B–Splines

As basis functions we use polynomial spline functions (splines in short) which are defined piecewise over a set of knots. The knots split up the range of variable x as

$$x_{min} = k_0 < \ldots < k_r = x_{max}.$$

Each basis function, respectively each spline, is

- a polynomial of degree l on the interval $[k_i, k_{i+1}]$, $i = 0, \ldots, r-1$
- $l-1$ times continuously differentiable at the knots k_i (l times at all other points besides the knots).

The function f can be written as a linear combination of the basis functions B_j, i.e.

$$f(x_i) = \beta_1 \cdot B_1(x_i) + \ldots + \beta_p \cdot B_p(x_i),$$

where $p = l + r$ (see De Boor (1978) or Dierckx (1995)). The terms $B_j(x_i)$ denote the value of the j–th basis function evaluated at observation point x_i and serve as new covariates. The function f itself can also be called a spline because it holds the same properties as described above. In matrix notation, each row i of the design matrix $\mathbf{X} = (B_j(x_i))$ contains the function evaluations of all basis functions for the respective observation point x_i. The vector of function evaluations $\mathbf{f}$ is given by $\mathbf{f} = \mathbf{X}\boldsymbol{\beta}$.
In this thesis, we use the B–spline basis whose basis functions are constructed recursively by

$$B_j^l(x) = \frac{x - k_j}{k_{j+l} - k_j} B_j^{l-1}(x) + \frac{k_{j+l+1} - x}{k_{j+l+1} - k_{j+1}} B_{j+1}^{l-1}(x) \tag{2.13}$$

with initial basis functions

$$B_j^0(x) = \begin{cases} 1 & , \text{ if } k_j \leq x < k_{j+1} \\ 0 & , \text{ else.} \end{cases}$$

For the construction of a basis using degree $l > 0$ a set of $2l$ additional knots has to be defined: l knots smaller than x_{min} and l knots larger than x_{max}. The B–spline basis possesses some useful properties:

- It forms a local basis since every basis function is positive only over the range of $l+2$ knots;
- The basis functions are bounded, giving the B–splines good numerical properties;
- The sum over the columns of the design matrix takes the value one in each row.

Figure 2.1 gives an illustration for the construction of a spline function: Part (a) shows B–spline basis functions of degree $l = 2$, part (b) shows weighted basis functions and part (c) the resulting function $f(x)$, that is the sum over all weighted basis functions.
Apart from polynomial splines, there are other possibilities for basis functions, e.g. radial basis functions with the special case of thin–plate splines used by Wood (2003).

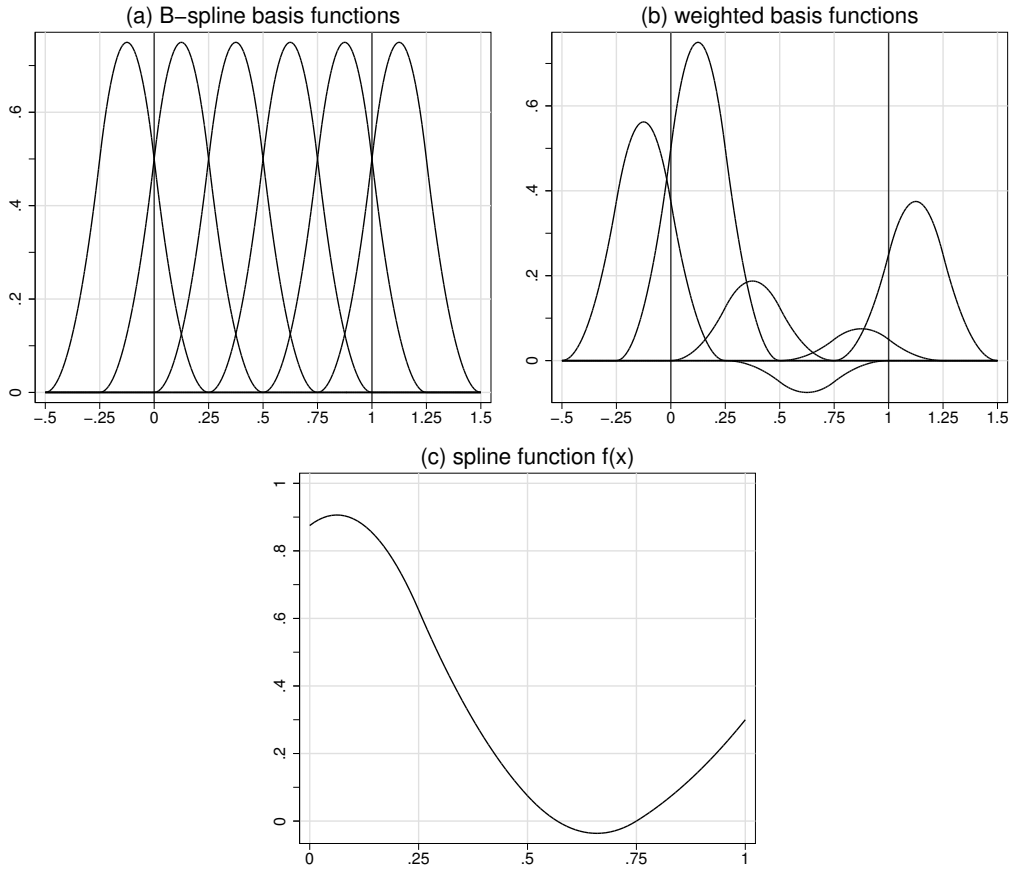

Figure 2.1: (a) B–spline basis functions of degree 2 over the range of $[0; 1]$ *with 5 knots at* $\{0, 0.25, 0.5, 0.75, 1\}$*, (b) weighted basis functions and (c) resulting spline function* $f(x)$*.*

2.2.3.2 P–Splines

One crucial issue with B–splines is the choice of the knots, affecting both the number of knots and their location: many knots result in a rough function, few knots in a smooth one. The question is how many knots should be chosen so that the resulting function is neither too rough nor to smooth. This problem is often called the bias–variance trade–off (see Hastie & Tibshirani (1990)): many knots result in a rough function that is close to the data and therefore has a small bias. But the variance of this function is large. Few knots result in a smooth function that has only a small variance but a high bias instead. A further problem when only a few knots are chosen is where to place the knots.
In order to overcome these problems, there are two different approaches in the literature: the first one is based on adaptive knot selection where the knots are chosen parsimoniously but on positions that result in a sufficiently flexible function. One example is the software MARS introduced by Friedman (1991). Bayesian approaches for adaptive knot selection

are described in Biller (2000). The second approach uses a roughness penalty. The idea is to use a relatively large number of basis functions to gain enough flexibility. Smoothness is achieved by a penalty term that imposes restrictions on the parameter vector $\boldsymbol{\beta}$, like e.g. shrinking the parameters towards zero or penalising too abrupt jumps between adjacent parameters. For that purpose, the log–likelihood is replaced by a penalised log–likelihood defined by

$$l_{pen}(y|\beta_1,\ldots,\beta_p) = \phi \cdot l(y|\beta_1,\ldots,\beta_p) - \frac{1}{2} \cdot \text{ penalty}(\lambda), \tag{2.14}$$

where the trade–off between bias and variance, i.e. between flexibility and smoothness, is controlled by the smoothing parameter λ.

A widely used version of a roughness penalty approach are smoothing splines (see Wahba (1990) or Hastie & Tibshirani (1990) who also present a Bayesian version) where a cubic natural spline basis with knots at all different observation points is used. The integral over the quadratic second derivative, i.e. the curvature, of the resulting function serves as a penalty.

We use the so-called P(enalised)–splines which were introduced by Eilers & Marx (1996) and Marx & Eilers (1998) and which are based on the B–spline basis. Here 20–40 knots are chosen, usually equidistant over the range of x. We describe here only the case of equidistant knots. In order to ensure smoothness a difference penalty term is used that consists of quadratic differences of adjacent coefficients, i.e.

$$\text{penalty}(\lambda) = \lambda \cdot \sum_{j=k+1}^{p} (\Delta^k \beta_j)^2 = \lambda \cdot \boldsymbol{\beta}' \mathbf{P}_k \boldsymbol{\beta},$$

where Δ^k denotes differences of order k. Usually differences of order $k = 1$ or $k = 2$ are used. For equidistant knots they take the form:

$$\Delta^1 \beta_j = \beta_j - \beta_{j-1} \quad \text{and} \quad \Delta^2 \beta_j = \beta_j - 2\beta_{j-1} + \beta_{j-2}. \tag{2.15}$$

Generally, differences of order k can be defined recursively as $\Delta^k \beta_j = \Delta^1(\Delta^{k-1}\beta_j)$ with $\Delta^0 \beta_j = \beta_j$. Hence, second order differences can be calculated as

$$\Delta^2 \beta_j = \Delta^1 \beta_j - \Delta^1 \beta_{j-1} = \beta_j - \beta_{j-1} - (\beta_{j-1} - \beta_{j-2}).$$

By defining $(p-k) \times p$ difference matrices $\mathbf{D}_k$, it is possible to write the differences for all parameters in matrix notation using the product $\mathbf{D}_k \boldsymbol{\beta}$. For $k = 1$ and $k = 2$ the matrices $\mathbf{D}_k$ have the form

$$\mathbf{D}_1 = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{D}_2 = \begin{pmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{pmatrix}.$$

For the penalty matrix $\mathbf{P}_k = \mathbf{D'}_k\mathbf{D}_k$ we get

$$\mathbf{P}_1 = \begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 1 & \end{pmatrix} \text{ and } \mathbf{P}_2 = \begin{pmatrix} 1 & -2 & 1 & & & & \\ -2 & 5 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 5 & -2 \\ & & & & 1 & -2 & 1 \end{pmatrix} \quad (2.16)$$

A small smoothing parameter λ results in a rough function with small bias but large variance. In the limit $\lambda \to 0$ there occurs no penalisation at all. In contrast, a large smoothing parameter results in a function with high bias and low variance. The limit $\lambda \to \infty$ leads to different results depending on the order k chosen for the difference penalty: penalisation of first differences leads to a constant function and penalisation of second order differences to a linear fit. An important issue is how to determine an optimal value for the smoothing parameter. This is one of the central topics of this thesis and will be discussed in detail in chapter 3. Figure 2.2 already illustrates the impact of the smoothing parameter on the estimated P–spline by showing the results for different smoothing parameter values in a simulated data set. The two extreme cases, the unpenalised spline for $\lambda = 0$ (a) and the limits for $\lambda \to \infty$ both for second (e) and for first order (f) penalty, are also shown. The optimal value for λ is determined by an approach described in chapter 3.
A Bayesian version of P–splines has been introduced by Lang & Brezger (2004) and Brezger & Lang (2006). They replace the difference penalties ensuring the smoothness of the function by random walk priors assuming that

$$\beta_j = \beta_{j-1} + u_j, \quad j = 2, \ldots, p \quad (2.17)$$

for a first order random walk and

$$\beta_j = 2 \cdot \beta_{j-1} - \beta_{j-2} + u_j, \quad j = 3, \ldots, p \quad (2.18)$$

for a second order random walk. In both cases, the error terms u_j are $N(0, \tau^2)$ distributed. For the initial values diffuse priors are assumed, i.e. $p(\beta_1) \propto const$ for a first order random walk or $p(\beta_1), p(\beta_2) \propto const$ for a second order random walk, respectively. Alternatively, the prior assumptions (2.17) and (2.18) can be written as

$$\beta_j | \beta_{j-1} \sim N(\beta_{j-1}, \tau^2)$$

for the first order random walk or

$$\beta_j | \beta_{j-1}, \beta_{j-2} \sim N(2\beta_{j-1} - \beta_{j-2}, \tau^2)$$

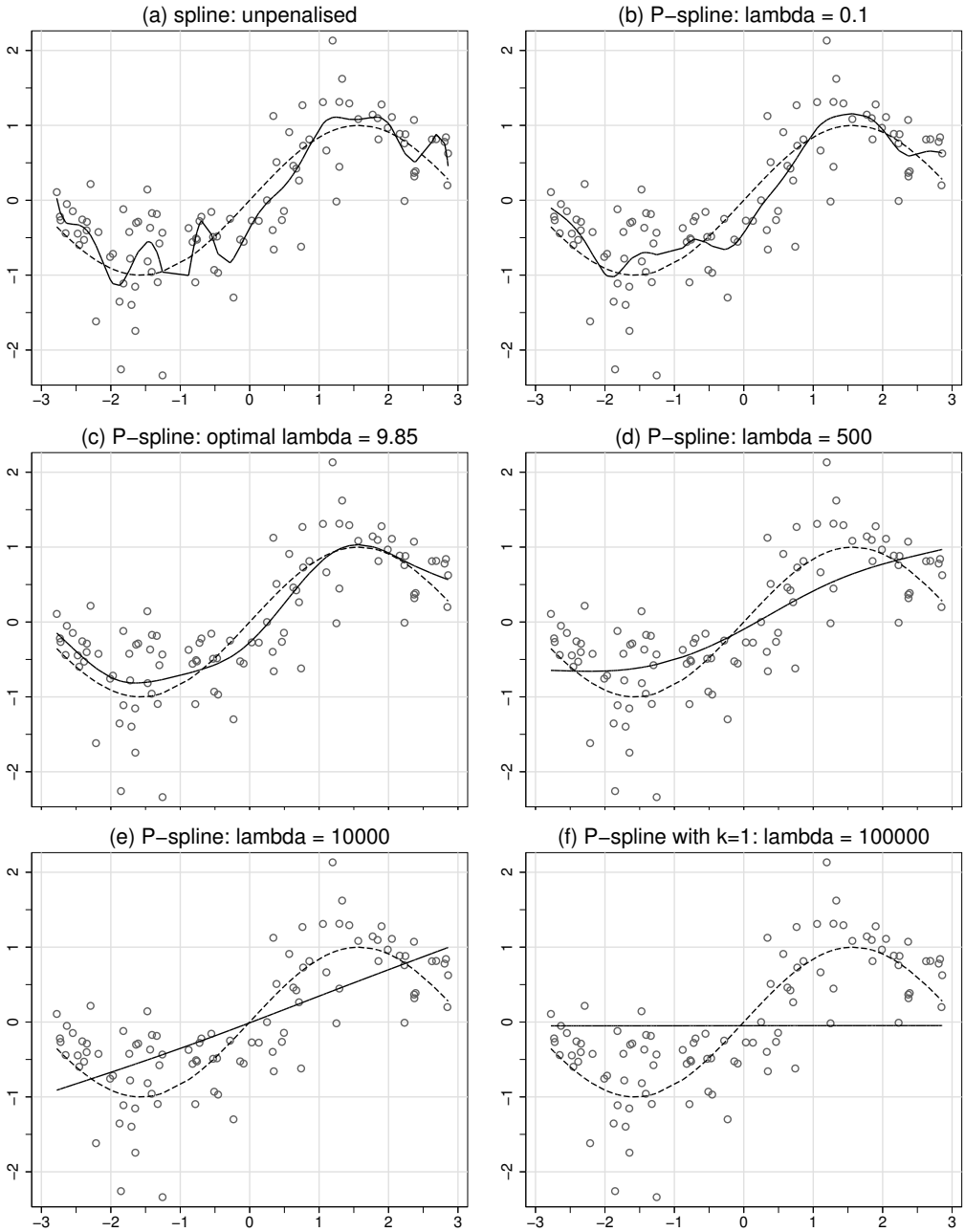

Figure 2.2: Shown are P–splines with different amount of smoothing for the same simulated data y. All plots show the data points, the true underlying function $f(x) = sin(x)$ (dashed line) and an estimated P–spline function (solid line). In each case, the spline consists of 22 cubic basis functions (what is equivalent to 20 knots in the range of x). For plots (a)–(e) a second order penalty was used, for plot (f) a first order penalty. Plots (e) and (f) show the limit of the P–spline for $\lambda \to \infty$: (e) is a straight line (second order penalty) and (f) a constant function (first order penalty).

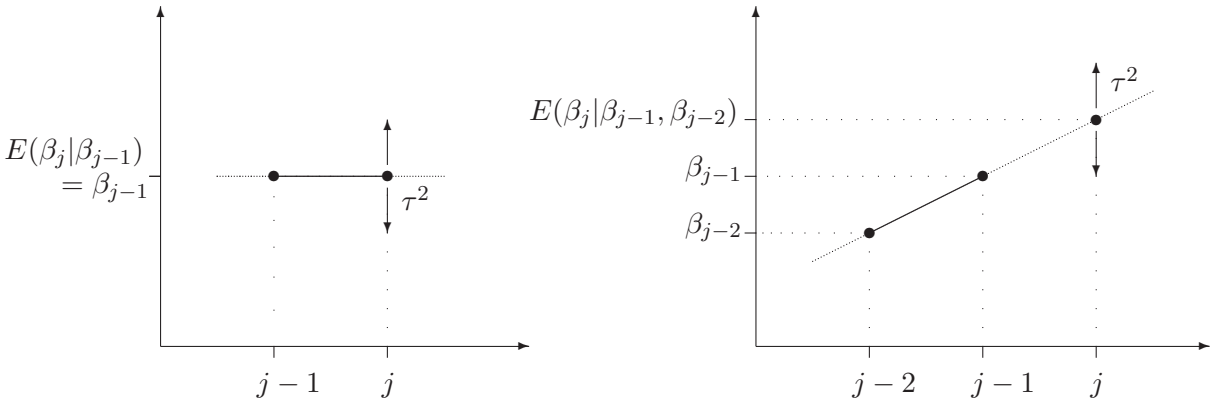

Figure 2.3: Shown is the Bayesian interpretation of the (left plot) first and (right plot) second order random walk. The first order random walk expects parameter β_j to vary around the previous parameter β_{j-1}, whereas the second order random walk expects parameter β_j to vary around the line spanned by the two previous parameters β_{j-1} and β_{j-2}.

for the second order random walk. For both orders $k = 1, 2$, the joint distribution of the regression coefficients $\boldsymbol{\beta}$ is an improper multivariate Gaussian distribution of the general form (2.6), i.e.

$$p(\boldsymbol{\beta}) \propto \exp\left(-\frac{1}{2\tau^2}\boldsymbol{\beta}'\mathbf{P}_k\boldsymbol{\beta}\right), \tag{2.19}$$

where τ^2 is the variance parameter controlling the smoothness of the function. The precision matrix $\mathbf{P}_k$ is for the same order k equal to the penalty matrix of formula (2.16). This fact explains the equivalence of the empirical Bayesian approach to the maximum penalised likelihood approach already mentioned in the introduction of this chapter.
The Bayesian approach allows for a nice interpretation of the penalties: The first order random walk induces a constant trend for the conditional distributions of $\beta_j|\beta_{j-1}$, $j = 2, \ldots, p$. This intuitively explains why the limit for $\lambda \to \infty$ is the constant function (see figure 2.2). In contrast, the second order random walk assumes a linear trend for the conditional distributions of $\beta_j|\beta_{j-1}, \beta_{j-2}$, $j = 3, \ldots, p$, and deviations from this linear trend are penalised. Again this intuitively explains the linear fit as the limit for $\lambda \to \infty$.
Now, we give a more formal explanation for the limiting behaviour if $\lambda \to \infty$. This explanation is provided by the constraint imposed on the parameters by the difference matrix $\mathbf{D}_k$. If $\lambda \to \infty$, maximising the penalised log–likelihood reduces to minimising the penalty term $\boldsymbol{\beta}'\mathbf{D}'_k\mathbf{D}_k\boldsymbol{\beta}$. This term reaches its minimum if $\boldsymbol{\beta}$ fulfils the constraint

$$\mathbf{D}_k\boldsymbol{\beta} = 0.$$

For a random walk of first order this constraint is fulfilled if all parameters are equal. For a second order random walk, the parameters have to lie on a straight line to fulfil the

condition. In general, for a random walk of order k, the parameters fulfil the constraint if they form a polynomial of order $k-1$. The same result is achieved by examining the null space of the penalty matrix. The null space consists of all vectors $\boldsymbol{\beta}$ fulfilling the condition $\mathbf{P}_k\boldsymbol{\beta} = 0$ and thus includes all values for $\boldsymbol{\beta}$ that are not penalised by the matrix. Penalty matrices are symmetric and so the basis of the null space can be calculated via the eigenvalue decomposition. In the case of symmetric matrices, the basis of the null space consists of the eigenvectors corresponding to the zero eigenvalues. The $p \times p$ matrix $\mathbf{P}_1$ for a first order random walk has rank $rk(\mathbf{P}_1) = p - 1$. Hence, the null space of $\mathbf{P}_1$ has dimension 1. Here its basis is a constant vector, i.e. vector $\mathbb{1} = (1, \ldots, 1)'$, which provides the basis for a constant function for the parameters $\boldsymbol{\beta}$.
The rank of the $p \times p$ penalty matrix $\mathbf{P}_2$ for a second order random walk amounts to $rk(\mathbf{P}_2) = p-2$. Hence the null space has dimension 2. The basis of the null space consists of the columns of

$$\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ \vdots & \vdots \\ 1 & p \end{pmatrix}$$

and generates polynomials of degree one for $\boldsymbol{\beta}$.
The constraint imposed on $\boldsymbol{\beta}$ by the penalty matrix also affects the resulting spline f. The null space containing the indices for $\boldsymbol{\beta}$ like in the formula above can be equivalently written using other equally spaced values instead. By dividing the range of variable x in $p-1$ equal parts, the respective null space

$$\begin{pmatrix} 1 & x_{min} \\ 1 & x_{min} + \frac{x_{max}-x_{min}}{p-1} \\ 1 & x_{min} + 2\frac{x_{max}-x_{min}}{p-1} \\ \vdots & \vdots \\ 1 & x_{max} \end{pmatrix}$$

forms a basis of straight lines over the range of x. Figure 2.4 illustrates how the constraints imposed on $\boldsymbol{\beta}$ are transferred to the resulting spline function f. The left part (a) shows the parameters $\boldsymbol{\beta}$ lying on a constant function or on a straight line, respectively. Plot (b) shows the resulting spline functions which use the basis functions of figure 2.1. These basis functions in their weighted version, i.e. multiplied by the respective parameter, are additionally shown in the figure. In the case of equal parameters, the resulting function is constant because of the equally shaped basis functions that sum up to one. Similar reasons lead to a straight line for the resulting function if the parameters lie on a straight line. Hence for the range of x, that is the interval $[0; 1]$, the functions f are also a constant

function or a straight line, respectively. However, a spline function of order l can only reduce to a polynomial of degree $k-1$ if $l \geq k-1$ (see Brezger (2004) who presents a proof for these facts). If, for example, the basis functions are of degree 0, i.e. constant functions, the resulting spline can only reduce to a uniform step function for a second order random walk penalty but not to a straight line.

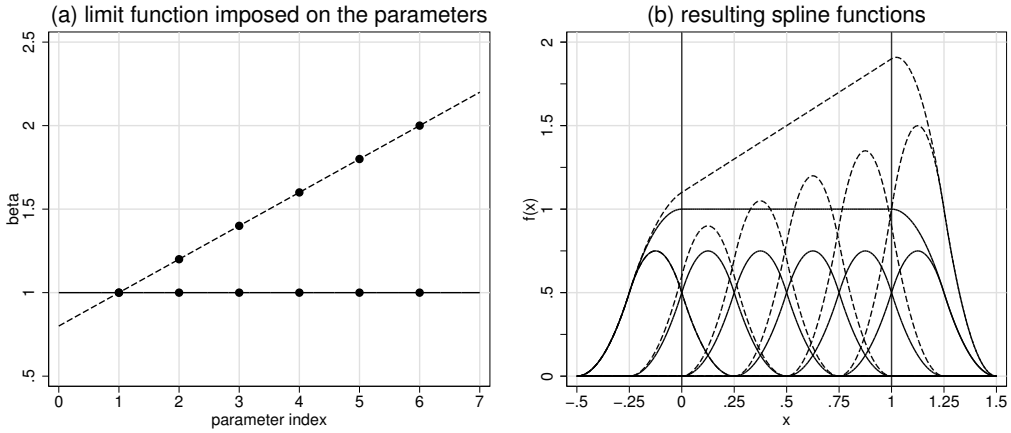

Figure 2.4: Part (a) shows the parameters β lying on a constant function (solid line) or on a straight line (dashed line). The right part (b) shows the resulting spline functions f togehter with the underlying weighted basis functions of figure 2.1. For the range of x, that is the interval $[0;1]$, the resulting function of the constant β is also constant (solid line), whereas the other function is a straight line in accordance to β.

2.2.3.3 Random Walks

A further possibility to model nonlinear functions of continuous variables are random walks (see e.g. Fahrmeir & Lang (2001a)). Random walks should be preferred to P–splines when there are merely few distinct observation points or when the covariate is ordinal. Here, a random walk prior is applied to the function evaluations $f(x)$. Suppose that

$$x_{(1)} < \ldots < x_{(j)} < \ldots < x_{(p)}$$

are the ordered distinct observation points of x. By defining a 0/1-incidence matrix $\mathbf{X}$ indicating the x–value for each observation and by setting $\beta_j := f(x_j)$, the vector of function evaluations can be written as a linear combination $f = \mathbf{X}\boldsymbol{\beta}$. The design matrix $\mathbf{X}$ coincides with a B–spline design matrix of degree $l = 0$ but with knots at every distinct observation point. That means, random walks can be seen as a special case of P–splines where the differences between adjacent function evaluations are penalised. Usually, the distinct observations are not equidistant, and so the priors (2.17) and (2.18) have to be

adjusted.
For the random walk prior of first order, the distribution of error u_j, $j = 2, \ldots, p$, has to account for the distance $\delta_j = x_{(j)} - x_{(j-1)}$ between two adjacent values and changes to

$$u_j \sim N(0, \delta_j \tau^2).$$

This leads to a different penalty matrix

$$\mathbf{P}_1 = \begin{pmatrix} \delta_2^{-1} & -\delta_2^{-1} & & & \\ -\delta_2^{-1} & \delta_2^{-1} + \delta_3^{-1} & -\delta_3^{-1} & & \\ & \ddots & \ddots & \ddots & \\ & & -\delta_{p-1}^{-1} & \delta_{p-1}^{-1} + \delta_p^{-1} & -\delta_p^{-1} \\ & & & -\delta_p^{-1} & \delta_p^{-1} \end{pmatrix}$$

that can be calculated from the ordinary matrix of first differences by

$$\mathbf{P}_1 = \mathbf{D}'_1 \operatorname{diag}(\delta_2^{-1}, \ldots, \delta_p^{-1}) \mathbf{D}_1.$$

The null space of this penalty matrix is again spanned by the vector $\mathbb{1} = (1 \ldots, 1)'$ leading to a constant function for $\lambda \to \infty$.
The adjustment for the second order random walk is more complicated. It can be derived by generalising the second order differences for equidistant values to the case of non–equidistant values. That means, formula

$$\Delta^2 \beta_j = \beta_j - 2\beta_{j-1} + \beta_{j-2} = (\beta_j - \beta_{j-1}) - (\beta_{j-1} - \beta_{j-2})$$

is generalised to formula

$$\Delta^2 \beta_j = \frac{\beta_j - \beta_{j-1}}{\delta_j} - \frac{\beta_{j-1} - \beta_{j-2}}{\delta_{j-1}}, \tag{2.20}$$

comparing the differences between two adjacent parameters with the respective distance. Formula (2.20) is equal to zero if the three parameters are on a straight line. It leads to the Bayesian formulation of the generalised second order random walk with

$$\beta_j = \left(1 + \frac{\delta_j}{\delta_{j-1}}\right) \beta_{j-1} - \frac{\delta_j}{\delta_{j-1}} \beta_{j-2} + u_j, \tag{2.21}$$

and $u_j \sim N(0, w_j \tau^2)$. As described in Fahrmeir & Lang (2001a), there exist several possible choices for the weights w_j. The most simple one is $w_j = \delta_j$. Another possibility that also accounts for the former distance δ_{j-1} is $w_j = \delta_j \left(1 + \frac{\delta_j}{\delta_{j-1}}\right)$.
The common prior for β is again an improper Gaussian distribution like formula (2.19) but the precision matrix has to be adjusted for the distances. It can be calculated as

$$\mathbf{P}_2 = \mathbf{D}'_2 \operatorname{diag}(w_3, \ldots, w_p) \mathbf{D}_2,$$

where $\mathbf{D}_2$ is a generalised second order difference matrix according to formula (2.20). In both cases a basis of the null space is given by

$$\begin{pmatrix} 1 & 0 \\ 1 & \delta_2 \\ 1 & \delta_3 \\ \vdots & \vdots \\ 1 & \delta_p \end{pmatrix} \Leftrightarrow \begin{pmatrix} 1 & x_{(1)} \\ 1 & x_{(2)} \\ 1 & x_{(3)} \\ \vdots & \vdots \\ 1 & x_{(p)} \end{pmatrix},$$

leading to a step function where the function evaluations $f(x_{(j)})$ can be connected by a straight line.

2.2.3.4 P–Splines with shape constraints

In some situations it is known beforehand that the function $f(x)$ posses a certain shape, e.g. it is known to be monotonically increasing. In these cases, it can be useful to apply certain constraints on the function so that the estimated function follows the given form. The type of restrictions most often used with nonparametric functions are monotonicity restrictions, i.e. function $f(x)$ is assumed to be either monotonically increasing or monotonically decreasing. There exist a variety of approaches dealing with imposing these kind of restrictions on splines, e.g. Ramsey (1988) or Tutz & Leitenstorfer (2006) for frequentist approaches and Brezger & Steiner (2006) for a Bayesian approach.
In this thesis we follow the idea introduced and described in Bollaerts, Eilers & Van Mechelen (2006) for a Gaussian response. This approach allows not only for monotonicity restrictions but also for restrictions resulting in a convex or concave function. Their idea is based on the fact that the first and second order derivatives of a B–spline $f(x)$ with equidistant knots can be written as

$$f^{(1)}(x) = \frac{\partial f(x)}{\partial x} = \frac{\partial}{\partial x} \sum_{j=1}^{p} \beta_j B_j^l(x) = (lh)^{-1} l \sum_{j=1}^{p+1} \Delta^1 \beta_j B_j^{l-1}(x) \tag{2.22}$$

for a spline of degree $l \geq 1$ or

$$f^{(2)}(x) = \frac{\partial f^{(1)}(x)}{\partial x} = \prod_{i=1}^{2} ((l+1-i)h)^{-1}(l+1-i) \sum_{j=1}^{p+2} \Delta^2 \beta_j B_j^{l-2}(x) \tag{2.23}$$

respectively, for a spline of degree $l \geq 2$, where h is the distance between adjacent knots. Restricting the differences $\Delta^o \beta_j$ with $o = 1, 2$ to be positive (negative) is a sufficient condition for getting a positive (negative) first ($o = 1$) or second ($o = 2$) order derivative because values h, $l+1-i$ and $B_j^{l-o}(x)$ are all positive. If the resulting derivative is piecewise

constant or piecewise linear, this condition is also necessary.
When using first order differences, function $f(x)$ becomes monotonely increasing for $\Delta^1\beta_j > 0$ and decreasing for $\Delta^1\beta_j < 0$. In contrast, using second order differences results in a convex function for $\Delta^2\beta_j > 0$ or a concave function for $\Delta^2\beta_j < 0$. This fact is also true for functions whose derivative of interest reduces to zero, i.e. if $l < o$.
These conditions can be formulated in the form of a penalty term

$$\boldsymbol{\beta}'\mathbf{P}_{mono}\boldsymbol{\beta} = \sum_{j=o+1}^{p} w(\beta_j)(\Delta_o\beta_j)^2 = \boldsymbol{\beta}'\mathbf{D}_o'\text{diag}(w_{o+1},\ldots,w_p)\mathbf{D}_o\boldsymbol{\beta}$$

with order of derivative $o = 1, 2$ and weights

$$w_j = w(\beta_j) = \begin{cases} 0 & \text{, if } \Delta_o\beta_j \text{ fulfils the restriction} \\ 1 & \text{, otherwise.} \end{cases}$$

Matrix $\mathbf{D}_o$ is the difference matrix of order o as introduced earlier in section 2.2.3.2.
The complete penalty term for function $f(x)$ is composed of two individual penalties: the usual P–spline penalty term of order $k = 1, 2$ which regulates the function's smoothness and the penalty term introduced above which imposes the monotonicity restriction. Thus, the overall penalty is

$$\text{penalty}(\lambda) = \lambda\boldsymbol{\beta}'\mathbf{P}_k\boldsymbol{\beta} + \kappa\boldsymbol{\beta}'\mathbf{D}_o\text{diag}(w_{o+1},\ldots,w_p)\mathbf{D}_o\boldsymbol{\beta} = \lambda\boldsymbol{\beta}'\mathbf{P}_k\boldsymbol{\beta} + \kappa\boldsymbol{\beta}'\mathbf{P}_{mono}\boldsymbol{\beta}, \quad (2.24)$$

where κ is an additional smoothing parameter that we set to a large value, e.g. $\kappa = 100000$, in order to ensure that the constraint is fulfilled. In contrast to λ which has to be determined appropriately, the value for κ is fixed.
In formula (2.24) the penalty matrix $\mathbf{P}_{mono}$ for the restriction depends on the values of $\boldsymbol{\beta}$. This fact complicates the minimisation of the penalised residual sum of squares. Bollaerts, Eilers & Van Mechelen (2006) use a Newton–Raphson method in order to find the optimal solution. This algorithm alternates between estimating parameters $\hat{\boldsymbol{\beta}}$ with fixed penalty matrix $\mathbf{P}_{mono}$, i.e. by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda\mathbf{P}_k + \kappa\mathbf{P}_{mono})^{-1}\mathbf{X}'\mathbf{W}y,$$

and calculating the penalty matrix using the current estimate for $\boldsymbol{\beta}$. This is repeated until the changes in the parameter estimates are sufficiently small. For the first estimate of $\boldsymbol{\beta}$, penalty matrix $\mathbf{P}_{mono}$ is set equal to zero.

2.2.4 Time Scales

The effect of calender time can often be split into a smooth trend and a seasonal component, i.e.

$$f_{time}(t) = f_{trend}(t) + f_{season}(t).$$

In order to approximate the smooth trend function we can use the same kind of nonlinear functions as for continuous covariates, i.e. P–splines or random walks. In this section we describe the modelling of a flexible seasonal component with a certain period *per*. For example, monthly seasonal data corresponds to $per = 12$.
For the modelling of the flexible seasonal component with period *per*, one can use a general autoregressive prior like in Fahrmeir & Lang (2001a), i.e.

$$f_{season}(t) = \beta_t = -\sum_{j=1}^{per-1} \beta_{t-j} + u_t, \tag{2.25}$$

with $t = per, \ldots, p$ and p denoting the latest season observed. The error terms are assumed to be normally distributed with a common variance parameter, i.e. $u_t \sim N(0, \tau^2)$. For the $per - 1$ initial values, we assume diffuse priors. If $\tau^2 \to 0$, the seasonal component becomes fixed and does not vary over time, i.e. $\beta_j = \beta_{j+per}$. In this case, the effect sums to zero and is equivalent to a representation by $per - 1$ effect variables (compare section 2.2.2).
The design matrix for a seasonal component is a $n \times p$ 0/1–incidence matrix where the value one indicates the season a certain observation belongs to. Similar to P–splines or random walks, the common prior distribution for all parameters $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ can be written as an improper Gaussian distribution of the general form (2.6). The precision matrix can be written as $\mathbf{P}_{per} = \mathbf{D}'_{per}\mathbf{D}_{per}$ using a $(p - per + 1) \times p$ difference matrix which is of the form

$$\mathbf{D}_{per} = \begin{pmatrix} 1 & \ldots & 1 & & & \\ & 1 & \ldots & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \underbrace{1 \quad \ldots \quad 1}_{per} & & \end{pmatrix}.$$

For a quarterly seasonal effect, i.e. $per = 4$, the penalty matrix is given by

$$\mathbf{P}_4 = \begin{pmatrix} 1 & 1 & 1 & 1 & & & & & \\ 1 & 2 & 2 & 2 & 1 & & & & \\ 1 & 2 & 3 & 3 & 2 & 1 & & & \\ 1 & 2 & 3 & 4 & 3 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ & & & 1 & 2 & 3 & 3 & 2 & 1 \\ & & & & 1 & 2 & 2 & 2 & 1 \\ & & & & & 1 & 1 & 1 & 1 \end{pmatrix}.$$

In order to identify the limit function for $\tau^2 \to 0$ or equivalently $\lambda \to \infty$, we consider the null space of the penalty matrix. For seasonal components, the null space of the penalty

matrix has dimension $per - 1$ and consists of all time–constant seasonal effects. For the special case of $per = 4$ the basis vectors of the null space are the columns of

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -1 & -1 \\ \vdots & \vdots & \vdots \end{pmatrix}.$$

It is obvious, that all three basis vectors b_1, b_2 and b_3 fulfil the constraint $\mathbf{D}_4 b_k = 0$. Time–constant seasonal effects could alternatively be modelled via effect coding (2.12) like a per-categorical variable.

2.2.5 Spatial covariates

This section deals with the modelling of spatial correlation when the data points are observed at different locations. Often, this spatial correlation can be ascribed to unobservable, spatially varying covariates. The construction of a spatial function where the function evaluations are correlated across the locations is the objective of this section. Sometimes, there are additional unobservable factors whose effect is independent of each other at different locations. In these situations, the spatial effect can be split into a smooth, spatially correlated (structured) part and a locally varying, spatially uncorrelated (unstructured) part (see Besag, York & Mollie (1991)), i.e.

$$f_{spat}(s) = f_{str}(s) + f_{unstr}(s). \tag{2.26}$$

The unstructured effect can be estimated via region–specific i.i.d. Gaussian random effects that are dealt with in section 2.2.6.

In the following, we consider the case that covariate s represents a location in connected geographical regions. In this case, a smooth spatial function can be modelled by a Markov Random Field (MRF). An important part in constructing MRFs is the set of neighbours that must be defined for each region s. Usually, the neighbourhood of one area s consists of all regions that share a common boundary with s. For more complex neighbourhood definitions see Besag, York & Mollie (1991). The idea is that adjacent regions are more alike than any arbitrary locations. Figure 2.5 shows the neighbourhood structure based on common boundaries.

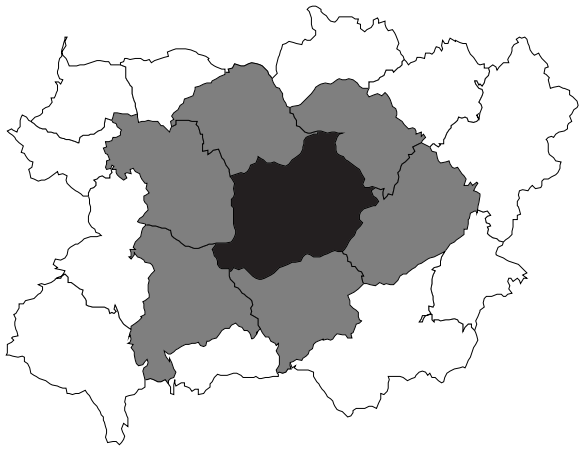

Figure 2.5: The map shows a neighbourhood defined by common boundaries. All grey coloured regions are neighbours to the black one.

The prior for the function evaluations $f_{spat}(s) = \beta_s$ is an extension of the univariate first order random walk. It takes the form

$$\beta_s|\beta_{s'}, s' \neq s \sim N\left(\frac{1}{N_s}\sum_{s'\in\delta_s}\beta_{s'}, \frac{\tau^2}{N_s}\right), \tag{2.27}$$

where δ_s denotes the set of neighbours to region s and $N_s = |\delta_s|$ the number of these neighbours. An improved prior accounting for irregularities in the map can be achieved by defining a weighted version similar to the one for one–dimensional random walks, i.e.

$$\beta_s|\beta_{s'}, s' \neq s \sim N\left(\sum_{s'\in\delta_s}\frac{w_{ss'}}{w_{s+}}\beta_{s'}, \frac{\tau^2}{w_{s+}}\right),$$

where $w_{s+} = \sum_{s'\in\delta_s} w_{ss'}$ and the weights $w_{ss'}$ depend on a distance measure between the regions s and s'. A distance measure can be specified according to one of the following examples:

- If one always assumes the same distance between adjacent regions, the weights become $w_{ss'} = 1$ and the prior reduces to formula (2.27).
- Weights can be chosen proportional to the length of the common boundary.
- Weights can be chosen inverse proportional to the Euclidian distance $d(s, s')$ between the centroids of two regions, i.e. $w_{ss'} \propto \exp(-d(s, s'))$.

For p regions, the design matrix X is a 0/1-incidence matrix of order $n \times p$ indicating whether observation i belongs to region s ($X_{is} = 1$) or not ($X_{is} = 0$). The common prior

for all parameters $\boldsymbol{\beta}$ is again an improper Gaussian prior of the form (2.6) with a $p \times p$ precision or penalty matrix $\mathbf{P}_{spat}$ depending on the weights

$$\begin{aligned} p_{ss} &= w_{s+} \\ p_{ss'} &= \begin{cases} -w_{s,s'} & \text{, if } s \text{ and } s' \text{ are neighbours} \\ 0 & \text{, otherwise.} \end{cases} \end{aligned} \tag{2.28}$$

The precision matrix $\mathbf{P}_{spat}$ is again rank–deficient with $rk(\mathbf{P}_{spat}) = p - 1$. Like in the case of a one–dimensional random walk, the basis of the null space is the $\mathbf{1}$–vector. Here, $f(s_j) = \beta_j$, so the penalty matrix influences f directly. The limit of the spatial function for $\lambda \to \infty$ or equivalently $\tau^2 \to 0$ is therefore a constant function indicating no differences between the regions.

Note, that the function evaluation for a region can be estimated even if there are no observations for this region available. This is due to the smoothness assumptions included in the prior distribution.

The MRF can be also applied when a relatively small number of exact locations $s = (s_x, s_y)$ are available by defining a symmetric neighbourhood structure. For a large set of different locations or if a surface estimation is required, there exist other, more preferable approaches. One possibility, not implemented for our selection algorithms, are Gaussian Random Field (GRF) priors that assume a two–dimensional correlation function to model spatial correlation (see Kammann & Wand (2003) or Kneib (2006) for instance). Another possibility basing the estimation on 2–dimensional penalised tensor–product splines is described in section 2.2.8 of this chapter. The disadvantage of this approach (in contrast to a GRF) are the anisotropic basis functions (see Kneib (2006)). Here, the lines of the basis functions' contour plots form no circles, especially for a small degree l. This implies, that different directions are treated unequally.

2.2.6 Unobserved heterogeneity

In this section we deal with data that consists of repeated observations of individuals or within clusters such as groups or regions. There can be differences between individual units or clusters that are due to unobserved factors. To overcome this problem, it is possible to estimate a random effect that models the differences between each unit and the overall mean. For this purpose, we use i.i.d. Gaussian random effects assuming the parameters β_i, $i = 1, \ldots, p$, for the p individuals to be independently normally distributed with a common variance parameter, i.e.

$$\beta_i \sim N(0, \tau^2).$$

Here the joint distribution for $\boldsymbol{\beta}$ is a *proper* normal distribution. Nevertheless, it can be written in the same general form (2.6) as all other priors by using the identity matrix as

precision matrix, i.e. $\mathbf{P}_{rand} = \mathbf{I}$. This matrix is of full rank, so that the null space is of dimension zero only containing the null vector $\mathbf{0} = (0, \ldots, 0)'$. In this case, using a large smoothing parameter results in a function equal to zero.
The design matrix $\mathbf{X}$ is again a 0/1-incidence matrix of order $n \times p$. If the random effect is used to estimate an unstructured spatial effect, the design matrix of the random effect is exactly identical to the one belonging to the structured spatial effect.
For the limit $\tau^2 \to \infty$ or equivalently $\lambda \to 0$, the random effect consists of unpenalised parameters for all p individuals. This is equivalent to estimating the function via p dummy variables. As was mentioned in section 2.2.2, this again is equivalent to using only $p - 1$ dummy variables and an intercept term. Hence, a random effect includes a constant term like all other univariate functions described in this chapter. But in contrast to other univariate functions, random effects penalise the constant term. This can be seen from the penalty matrix whose null space contains merely the null vector.

2.2.7 Varying Coefficients

In the preceding sections various approaches for the modelling of different kinds of one–dimensional effects have been introduced. We now describe extensions that allow us to model two–dimensional interactions. Varying coefficients were first popularised by Hastie & Tibshirani (1993) in the context of smoothing splines. Here, the slope of a variable z varies smoothly over the range of another variable v by defining the term

$$f(v, z) = g(v)z. \tag{2.29}$$

Often, the interacting variable z is categorical, but it can be continuous as well. The effect modifier v can be either a continuous variable, a spatial location or a group indicator. The vector of function evaluations $\mathbf{f}$ can be written as linear combination

$$f = \mathbf{X}\boldsymbol{\beta},$$

using a design matrix $\mathbf{X}$ and a vector of coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$. The design matrix for $f(v, z)$ is based both on the observations of z and of v and is calculated as

$$\mathbf{X} = \text{diag}(z_1, \ldots, z_n)\mathbf{V},$$

where $\mathbf{V}$ is the design matrix corresponding to $g(v)$. The prior for the effect modifying function g, or the parameters $\boldsymbol{\beta}$ respectively, can be any of the priors introduced in sections 2.2.3–2.2.6 and 2.2.8 according to the type of covariate v.
Some special cases of varying coefficients sometimes appear under a different name: if

the effect modifier v is a group indicator, the two–dimensional function $f(v, z)$ is called random slope. Models including a varying coefficient with a spatial effect as modifying function are known as geographically weighted regression in the geography literature (see Fotheringham, Brunsdon & Charlton (2002)). Dynamic models are based on time–varying coefficients (see Fahrmeir & Tutz (2001)).
Finally, we take a look at the limit of $f(v, z)$ for $\lambda \to \infty$ or equivalently $\tau^2 \to 0$. This depends on the prior distribution imposed on the univariate function $g(v)$. The limit functions $g^{(\infty)}(v)$ were described in the respective sections for all univariate functions g. The limit function of the varying coefficient is

$$f^{(\infty)}(v, z) = g^{(\infty)}(v)z.$$

That means, $f^{(\infty)}(v, z)$ is equal to zero if $g(v)$ is a random effect or is a linear effect of z for a random walk prior of first order (MRF or P–spline of first order). For a second order random walk prior we obtain an interaction of the form $f^{(\infty)}(v, z) = c_1 \cdot z + c_2 \cdot v \cdot z$.

2.2.8 Interaction surfaces

A varying coefficient can be too restrictive if both interacting variables x_1 and x_2 are continuous. In this case, a more flexible approach is achieved by estimating a smooth two–dimensional surface. As described in Lang & Brezger (2004) and Brezger & Lang (2006), we use an approach based on bivariate P–splines. Similar to the univariate P–splines described in section 2.2.3, it is assumed that the unknown smooth surface $f(x_1, x_2)$ can be approximated by a linear combination of basis functions, i.e.

$$f(x_1, x_2) = \sum_{j=1}^{p_1} \sum_{k=1}^{p_2} \beta_{jk} B_{jk}(x_1, x_2),$$

where the two–dimensional basis functions form a tensor product of univariate B–spline basis functions for x_1 and x_2, i.e.

$$B_{jk}(x_1, x_2) = B_j(x_1) \cdot B_k(x_2).$$

Figure 2.6 shows some of those tensor–product basis functions for degree $l = 2$. Shown are only nonoverlapping basis functions.
The function evaluations of the two–dimensional basis functions can be written as a $n \times p_1 p_2$ design matrix $\mathbf{X} = (B_{jk}(x_{i1}, x_{i2}))$ with an associated parameter vector $\boldsymbol{\beta} = (\beta_{1,1}, \ldots, \beta_{1,p_2}, \ldots, \beta_{p_1,p_2})'$. We confine bivariate B–splines to the case of $p_1 = p_2 = p$ so that both the x_1– and the x_2–direction are treated equally.
For the prior distribution of the parameter vector $\boldsymbol{\beta}$ we distinguish two different cases:

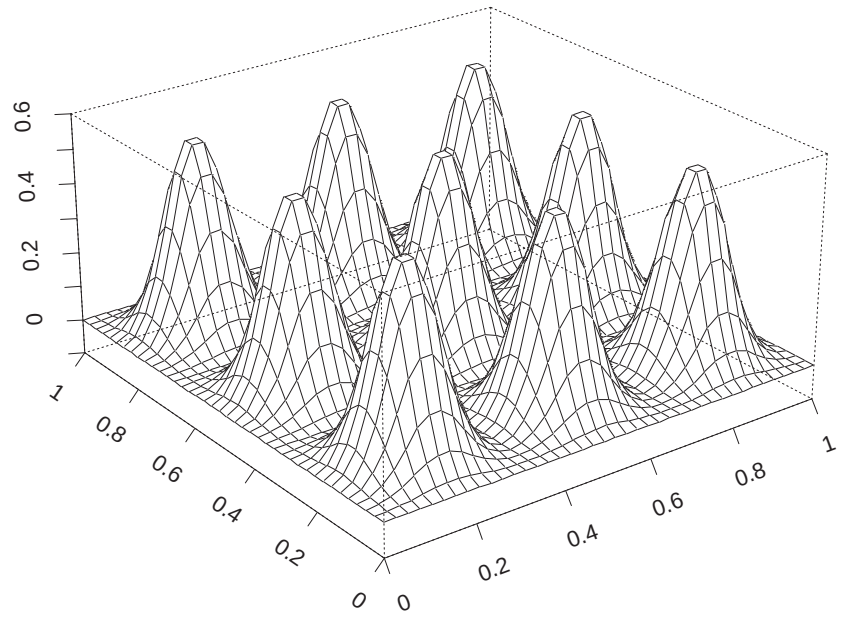

Figure 2.6: Tensor product B–spline basis functions of degree $l = 2$. The plot shows only nonoverlapping basis functions.

- We are only interested in the two–dimensional effect $f(x_1, x_2)$ of x_1 and x_2.
- We want to estimate an ANOVA type interaction model, i.e. the overall surface $f(x_1, x_2)$ consists of an interaction component $f_{inter}(x_1, x_2)$ and two main effects $f_1(x_1)$ and $f_2(x_2)$ (see Chen (1993)). The two main effects are supposed to contain as much information as possible whereas the interaction component is supposed to represent only the deviation of the overall surface from the sum of main effects.

In the following sections we will describe the two cases in more detail.

2.2.8.1 Interaction surfaces as functions of two–dimensional covariates

In this section we describe prior distributions for the first case when the predictor only includes the two–dimensional function $f(x_1, x_2)$ and no main effect. Here, we use two different possibilities for the prior distribution of $\boldsymbol{\beta}$: a bivariate first and a bivariate second order random walk.
A bivariate first order random walk can be obtained by applying an unweighted MRF prior (2.27) on the four adjoining parameters which lie on a regular grid. In this case, the

conditional prior distributions for parameters with four neighbours take the form

$$\beta_{jk}|\beta_{j'k'}, j' \neq j, k' \neq k \sim N\left(\frac{1}{4}(\beta_{j-1,k} + \beta_{j,k-1} + \beta_{j+1,k} + \beta_{j,k+1}), \frac{\tau^2}{4}\right), \tag{2.30}$$

with $j, k = 2, \ldots, p-1$. This is illustrated in figure 2.7 (a). The conditional prior distributions for parameters at the corners and edges have to be adjusted appropriately, see Lang & Brezger (2004).

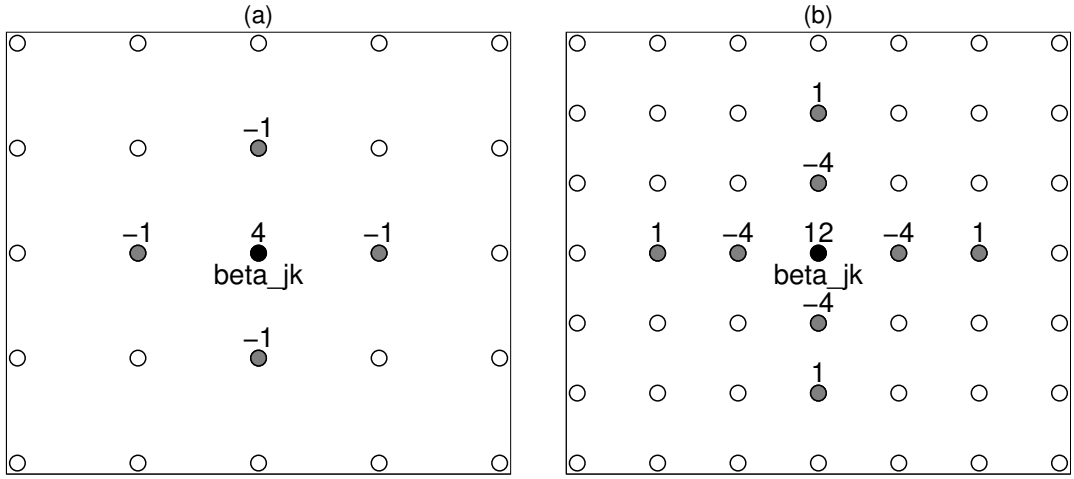

Figure 2.7: Conditional prior distributions for β_{jk}, indicated by a black dot, together with the coefficients of the precision matrix for (a) a first order and (b) a second order random walk. The neighbours are indicated in grey.

The joint prior distribution of $\boldsymbol{\beta}$ can be written in the general form (2.6) by using the $p^2 \times p^2$ precision matrix $\mathbf{P}_1^{(2)}$ which is defined by formula (2.28). Here, the upper index $^{(2)}$ indicates the penalisation of a two–dimensional function. Matrix $\mathbf{P}_1^{(2)}$ corresponds to the penalty term

$$\text{penalty}(\lambda) = \lambda \boldsymbol{\beta}' \mathbf{P}_1^{(2)} \boldsymbol{\beta}$$

where the amount of smoothness is controlled by one smoothing parameter. Hence, the same amount of smoothing is applied both in the direction of x_1 and of x_2. Alternatively, matrix $\mathbf{P}_1^{(2)}$ can be calculated from the one–dimensional $p \times p$ precision matrix $\mathbf{P}_1$ of a first order random walk that is applied in both directions as

$$\mathbf{P}_1^{(2)} = \mathbf{I} \otimes \mathbf{P}_1 + \mathbf{P}_1 \otimes \mathbf{I}. \tag{2.31}$$

Eilers & Marx (2003) use this representation (2.31) for the definition of an anisotropic penalty where the strength of the penalisation may differ between the directions of x_1

and x_2. This is achieved by using an individual smoothing parameter for each of the two directions leading to the penalty

$$\text{penalty}(\lambda_1, \lambda_2) = \boldsymbol{\beta}'(\lambda_1 \mathbf{I} \otimes \mathbf{P}_1 + \lambda_2 \mathbf{P}_1 \otimes \mathbf{I})\boldsymbol{\beta}. \tag{2.32}$$

We use the penalty based on one smoothing parameter which corresponds to the general form (2.6). In this case, the limit for $\lambda \to \infty$ or $\tau^2 \to 0$ is a constant function because vector $\mathbf{1}$ forms the basis of the null space of $\mathbf{P}_1^{(2)}$. The penalty matrix is of rank $rk(\mathbf{P}_1^{(2)}) = p^2 - 1$. There are several proposals for constructing a bivariate second order random walk (see e.g. Rue & Held (2005)). The easiest possibility is to replace the univariate penalty matrices of first order in formula (2.31) by matrices of second order, i.e.

$$\mathbf{P}_2^{(2)} = \mathbf{I} \otimes \mathbf{P}_2 + \mathbf{P}_2 \otimes \mathbf{I}. \tag{2.33}$$

This leads to a dependency structure where the parameter β_{kj} depends on the eight nearest neighbours in x_1– and x_2–direction. Similar to the first order random walk the parameter does not depend on parameters apart from the main directions, like e.g. on parameters on the diagonals. The conditional prior distribution for parameters β_{jk} for $j, k = 3, \ldots, p-2$, i.e. having a complete set of neighbours, is illustrated in figure 2.7 (b). Again, the priors have to be adjusted appropriately for the corners and edges.
The precision matrix (2.33) also allows for an unequal penalisation in the directions of x_1 and x_2 by using two different smoothing parameters as described in Eilers & Marx (2003). Again, we use only one smoothing parameter and thus the same amount of smoothing in both directions. This makes it possible to write the joint prior distribution of $\boldsymbol{\beta}$ in the general form (2.6).
The basis of the null space of matrix $\mathbf{P}_2^{(2)}$ is presented by the columns of matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 \cdot 1 \\ 1 & 1 & 2 & 1 \cdot 2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & p & 1 \cdot p \\ 1 & 2 & 1 & 2 \cdot 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 2 & p & 2 \cdot p \\ 1 & 3 & 1 & 3 \cdot 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & p & p & p \cdot p \end{pmatrix}.$$

Hence in this case, the limit for $\lambda \to \infty$ or $\tau^2 \to 0$ is a linear interaction of the form

$$f^{(\infty)}(x_1, x_2) = c_0 + c_1 \cdot x_1 + c_2 \cdot x_2 + c_3 \cdot x_1 \cdot x_2.$$

2.2.8.2 Interaction surfaces for ANOVA type interactions

In this second, more difficult case, the predictor contains not only the interaction $f_{inter}(x_1, x_2)$ but also the main effects $f_1(x_1)$ and $f_2(x_2)$, i.e.

$$\eta = \gamma_0 + f_1(x_1) + f_2(x_2) + f_{inter}(x_1, x_2).$$

Here, the interaction component $f_{inter}(x_1, x_2)$ represents only the deviation of the predictor from the sum of the two main effects (see Gu (2002)). Hence, the two main effects must contain as much information as possible whereas the interaction contains only the information that cannot be modelled by the main effects. In this case, usually a two–dimensional surface smoother together with two one–dimensional smoothers is estimated. This approach, however, has considerable drawbacks regarding the calculation of degrees of freedom (see section 3.3): The sum of the three individual degrees of freedom cannot be used as an approximation to the overall degrees of freedom. Moreover, the convergence of modular algorithms like the backfitting algorithm (compare section 2.3.2) is slow for such highly correlated functions. We therefore follow a different approach: We specify and estimate a two–dimensional surface based on tensor product P–splines and compute the resulting decomposition into main effects and interaction component thereafter.

Penalty matrix for a decomposition of the surface smoother into main effects

In the following we construct a penalty matrix such that, for the limit $\lambda \to \infty$, we get an exact decomposition of the overall surface into two main effects (without an interaction component). Hence, we need to know the conditions under which a two–dimensional tensor product function can be split into two main effects, i.e.

$$f(x_1, x_2) = \sum_{j,k=1}^{p} \beta_{jk} B_j(x_1) B_k(x_2) \stackrel{!}{=} \sum_{j=1}^{p} a_j B_j(x_1) + \sum_{k=1}^{p} b_k B_k(x_2) = f_1(x_1) + f_2(x_2),$$

with main effects coefficients a_j and b_k for $j, k = 1, \ldots, p$. The exact calculation of these conditions is described in section A.1 of the appendix. It turns out that, for $\lambda \to \infty$, function $f(x_1, x_2)$ can be decomposed into two main effects by using a penalty which is based on differences of differences of the parameters, i.e. on

$$\Delta^{(1,0)} \Delta^{(0,1)} \beta_{j,k} = \beta_{j,k} - \beta_{j-1,k} - \beta_{j,k-1} + \beta_{j-1,k-1},$$

with $j, k = 2, \ldots, p$ and a two–dimensional difference operator Δ. The resulting penalty term (compare Rue & Held (2005)) is given by

$$\lambda \cdot \sum_{j=2}^{p} \sum_{k=2}^{p} (\Delta^{(1,0)} \Delta^{(0,1)} \beta_{j,k})^2.$$

These $(p-1)^2$ differences of differences can be summarised in the $(p-1)^2 \times p^2$ difference matrix $\mathbf{D}$ given by

$$\mathbf{D} = \left(\begin{array}{cccc|cccc|cccc|cccc} 1 & -1 & & & -1 & 1 & & & & & & & & & & \\ & \ddots & \ddots & & & \ddots & \ddots & & & & & & & & & \\ & & 1 & -1 & & & -1 & 1 & & & & & & & & \\ \hline & \cdots & & & & \cdots & & & & \cdots & & & & \cdots & & \\ \hline & & & & 1 & -1 & & & -1 & 1 & & & & & & \\ & & & & & \ddots & \ddots & & & \ddots & \ddots & & & & & \\ & & & & & & 1 & -1 & & & -1 & 1 & & & & \end{array}\right) \tag{2.34}$$

where each of the $(p-1) \cdot p$ submatrices is of order $(p-1) \times p$. For $\mathbf{D}\boldsymbol{\beta} = 0$ the surface is exactly decomposed into main effects (compare section A.1 of the appendix). By using the corresponding penalty matrix $\mathbf{P} := \mathbf{D}'\mathbf{D}$ it is possible to estimate $\hat{\boldsymbol{\beta}}$ such that $\mathbf{D}\hat{\boldsymbol{\beta}} = 0$ for $\lambda \to \infty$. Matrix $\mathbf{P}$ can alternatively be derived as Kronecker product of two one–dimensional first order random walk penalty matrices, i.e.

$$\mathbf{P} = \mathbf{P}_1 \otimes \mathbf{P}_1. \tag{2.35}$$

This penalty matrix describes a neighbourhood structure where every parameter depends on its eight nearest neighbours, i.e. both on parameters in x_1– and x_2–direction and on parameters on the diagonals (see Rue & Held (2005)). The conditional prior distribution for parameters β_{jk}, with $j, k = 2, \ldots, p-1$, i.e. having a complete set of neighbours, is illustrated in figure 2.8. Again, the priors have to be adjusted appropriately for parameters at corners and edges, shown in figure 2.9.
The rank of matrix $\mathbf{P}$ is $(p-1)^2$ because of the property $rk(\mathbf{P}) = rk(\mathbf{P}_1) \cdot rk(\mathbf{P}_1)$ which holds for the rank of a Kronecker product. Hence, the null space of $\mathbf{P}$ has dimension $p^2 - (p-1)^2 = 2p - 1$ which is in accordance with the degrees of freedom of two unpenalised one–dimensional spline functions. That means, using penalty matrix $\mathbf{P}$ from formula (2.35) yields two unpenalised main effects for the limit $\lambda \to \infty$.

Combined penalty matrix

In the last subsection we presented a penalty matrix that, for $\lambda \to \infty$, leads to an exact decomposition of the tensor product spline into two unpenalised main effects. Since unpenalised splines usually are too wiggly, we now modify the penalty matrix in such a way that the overall surface can be decomposed into two penalised main effects. For that purpose, we combine penalty matrix (2.35) with anisotropic two–dimensional penalty matrices (compare formula (2.32)). Hence, the two directions of x_1 and x_2 are no longer treated equally (compare Eilers & Marx (2003)), but there is no reason why they should. Two

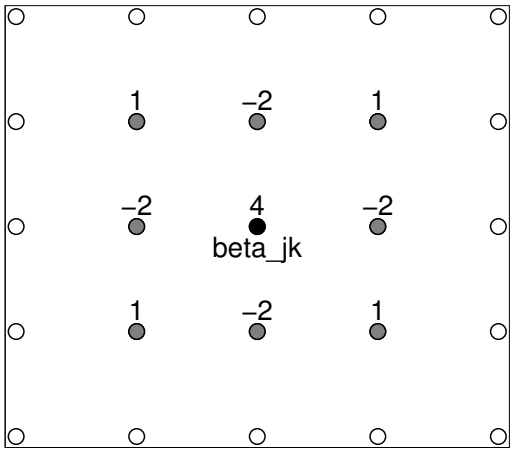

Figure 2.8: Shown are the conditional prior distributions for β_{jk}, indicated by a black dot, together with the coefficients of the precision matrix for the Kronecker product of two one–dimensional first order random walk matrices. The neighbours are indicated in grey.

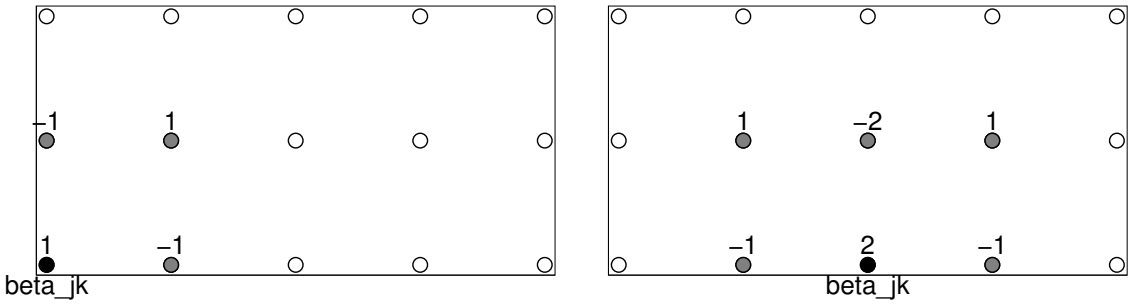

Figure 2.9: Shown are the conditional prior distributions for β_{jk} at the corners (left) or edges (right) together with the coefficients of the precision matrix. β_{jk} is indicated by a black dot, the neighbours are indicated in grey.

main effects not connected through an interaction do not have the same penalty, either. The combined penalty matrix is given by

$$\mathbf{P}_{comp} = \lambda \mathbf{P} + \frac{\lambda_1}{p}\mathbf{P}_{x_1} + \frac{\lambda_2}{p}\mathbf{P}_{x_2}. \tag{2.36}$$

Matrix $\mathbf{P}_{x_1} = \mathbf{P}_{k_1} \otimes \mathbf{I}_p$ and smoothing parameter λ_1 control the penalisation in the direction of x_1, whereas $\mathbf{P}_{x_2} = \mathbf{I}_p \otimes \mathbf{P}_{k_2}$ and λ_2 do the same for x_2. The one–dimensional penalty matrices $\mathbf{P}_{k_1}$ and $\mathbf{P}_{k_2}$ can be based on first or second order random walks (i.e. $k_1, k_2 = 1, 2$) and the order of the penalties may be different.

Note that formula (2.36) does not use the smoothing parameters λ_1 and λ_2 themselves but the values λ_1/p and λ_2/p instead. This is done in order to account for the fact that the penalty matrices $\mathbf{P}_{x_1}$ and $\mathbf{P}_{x_2}$ are p times as strong as matrices $\mathbf{P}_{k_1}$ and $\mathbf{P}_{k_2}$. This fact

is explained in detail in section A.2 of the appendix. The penalty term corresponding to matrix $\mathbf{P}_{comp}$ is given by

$$\text{penalty}(\lambda, \lambda_1, \lambda_2) = \boldsymbol{\beta}' \, \mathbf{P}_{comp} \, \boldsymbol{\beta} \tag{2.37}$$

and serves as overall penalty for the surface $f(x_1, x_2)$.
The overall penalty matrix $\mathbf{P}_{comp}$ imposes a neighbourhood structure where each parameter β_{jk} with $j, k = 2, \ldots, p-1$ depends either on 8, 10 or 12 nearest neighbours depending on the order of the penalisation of the main effects. The different neighbourhood structures are shown in figure 2.10.

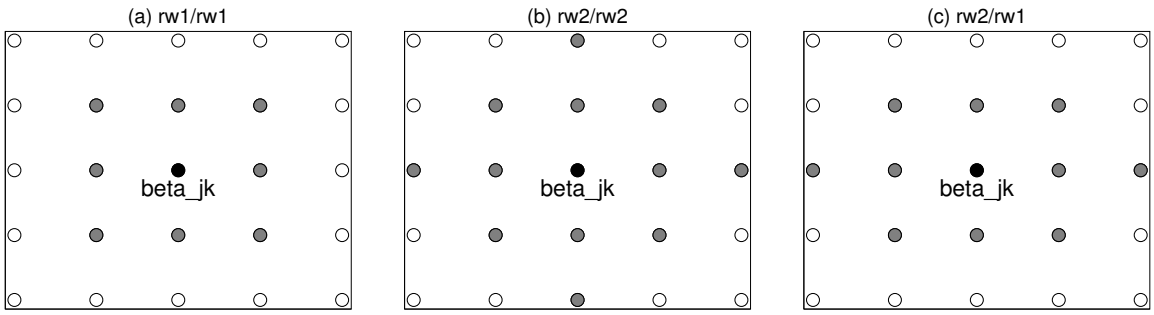

Figure 2.10: Shown is the neighbourhood structure for β_{jk} for different one–dimensional penalisations. Plot (a) shows the neighbourhood structure for two first order random walks and plot (b) for two second order random walks. Plot (c) shows a combined neighbourhood structure using a second order random walk in the direction from left to right and a first order random walk otherwise. The parameter β_{jk} is in each case indicated by a black dot, the neighbours are indicated in grey.

The combination of the three penalty matrices has the following nice properties:

- The limit $\lambda \to \infty$ results in a main effects model. The main effects are P–splines with smoothing parameters λ_1 and λ_2.
- The limit $\lambda \to 0$ yields the anisotropic penalties described in Eilers & Marx (2003) as a special case.
- The limit $\lambda_1 \to 0$ and $\lambda_2 \to 0$ yields the Kronecker product (2.35) as a special case.
- The limit $\lambda \to \infty$, $\lambda_1 \to \infty$ and $\lambda_2 \to \infty$ results in a main effects model with linear or constant main effects depending on the order of matrices $\mathbf{P}_{k_1}$ and $\mathbf{P}_{k_2}$.

Some examples for different combinations of the three smoothing parameters are illustrated in the appendix A.4.

After estimation, the overall surface $\hat{f}(x_1, x_2)$ is decomposed into the two main effects $\hat{f}_1(x_1)$ and $\hat{f}_2(x_2)$ and the interaction component $\hat{f}_{inter}(x_1, x_2)$ by

$$\hat{f}(x_1, x_2) = \hat{f}_1(x_1) + \hat{f}_2(x_2) + \hat{f}_{inter}(x_1, x_2).$$

In order to ensure that the two main effects contain as much information as possible we impose the following constraints on the interaction component (compare Chen (1993) and Lang & Brezger (2004)):

$$\begin{aligned}
\bar{f}_{inter}(x_2) &= \frac{1}{r(x_1)} \int_{x_{1,min}}^{x_{1,max}} f_{inter}(x_1, x_2) dx_1 = 0 \text{ for all distinct values of } x_2, \\
\bar{f}_{inter}(x_1) &= \frac{1}{r(x_2)} \int_{x_{2,min}}^{x_{2,max}} f_{inter}(x_1, x_2) dx_2 = 0 \text{ for all distinct values of } x_1, \\
\bar{f}_{inter} &= \frac{1}{r(x_1) r(x_2)} \int_{x_{2,min}}^{x_{2,max}} \int_{x_{1,min}}^{x_{1,max}} f_{inter}(x_1, x_2) dx_1 dx_2 = 0
\end{aligned}$$

with $r(x_1) = x_{1,max} - x_{1,min}$ and $r(x_2) = x_{2,max} - x_{2,min}$. Hence row wise, column wise and overall means of the interaction component are supposed to be zero. In order to obtain a function fulfilling these constraints the integrals

$$\begin{aligned}
\bar{f}_{1|2}(x_2) &= \frac{1}{r(x_1)} \int_{x_{1,min}}^{x_{1,max}} f(x_1, x_2) dx_1, \\
\bar{f}_{1|2}(x_1) &= \frac{1}{r(x_2)} \int_{x_{2,min}}^{x_{2,max}} f(x_1, x_2) dx_2, \\
\bar{f}_{1|2} &= \frac{1}{r(x_1) r(x_2)} \int_{x_{2,min}}^{x_{2,max}} \int_{x_{1,min}}^{x_{1,max}} f(x_1, x_2) dx_1 dx_2
\end{aligned}$$

of the overall two–dimensional function must be calculated first. Then the interaction component is calculated by

$$\hat{f}_{inter}(x_1, x_2) = \hat{f}(x_1, x_2) - \bar{f}_{1|2}(x_2) - \bar{f}_{1|2}(x_1) + \bar{f}_{1|2}.$$

Afterwards, the two main effects are extracted. For the main effects we consider the additional constraints (compare section 2.3.1)

$$\begin{aligned}
\bar{f}_1 &= \frac{1}{r(x_1)} \int_{x_{1,min}}^{x_{1,max}} f_1(x_1) dx_1 = 0, \\
\bar{f}_2 &= \frac{1}{r(x_2)} \int_{x_{2,min}}^{x_{2,max}} f_2(x_2) dx_2 = 0
\end{aligned}$$

so that the main effects are obtained by

$$\begin{aligned}
\hat{f}_1(x_1) &= \bar{f}_{1|2}(x_1) - \bar{f}_{1|2}, \\
\hat{f}_2(x_2) &= \bar{f}_{1|2}(x_2) - \bar{f}_{1|2}.
\end{aligned}$$

Note, that the intercept term γ_0 of the predictor has to be corrected by

$$\hat{\gamma}_0 \longrightarrow \hat{\gamma}_0 + \bar{f}_{1|2}$$

in order to ensure that the predictor remains unchanged.
Both main effects $\hat{f}_1(x_1)$ and $\hat{f}_2(x_2)$ are P–splines what is easily shown by inserting the tensor product representation of f into $\bar{f}_{1|2}(x_2)$ and $\bar{f}_{1|2}(x_1)$ (compare section A.3 of the appendix).
Note that this approach for two–dimensional interactions as described here can be used for non–overlapping interactions only. That means that two interaction terms must not have a common main effect.

2.3 Inference

In this section, we describe inference for the regression coefficients in a model with a structured additive predictor (2.4). For the moment, inference is conditional on the model and the smoothing parameters. Model selection is described in detail in chapter 3.
For the description of inference methods we consider a structured additive predictor containing several nonlinear components and a parametric part, i.e.

$$\eta = f_1(x_1) + \ldots + f_q(x_q) + \gamma' u.$$

Due to the general representation of nonlinear functions, we don't need to distinguish between different functions here. Estimators for the regression coefficients are obtained by maximising the penalised log–likelihood which takes the form (using scale parameter ϕ)

$$l_{pen}(y|\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_q, \gamma) = \phi \cdot l(y|\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_q, \gamma) - \frac{1}{2}\sum_{j=1}^{q} \lambda_j \boldsymbol{\beta}_j' \mathbf{P}_j \boldsymbol{\beta}_j,$$

where the individual penalty terms are added to an overall penalty.
In the following sections we will describe algorithms for maximising this penalised log–likelihood: in section 2.3.2 for the case of a Gaussian response variable and in section 2.3.3 for the more general case of a response variable belonging to an exponential family. Additionally, we will describe the most important features of generalised regression models and give some examples of exponential families in this section. But first we start with some details regarding the identifiability in structured additive predictors.

2.3.1 Identifiability problems in structured additive predictors

An important issue is the identifiability of the individual nonlinear functions. In most cases there exist no unique solutions for the functions f_j, i.e. the functions are not identifiable. Hence, additional constraints have to be imposed to guarantee identifiabilty. The following sections describe different kinds of predictors together with their specific identifiability problems.

2.3.1.1 Identifiability of univariate functions

Suppose, we have an additive predictor only containing univariate functions of any of the possibilities described in sections 2.2.3–2.2.5 (i.e. the functions can be any nonlinear function apart from i.i.d. Gaussian random effects). In this case, we could change some of the functions by adding or subtracting constant terms without changing the predictor, for example

$$\eta = f_1(x_1) + \ldots + f_q(x_q) + \gamma' u = (f_1(x_1) + c) + \ldots + (f_q(x_q) - c) + \gamma' u. \tag{2.38}$$

That means, only the shape of the individual functions f_j is uniquely determined but not their absolute level. This difficulty is due to the fact that every type of nonlinear function, apart from i.i.d. Gaussian random effects, includes an unpenalised constant term. In other words: every function contains its own intercept. Whether a function includes an unpenalised constant term or not can be detected by looking at the null space of its penalty matrix: If the null space contains constant functions, the respective function includes an unpenalised intercept term. This is true for all univariate functions introduced in the last section with the only exception of i.i.d. Gaussian random effects. As shown in predictor (2.38) above, these constant terms can be shifted either between two functions or between a function and the overall intercept γ_0.
In order to overcome this identifiability problem, additional constraints are imposed on the functions so that their level becomes unique too. We use the following constraints: For random walks of first or second order, Markov random fields or seasonal components, we assume

$$\bar{f} = \frac{1}{p}\sum_{k=1}^{p} f_k = \frac{1}{p}\sum_{k=1}^{p} \beta_k = 0,$$

whereas for P–splines of first or second order we assume (compare Lang & Brezger (2004))

$$\bar{f} = \frac{1}{x_{max} - x_{min}} \int_{x_{min}}^{x_{max}} f(x)\, dx = 0.$$

A two–dimensional P–spline used as surface estimator for a two–dimensional covariate also contains its own intercept and is treated like any of the univariate functions. In this case, we have to deal with the same identifiability problem as described above and the constraint used here is (compare Lang & Brezger (2004))

$$\bar{f} = \frac{1}{(x_{1,max} - x_{1,min})(x_{2,max} - x_{2,min})} \int_{x_{2,min}}^{x_{2,max}} \int_{x_{1,min}}^{x_{1,max}} f(x)\, dx_1 dx_2 = 0.$$

These additional constraints are fulfilled through the centering of each function and by adding the values $\bar{f}_j$ to the overall intercept γ_0. Then the identifiable predictor is given by

$$\begin{aligned} \eta &= (f_1(x_1) - \bar{f}_1) + \ldots + (f_q(x_q) - \bar{f}_q) + \gamma' u + \sum_{j=1}^{q} \bar{f}_j \\ &= f_1^{(c)}(x_1) + \ldots + f_q^{(c)}(x_q) + \gamma' u + \sum_{j=1}^{q} \bar{f}_j, \end{aligned} \tag{2.39}$$

where the functions $f^{(c)}j(x_j)$ are uniquely determined.

2.3.1.2 Identifiability in ANOVA type interaction models

More complex identifiability problems arise in models including interactions between several covariates. In ANOVA type interaction models including a complex interaction and the respective main effects, i.e. in predictor

$$\eta = \gamma_0 + f_1(x_1) + f_2(x_2) + f_{inter}(x_1, x_2),$$

it is principally possible to shift functions of x_1 or x_2 between the interaction and the respective main effect. For example the predictor above is equal to

$$\eta = \gamma_0 + (f_1(x_1) + g(x_1)) + f_2(x_2) + (f_{inter}(x_1, x_2) - g(x_1)).$$

In this thesis, we use the approach described in section 2.2.8.2 for the estimation of this kind of interaction. In this case, both main effects and interaction are uniquely determined regarding this identifiability problem and no further constraints are necessary than those already imposed in section 2.2.8.2.
Note, that it is not possible to estimate several overlapping interactions by this approach. This is due to identifiability problems between the two–dimensional functions, e.g.

$$\begin{aligned} \eta &= \gamma_0 + f_1(x_1) + f_2(x_2) + f_3(x_3) + f_{inter}(x_1, x_2) + f_{inter}(x_1, x_3) \\ &= \gamma_0 + f_1(x_1) + f_2(x_2) + f_3(x_3) + (f_{inter}(x_1, x_2) + g(x_1)) + (f_{inter}(x_1, x_3) - g(x_1)). \end{aligned}$$

The estimation of this predictor would require additional identifiability constraints.

2.3.1.3 Identifiability in varying coefficient models

In a predictor including several varying coefficients which modify the effect of the same interacting variable we have similar identifiability problems as in ANOVA type interaction models. For example in predictor

$$\eta = \gamma_0 + g_1(v_1)z + \ldots + g_q(v_q)z,$$

it is possible to shift linear effects between two varying coefficients, i.e.

$$\eta = \gamma_0 + g_1(v_1)z + \ldots + g_q(v_q)z = \gamma_0 + (g_1(v_1) - c)z + \ldots + (g_q(v_q) + c)z. \tag{2.40}$$

Hence, all modifying functions g_j have to be centered using the respective constraint described in section 2.3.1.1. The values $\bar{g}_j$ from the centering are then collected in a linear effect for variable z that has to be additionally included to the predictor, i.e. predictor (2.40) changes to the identifiable predictor

$$\eta = \gamma_0 + (g_1(v_1) - \bar{g}_1)z + \ldots + (g_q(v_q) - \bar{g}_q)z + \left(\sum_{j=1}^{q} \bar{g}_j\right) z. \tag{2.41}$$

A further kind of varying coefficient model often used is

$$\eta = \gamma_0 + f(v) + g_1(v)z_1 + \ldots + g_k(v)z_k,$$

where variables $z_1, \ldots, z_k$ represent a $k+1$–categorical variable z. This predictor makes it possible to estimate separate effects for the categories of z with $f(v)$ representing either the effect of the reference category (dummy–coding) or an average effect (effect–coding). Identifiability problems arise if the range of variable v differs between the categories. This problem affects here only ranges of values that were not observed for all categories. Hence, this predictor should be used merely if all categories have largely the same range of values for v. If this is not fulfilled, the predictor

$$\eta = \gamma_0 + f(v)z_{k+1} + g_1(v)z_1 + \ldots + g_k(v)z_k,$$

together with dummy–coded variables z_j can be used instead.
Furthermore, for varying coefficient models of the kind described above, i.e.

$$\eta = \gamma_0 + f(v) + g(v)z,$$

the convergence of the iterative estimation algorithm (described in the next section) improves considerably if the (continuous) interacting variable z is centered around zero. The iterative estimation algorithm estimates both functions $f(v)$ and $g(v)$ alternately and its

performance decreases with an increasing degree of dependency between the two functions. The centering of z causes a reduction of the dependency between main effect $f(v)$ and varying coefficient $g(v)$. We want to illustrate this here: In case of a Gaussian response, the covariance matrix of the joint parameter vector $\boldsymbol{\beta} = (\boldsymbol{\beta}_f', \boldsymbol{\beta}_g')'$ is given by

$$\text{Cov}(\boldsymbol{\beta}) = \sigma^2 \begin{pmatrix} \mathbf{V}'\mathbf{V} & \mathbf{V}'\mathbf{Z}\mathbf{V} \\ \mathbf{V}'\mathbf{Z}\mathbf{V} & \mathbf{V}'\mathbf{V} \end{pmatrix}^{-1}$$

Suppose, design matrix $\mathbf{V}$ is a 0/1–incidence matrix (what applies to many functions) whereas matrix $\mathbf{Z} = \text{diag}(z_1, \ldots, z_n)$ contains the observations of the interacting variable. In this case, matrix $\mathbf{V}'\mathbf{Z}\mathbf{V} = \text{diag}\left(\sum_{1 \leq i \leq n: v_{i1}=1} z_i, \ldots, \sum_{1 \leq i \leq n: v_{ip}=1} z_i, \right)$ contains all pairwise correlations between parameters of the two functions. The absolute value of the sum of individual correlations is for a centered variable equal to zero indicating that this number has to be larger for a non–centered variable, i.e.

$$\left| \sum_{k=1}^{p} \sum_{i: v_{ik}=1} (z_i - \bar{z}) \right| = \left| \sum_{i=1}^{n} (z_i - \bar{z}) \right| = 0 \leq \left| \sum_{i=1}^{n} z_i \right| = \left| \sum_{k=1}^{p} \sum_{i: v_{ik}=1} z_i \right|.$$

This implies that a centered function leads to the minimal possible overall dependency between both functions. For a categorical variable z similar facts apply: Here, z is represented by k dummy or effect variables and a centering of these variables is not common. However, effect coding mostly reduces the dependency of $f(v)$ and $g(v)$ compared to dummy coding. With effect coding $f(v)$ represents the average effect of the categories rather than the average over of all observations. Nevertheless, with effect coding $f(v)$ is mostly nearer to the average of all observations than with dummy coding where $f(v)$ represents the effect of one category.

2.3.2 Gaussian Response

In this section we consider models with a Gaussian distributed response y, i.e.

$$y_i = \eta_i + \varepsilon_i,$$

for $i = 1, \ldots, n$, with independently distributed errors ε_i. In most cases, the errors are assumed to have the same distribution $N(0, \sigma^2)$ but it is also possible to deal with heteroscedastic error terms with distributions $N(0, \sigma^2/w_i)$. Conditional on covariates and parameters, the observations y_i are independent and $N(\eta_i, \sigma^2)$– or $N(\eta_i, \sigma^2/w_i)$–distributed. In both cases, the maximum of the penalised log–likelihood is equivalent to the minimum of a penalised residual sum of squares

$$\text{RSS}_{pen}(y|\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_q, \boldsymbol{\gamma}) = (y - \eta)' \mathbf{W} (y - \eta)' + \sum_{j=1}^{q} \lambda_j \cdot \boldsymbol{\beta}_j' \mathbf{P}_j \boldsymbol{\beta}_j,$$

where $\mathbf{W} = \mathbf{I}$ for homoscedastic errors or $\mathbf{W} = \text{diag}(w_1, \ldots, w_n)$ otherwise.
One approach capable of estimating additive predictors with different components is the backfitting algorithm described by Hastie & Tibshirani (1990). It works as follows:

Backfitting Algorithm

1. Initialisation:
   Set $\hat{\boldsymbol{\gamma}}^{(0)} = 0$ and $\hat{\boldsymbol{\beta}}_j^{(0)} = 0$ for $j = 1, \ldots, q$. Set $r = 1$.

2. Compute

$$\hat{\boldsymbol{\gamma}}^{(r)} = (\mathbf{U}'\mathbf{W}\mathbf{U})^{-1}\mathbf{U}'\mathbf{W}\left(y - \sum_{j=1}^{q} \hat{f}_j^{(r-1)}\right)$$

   and for $j = 1, \ldots, q$:

$$\hat{\boldsymbol{\beta}}_j^{(r)} = (\mathbf{X}'_j\mathbf{W}\mathbf{X}_j + \lambda_j\mathbf{P}_j)^{-1}\mathbf{X}'_j\mathbf{W}\left(y - \mathbf{U}\hat{\boldsymbol{\gamma}}^{(r)} - \sum_{k=1}^{j-1} \hat{f}_k^{(r)} - \sum_{k=j+1}^{q} \hat{f}_k^{(r-1)}\right).$$

3. Centering of the nonlinear functions $\hat{f}_j^{(r)} = \mathbf{X}_j\hat{\boldsymbol{\beta}}_j^{(r)}$ for $j = 1, \ldots, q$:

$$\hat{f}_j^{(c,r)} = \hat{f}_j^{(r)} - \bar{f}_j^{(r)}$$

   and adjustment of the intercept term, i.e.

$$\hat{\gamma_0}^{(r)} = \hat{\gamma_0}^{(r)} + \sum_{j=1}^{q} \bar{f}_j^{(r)}$$

   or of the common linear effect for varying coefficients.
   Set $r = r + 1$.

4. Repeating 2. and 3. until there are no changes in the estimated parameters.

Remarks concerning the convergence of the backfitting algorithm can also be found in Hastie & Tibshirani (1990). Usually, with linear smoothers as those described in this thesis, the algorithm converges.
The algorithm is built modular insofar as all functions are estimated separately and alternately. This allows to utilise the sparse structure of design and penalty matrices of the nonlinear functions for an efficient computation (see Rue (2001) and George & Liu (1981)). Alternatively, in a Gaussian model all coefficients could be estimated simultaneously without an iterative algorithm. However, this approach has the disadvantage that the sparse structures of penalty and design matrices get lost. Moreover, identifiability constraints have to be imposed on the overall design matrix to guarantee that the matrix is of full

rank.
The backfitting algorithm is based on the fact that the expected value of the posterior distribution for one set of parameters $\boldsymbol{\beta}_j$ given the data and all other parameters is

$$\mathrm{E}(\boldsymbol{\beta}_j|y, \beta_k, k \neq j, \boldsymbol{\gamma}) = (\mathbf{X}'_j\mathbf{W}\mathbf{X}_j + \lambda_j\mathbf{P}_j)^{-1}\mathbf{X}'_j\mathbf{W}\left(y - \mathbf{U}\boldsymbol{\gamma} - \sum_{k,k\neq j} f_k\right).$$

This relationship is also true for the estimated parameters after convergence of the backfitting algorithm. The part $\left(y - \mathbf{U}\boldsymbol{\gamma} - \sum_{k,k\neq j} f_j\right)$ serves as vector of response values during the progression of the algorithm. Its elements are called partial residuals.
A Bayesian approach based on backfitting for estimating the entire posterior distribution rather than merely the posterior mode was presented by Hastie & Tibshirani (2000).
Based on the estimates of the response, the variance parameter σ^2 can be estimated using formula

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} w_i(y_i - \hat{y}_i)^2, \tag{2.42}$$

which is the maximum likelihood estimate for σ^2. This estimator is only asymptotically unbiased. An unbiased estimator corrects the factor $1/n$ with the number of estimated parameters.

2.3.3 Response of an univariate exponential family

Now, we consider models with an univariate response variable belonging to an exponential family. Examples are count data or binary response variables. These models in combination with a linear predictor are called generalised linear models (see e.g. McCullagh & Nelder (1989)). Here again, like in Gaussian models, it is possible to replace the linear predictor with a structured additive predictor (2.4) leading to generalised STAR models.
Before we will describe the estimation of regression coefficients in section 2.3.3.2, we will introduce some facts about model specification.

2.3.3.1 Model specification

Here, we sketch the most important facts about model specification in generalised regression models. More details about model specification and estimation can be found in Fahrmeir & Tutz (2001) for instance. In generalised regression models, model specification is based on two different assumptions. This fact results in several possible models for the same data even when using the same predictor. The two assumptions are:

1. **Distributional assumption**

   Given the predictor values η_i, $i = 1, \ldots, n$, the response values y_i are conditionally independent and their distributions belong to an exponential family, i.e. the respective density can be written as

$$f(y_i|\theta_i, \phi, w_i) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{\phi} w_i + c(y_i, \phi, w_i)\right\},$$

   where

θ_i	is the natural parameter of the exponential family,
ϕ	is a scale or dispersion parameter common to all observations,
w_i	is a weight and
$b(.)$ and $c(.)$	are specific functions depending on the particular exponential family.

2. **Structural assumption**

   The (conditional) expectation $\mu_i = \mathrm{E}(y_i|\eta_i)$ is related to the predictor η_i by

$$\mu_i = h(\eta_i) \text{ or } \eta_i = g(\mu_i),$$

   where

h	is a known bijective, sufficiently smooth response function and
g	is the inverse of h, called link function.

The natural parameter θ is a function of the mean μ and is for every exponential family uniquely determined by the relation

$$\mu = b'(\theta) = \frac{\partial b(\theta)}{\partial \theta}.$$

For a single observation, we have the relation $\theta_i = \theta(\mu_i)$. The natural parameter provides a special kind of link function, the natural link function. Here, the natural parameter is directly linked to the predictor, i.e.

$$\theta = \theta(\mu) = \eta.$$

The variance of the observations y_i is of the form

$$\mathrm{Var}(y_i|\eta_i) = \frac{\phi v(\mu_i)}{w_i},$$

where the variance function is also for every exponential family uniquely determined by

$$v(\mu) = b''(\theta) = \frac{\partial^2 b(\theta)}{\partial \theta^2}.$$

Distribution	Notation	$\theta(\mu)$	$b(\theta)$	ϕ	$b'(\theta)$	$b''(\theta)$
Normal	$N(\mu, \sigma^2)$	μ	$\theta^2/2$	σ^2	$\mu = \theta$	1
Bernoulli	$B(1, \pi)$	$\log(\pi/(1-\pi))$	$\log(1+\exp(\theta))$	1	$\pi = \frac{\exp(\theta)}{1+\exp(\theta)}$	$\pi(1-\pi)$
Poisson	$Po(\lambda)$	$\log(\lambda)$	$\exp(\theta)$	1	$\lambda = \exp(\theta)$	λ
Gamma	$G(\mu, \nu)$	$-1/\mu$	$-\log(-\theta)$	ν^{-1}	$\mu = -1/\theta$	μ^2

Table 2.1: Important quantities of some exponential families.

Important quantities of some exponential families, like e.g. the natural parameter and the variance function, are shown in table 2.1.
As already mentioned above, the same distributional assumption together with different choices for the response function in the structural assumption leads to several possible models for the same data. The following passages describe frequently used response functions for different types of dependent variables.

- **Normal distribution**

  The normal distribution is also an exponential family. When using the natural link function $\theta(\mu) = \mu$, the response function is simply the identity $h(\eta) = \eta$ and we get back to the classical linear (or STAR) model as in section 2.3.2.

- **Bernoulli and binomial distribution**

  First, we consider the case of ungrouped data with a binary response coded by 0 and 1. Here the expected value is the probability for observing the value 1, i.e. $\mathrm{E}(y_i|\eta_i) = P(y_i = 1|\eta_i) = \pi_i$. The natural link function is

  $$\theta = \log\left(\frac{\pi}{1-\pi}\right) = \eta$$

  with the logistic distribution function

  $$\pi = h(\eta) = \frac{\exp(\eta)}{1+\exp(\eta)}$$

  as resulting response function. Applying a distribution function on the predictor η ensures that the probability π lies in the interval $[0; 1]$. The model using the natural link function is called the logit model.
  Another possible choice for the response function is the standard normal distribution function, i.e.

  $$\pi = h(\eta) = \Phi(\eta).$$

  This model is called the probit model. There exist further possibilities for choosing the response function. Here we have restricted to the ones mentioned above.

If we can group the data, i.e. if there are several independent trials for every combination of covariates, we get $y_i \sim B(m_i, \pi_i)$ with $i = 1, \ldots, n$. In this case, the relative frequencies $\bar{y}_i = y_i/m_i$ are used as dependent variable leading to a scaled binomial distribution with $\mathrm{E}(\bar{y}_i) = \pi_i$. By defining weights $w_i = m_i$ for $i = 1, \ldots, n$, both logit and probit model can also be used for grouped data.

- **Count data**

  Here, we assume to have a Poisson distributed response variable, i.e. $y \sim Po(\lambda)$. In this case the most natural choice is using the natural link and the respective response function which are given by

  $$g(\lambda) = \log(\lambda) = \eta \text{ and } h(\eta) = \exp(\eta) = \lambda.$$

  This ensures a positive value for the mean λ. The model in combination with a simple linear predictor is often called a loglinear model.

- **Gamma distribution**

  Here, we deal with a nonnegative continuous response variable that usually has an asymmetric distribution. One possible model for these data is the lognormal model where the identity link of the normal model is replaced by a log link. The other possibility is to assume a distribution that by definition only has the support $\mathbb{R}_+$, e.g. the gamma distribution. Additionally, the gamma distribution has the property that it includes asymmetric distributions. The most common choice for the structural assumption, that we also use, is the log link

  $$g(\mu) = \log(\mu) = \eta$$

  with the respective response function

  $$h(\eta) = \exp(\eta) = \mu.$$

  This choice ensures a nonnegative value for μ. This, however, is not ensured when using the natural response function

  $$h(\eta) = -\eta^{-1} = \mu.$$

  Note that for the notation used here the gamma distribution is parameterised as follows:

  $$f(y|\mu, \nu) = \frac{1}{\Gamma(\nu)} \left(\frac{\nu}{\mu}\right)^{\nu} y^{\nu-1} \exp\left(-\frac{\nu}{\mu} y\right),$$

  with $\mathrm{E}(y) = \mu$ and $\mathrm{Var}(y) = \mu^2/\nu$.

2.3.3.2 Inference

As the system of estimation equations is nonlinear for generalised models, it is no longer possible to calculate maximum likelihood estimates for the coefficients in the linear predictor analytically. And the analytical calculation of posterior mode or maximum penalised likelihood estimates in a structured additive predictor is not possible, either. Instead, we need iterative algorithms. First, we want to describe the IWLS algorithm for computing maximum likelihood estimates $\hat{\gamma}$ in a linear predictor without penalisation. IWLS is short for iteratively weighted least squares. In every iteration, weighted least squares estimates are calculated where the weights and the dependent variable are adjusted with respect to the current estimates of γ.

IWLS Algorithm

1. Initialisation:
   Set (e.g.) $\hat{\gamma}^{(0)} = 0$. Set $r = 1$.

2. Computation of weight matrix and dependent variable:

$$\begin{aligned}
\mathbf{W}^{(r-1)} &= \text{diag}(d_1^{(r-1)}, \ldots, d_n^{(r-1)}) \\
\eta_i^{(r-1)} &= u_i' \hat{\boldsymbol{\gamma}}^{(r-1)} \\
\mu_i^{(r-1)} &= h\left(u_i' \hat{\boldsymbol{\gamma}}^{(r-1)}\right) \\
\theta_i^{(r-1)} &= \theta\left(h\left(u_i' \hat{\boldsymbol{\gamma}}^{(r-1)}\right)\right) \\
d_i^{(r-1)} &= w_i \left(\frac{\partial h(\eta_i^{(r-1)})}{\partial \eta}\right)^2 \left(\frac{\partial^2 b(\theta_i^{(r-1)})}{\partial \theta^2}\right)^{-1} \\
\tilde{y}_i^{(r-1)} &= \eta_i^{(r-1)} + \left(\frac{\partial h(\eta_i^{(r-1)})}{\partial \eta}\right)^{-1} (y_i - \mu_i^{(r-1)})
\end{aligned}$$

3. Computation of the weighted least squares estimate

$$\hat{\boldsymbol{\gamma}}^{(r)} = (\mathbf{U}'\mathbf{W}^{(\mathbf{r-1})}\mathbf{U})^{-1}\mathbf{U}'\mathbf{W}^{(\mathbf{r-1})}\tilde{y}^{(r-1)}$$

4. Computation of the stop criterion

$$\frac{||\hat{\boldsymbol{\gamma}}^{(r)} - \hat{\boldsymbol{\gamma}}^{(r-1)}||}{||\hat{\boldsymbol{\gamma}}^{(\mathbf{r-1})}||}.$$

   If the stop criterion is larger than a specified $\varepsilon > 0$, set $r = r + 1$ and go back to 2., otherwise terminate the process.

This algorithm is equivalent to Fisher–Scoring which is a modified Newton–Raphson method. Fisher–Scoring uses the expected Fisher information matrix instead of the matrix containing the second derivatives of the log–likelihood, the observed information matrix. When using the natural link function, expected and observed Fisher information are identical.
If we have a structured additive predictor, i.e. if we want to maximise a penalised log–likelihood, step 3. of the IWLS algorithm is replaced by the backfitting algorithm. This combined algorithm is called Local Scoring Procedure by Hastie & Tibshirani (1990). In fact, it calculates the zero point of the first derivative of the penalised log–likelihood $\partial\, l_{pen}(y|\gamma, \beta_1, \ldots, \beta_q)/\partial\,(\gamma, \beta_1, \ldots, \beta_q)$.
If the scale parameter is unknown, as is the case for a Gamma or normally distributed response, it can be estimated by

$$\hat{\phi} = \frac{1}{n} \sum_{i=1}^{n} w_i \frac{(y_i - \hat{\mu}_i)^2}{v(\hat{\mu}_i)}, \tag{2.43}$$

where $\hat{\mu}_i = h(\hat{\eta}_i)$ and $v(\hat{\mu}_i)$ are the respective mean and variance function of y_i. For a normally distributed response formula (2.43) results in the ML–estimate $\hat{\sigma}^2$ from formula (2.42). In contrast to the usually used estimator (see e.g. Fahrmeir & Tutz (2001)), we do not correct n with the number of model parameters in order to get an estimator analogous to the ML–estimate in the Gaussian case.

Chapter 3

Selection of Variables and Smoothing Parameters

In chapter 2, we already mentioned the influence of the smoothing parameter λ (or equivalently the variance parameter τ^2) on the estimated effect of a covariate (see figure 2.2 for the case of P–splines in section 2.2.3.2). We also described approaches for inference in structured additive models if all smoothing parameters are fixed. In this chapter, we deal with the problem of determining appropriate values for the smoothing parameters. Additionally, we want to deal with a second, but similar, problem: the selection of important variables. This question was not mentioned in the last section. But in many applications, a lot of potentially influential covariates are available although only a few of them actually have an influence on the response. Altogether, there arise the following questions:

- Which terms (covariates) are to be included in the model?
- Is the effect of a certain continuous variable linear or nonlinear, i.e. is it necessary to use a spline function or would a linear effect be sufficient?
- Which value should be used for the smoothing parameter of a nonlinear function?
- Does a nonlinear effect vary over the range of another variable or is the effect constant?
- Is there a complex interaction between two continuous variables?

In this chapter, we want to deal with these questions simultaneously and introduce algorithms that can answer them.
This chapter is organised as follows: The first section 3.1 gives an overview of alternative and related methods for variable and/or smoothing parameter selection. All other sections

explain details that are in close connection with our selection approach: section 3.2 describes several selection criteria, the concept of degrees of freedom in additive models is explained in section 3.3 and the selection algorithms are described in the last section 3.4.

3.1 Alternative Approaches

In the last two decades, considerable research has been carried out on the topic of variable selection and on determining values for smoothing parameters. Nevertheless, most of the existing approaches can either select subsets of variables in (generalised) linear models or can determine smoothing parameters for a fixed set of covariates. Altogehter, none of the approaches introduced in this section can deal with a simultaneous variable and smoothing parameter selection in such a broad class of models as our approach described in section 3.4.

3.1.1 Approaches for variable selection

An overview over methods for subsect selection in (generalised) linear models can be found in Miller (2002) or Kadane & Lazar (2004) for instance. The best known approaches are forward selection and backward elimination. *Forward selection* starts with the empty model containing the intercept term only. Then in each step, the best variable according to a selection criterion (compare subsection 3.2) or a certain test statistic is added to the model (among those that have not been added previously). The algorithm stops when the model is not improved by adding one of the remaining variables.
Unlike forward selection, *backward elimination* starts with the full model containing all variables. At each step, it removes the least important variable from the model basing the decision again either on a selection criterion or on a test statistic. The process stops when the model is not improved by removing one of the remaining variables from the model. These two approaches can be combined leading to stepwise regression (see e.g. Miller (2002)).
Alternative approaches for subset selection in linear models which are closely related to each other are Lasso, forward stagewise regression and LARS (compare Efron, Hastie, Johnstone & Tibshirani (2004)). For all three approaches we assume that the response variable and all covariates are centered around zero and that the covariates are additionally standardised.
Lasso was introduced by Tibshirani (1996) and estimates the regression coefficients by minimising the residual sum of squares subject to the condition that the sum of absolute

coefficient values is smaller than a certain threshold value, i.e.

$$\sum_{j=1}^{p} |\beta_j| \leq t$$

This threshold value t serves as a tuning parameter and has to be determined appropriately, e.g. using cross validation. If the threshold value is large enough, the estimated coefficients are identical to the usual least squares estimates. In contrast, if the threshold value is small the parameter estimates are shrunken towards zero. Often some of the coefficients are even equal to zero so that the respective covariates can be considered having no effect on the response.

Forward stagewise regression is an iterative method that chooses in each step the covariate x_j with the highest absolute correlation to the current residual vector $r = (y - \hat{\mu})$. Then, the current linear predictor $\hat{\mu}$ is adjusted and replaced by

$$\hat{\mu} + \epsilon \cdot sign(cor(x_j, r))x_j$$

using a small value for the constant ϵ. For $\epsilon = cor(x_j, r)$ this approach is equivalent to the simple forward selection. The starting values for the parameter estimates are zero. Variable selection is included implicitly by not choosing certain covariates during the entire process.

Least Angle Regression (LARS) introduced by Efron, Hastie, Johnstone & Tibshirani (2004) is a modified version of the forward stagewise regression. Similar to the formula for stagewise regression above, the linear predictor is in each step adjusted using the variable with the largest absolute correlation to the current residual vector r. There are two differences to forward stagewise regression: the value ϵ is not fixed but is in each step chosen such that the correlation between the newly adjusted residual vector and the actual chosen variable is as big as the correlation between the predictor and the next best covariate x_k, i.e.

$$|cor[y - (\hat{\mu} + \epsilon \cdot sign(cor(x_j, r)) \cdot x_j), x_j]| = |cor[y - (\hat{\mu} + \epsilon \cdot sign(cor(x_j, r)) \cdot x_j), x_k]|$$

must hold. Out of these two variables a new variable $x_{k'}$ is built such that the angle between the variable vectors x_j and x_k is divided equally by this new variable. The algorithm continues using this artificial variable. Variable selection is again included implicitly by not choosing certain covariates during the entire process. The LARS algorithm can also be modified to provide solutions for Lasso.

Bayesian approaches for model selection can be based on Bayes factors which compare different models (compare Kass & Raftery (1995) or section 3.2.3 of this chapter). Other

Bayesian approaches for subset selection of variables in linear models can be based on indicator variables γ_j for each of the covariates x_j leading to the predictor

$$\eta = \beta_0 + \gamma_1\beta_1 x_1 + \ldots + \gamma_p\beta_p x_p$$

An example is the approach presented by George & McCulloch (1997). They use hierarchical Bayes mixture models in combination with MCMC methods like the Gibbs sampler or the Metropolis–Hastings algorithm (compare Green (2001)) to perform the selection. The lowest level of the hierarchy is represented by the indicator variables γ_j. These are provided independently of each other with prior probabilities $\pi_j = P(\gamma_j = 1)$ indicating the probability that the j-th covariate has an influence on the response. The next level are the prior distributions for the regression parameters conditional on the indicator variables. Here, it is possible to use a normal mixture of the form

$$\beta_j|\gamma_j = (1-\gamma_j)N(0,\tau_{j0}^2) + \gamma_j N(0,\tau_{j1}^2),$$

with a small value for τ_{j0}^2 and a large one for τ_{j1}^2. How to choose the values for the variances is described in George & McCulloch (1997). The parameter τ_{j0}^2 can also be set to zero leading to a point mass on $\beta_j = 0$. This was considered in Geweke (1996). The decision which model to use can be based on the posterior distributions of different models. Alternatively, these approaches also allow the performance of a kind of model averaging (compare chapter 5 of this thesis).
The earlier approach of Mitchell & Beauchamp (1988) works similar. As prior distribution for each regression parameter they choose what they call *slab and spike* distribution: a mixture prior with a point mass at zero and a diffuse uniform distribution elsewhere. This prior depends on the ratio of the probability assigned to zero to the probability assigned to all other values. This ratio has to be chosen by the user, e.g. by using a kind of Bayesian cross validation.

3.1.2 Approaches for determining smoothing parameters

There exists a variety of approaches for determining smoothing parameters in (generalised) additive models or even in (generalised) STAR models. Two methods that can be applied to (generalised) STAR models with as many different possible function types as described in chapter 2 are a fully Bayesian approach using MCMC methods described in Fahrmeir & Lang (2001a), Fahrmeir & Lang (2001b) or Lang & Brezger (2004) and the restricted maximum likelihood (REML) estimation described in Fahrmeir, Kneib & Lang (2004) or Kneib (2006).
In the fully Bayesian approach, the variance parameters τ_j^2 are considered as random and

are therefore each provided with a hyperprior. A common assumption is that all variance parameters are independent and inverse gamma distributed, i.e. $\tau^2 \sim IG(a_j, b_j)$, with fixed parameters a_j and b_j. Possible choices for the parameters would be the same small value for both parameters, like e.g. $a_j = b_j = 0.001$, or alternatively $a_j = 1$ and a small value for b_j, e.g. $b_j = 0.005$ (see Fahrmeir & Lang (2001a) for instance). Considering the variance parameters as random allows to estimate them simultaneously with the regression coefficients. The prior distribution of each set of regression parameters β_j is now considered conditional on the current value of the respective variance parameter τ_j^2. In contrast to (2.7), the posterior distribution of all parameters given the data is now of the form

$$p(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_q, \boldsymbol{\gamma}, \tau_1^2, \ldots, \tau_q^2 | y) \propto L(y|\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_q, \boldsymbol{\gamma}) \prod_{j=1}^{q} (p(\boldsymbol{\beta}_j|\tau_j^2) p(\tau_j^2)). \tag{3.1}$$

The estimation is carried out using either the Gibbs sampler for a Gaussian response (compare Lang & Brezger (2004)) or a Metropolis–Hastings algorithm otherwise (see Brezger & Lang (2006)), where the regression coefficients and variance parameters are updated alternately.

REML estimation is based on the transformation of a STAR model in a (generalised) linear mixed model (GLMM). In doing so, every parameter vector $\boldsymbol{\beta}_j$ is decomposed in its penalised and its unpenalised part, i.e.

$$\boldsymbol{\beta}_j = \mathbf{X}_j^{pen} \boldsymbol{\beta}_j^{pen} + \mathbf{X}_j^{unp} \boldsymbol{\beta}_j^{unp}.$$

The unpenalised part $\boldsymbol{\beta}_j^{unp}$ is the part of $\boldsymbol{\beta}_j$ that is not penalised by the respective penalty matrix and depends on its null space (compare chapter 2), i.e. the length of $\boldsymbol{\beta}_j^{unp}$ corresponds to the dimension of this null space. Accordingly, the length of the penalised vector $\boldsymbol{\beta}_j^{pen}$ is the difference between the number of parameters p_j and the length of $\boldsymbol{\beta}_j^{unp}$. Function f_j can now be decomposed in

$$f_j = \mathbf{X}_j \boldsymbol{\beta}_j = \mathbf{X}_j \mathbf{X}_j^{pen} \boldsymbol{\beta}_j^{pen} + \mathbf{X}_j \mathbf{X}_j^{unp} \boldsymbol{\beta}_j^{unp},$$

where the first unpenalised part contains only fixed effects. The penalty matrix belonging to the new parameter vector $\boldsymbol{\beta}_j^{pen}$ is the identity matrix as is the case for i.i.d. Gaussian random effects. Altogether, the transformed model contains now only fixed effects and random effects. That allows to estimate the variance parameters τ_j^2 with methods developed for mixed models. The regression coefficients and variance parameters are estimated alternately: the regression coefficients through maximisation of the penalised log–likelihood with given variance parameters and the variance parameters through maximisation of a restricted marginal log–likelihood.

A widely known method for selection of smoothing parameters in (generalised) spline models is provided by the R software package *mgcv* described in Wood (2006b), Wood (2000) and Wood (2004). The original approach from Wood (2000) goes back to Gu & Wahba (1991). Here, the algorithm alternates between the determination of an overall smoothing parameter by using a one–dimensional direct search and the selection of the individual relative smoothing parameters of the functions by using Newton updates. The selection is based on the minimisation of a criterion like e.g. GCV. Wood (2004) presents a modified and improved selection method that is numerically more stable and can deal with user–specified, fixed smoothing parameters. Here, the optimisation is carried out using the Newton algorithm where some Newton steps are replaced by steepest descent steps in case the criterion is not locally concave. With both selection methods, penalties combining a difference penalty with a small shrinkage component can be used for the spline functions. The shrinkage component sets a function equal to zero for a large enough value of the smoothing parameter, i.e. if the function is practically completely smooth according to the difference penalty (see Wood (2006b)). For small smoothing parameters, the shrinkage component has hardly any influence on the estimated effect. Hence, an automatic variable selection can be performed by using these shrinkage penalties.
Another approach which is able to determine the degree of smoothness of nonlinear functions is *boosting* (compare Bühlmann (2004) or Bühlmann & Yu (2003) for an overview). Here, starting from the empty model, so called weak learners which are relatively smooth are successively applied to the current residuals $(y - \hat{\eta})$. In each iteration, only the weak learner of one variable is chosen to be added to $\hat{\eta}$. The chosen variable is the one that minimises a selection criterion. If the addition of each of the variables to the predictor increases the selection criterion the process is finished. With boosting, the degree of smoothness of every nonlinear function is controlled by the number of times the respective weak learner is chosen during the process. The nonlinear functions can be of ridge type (see Tutz & Binder (2006)) in which case the approach becomes for Gaussian responses similar to the selection algorithm introduced later in this chapter. Boosting can perform a variable selection implicitly by never choosing the weak learner of a certain function during iterations.
One approach developed for the simultaneous selection of variables and smoothing parameters in additive models is based on *genetic algorithms* and is presented in Krause & Tutz (2004) and Krause & Tutz (2006). The method is based on ideas adopted from biological inheritance: mutation, crossover and selection. Mutation and crossover make sure that the model space is searched thoroughly, whereas selection causes to reject bad models. The selection is once again based on a criterion like e.g. AIC.

3.2 Selection Criteria

In our approach the selection of variables and smoothing parameters is based on selection criteria. There is a wide variety of criteria available. Here, we restrict to some of the most widely used criteria which can be used in combination with our selection algorithms. A detailed overview of this topic can be found in Miller (2002) for instance.

3.2.1 Akaike Information Criterion (AIC)

The Akaike Information Criterion or AIC was originally introduced by Akaike (1973). Burnham & Anderson (1998) or Cavanaugh (1997) derive the AIC from the Kullback–Leibler distance

$$I(f,g) = \int f(y) \ln\left(\frac{f(y)}{g(y|\theta)}\right) dy = \int f(y) \ln(f(y)) dy - \int f(y) \ln(g(y|\theta)) dy, \tag{3.2}$$

that measures the distance between the true, but unknown model f of the dependent variable y and the assumed model g. Often, the model g presents a family of models depending on parameters θ. The smaller the value of $I(f,g)$ the better is the assumed model g. The Kullback–Leibler distance is a directed distance because $I(f,g) \neq I(g,f)$. It is always positive, with the exception of $f \equiv g$ when it is zero. The term $\int f(y) \ln(f(y)) dy$ is unknown because of the unknown function f but it is equal for all models. This means, for the comparison of models the first term in (3.2) can be neglected. The AIC is an estimate for the expectation of the second term, multiplied by two. Therefore, as an estimate of the relative expected Kullback–Leibler distance the AIC has no natural zero. That means, AIC can be used to compare models but gives no evidence of the actual quality of a certain assumed model. The formula for AIC is

$$\text{AIC} = -2 \cdot l(\theta|y) + 2 \cdot p, \tag{3.3}$$

where $l(\theta|y) = \ln(g(y|\theta))$ is the log–likelihood of the model and p is the number of estimated parameters in θ. For selection in linear regression models, the vector θ includes all regression coefficients and possibly a scale parameter (depending on the type of response distribution). By setting a certain coefficient equal to zero, the respective variable is removed from the model and the number of estimated parameters reduced.
In the special case of Gaussian distributed response variables, when the variance σ^2 is also estimated, we get the simplified formula (compare Burnham & Anderson (1998))

$$\text{AIC} = n \log(\hat{\sigma}^2) + 2\,(p+1), \tag{3.4}$$

with p degrees of freedom for the p regression coefficients and one degree of freedom for the variance estimate. The estimate $\hat{\sigma}^2$ is the maximum likelihood estimate $\hat{\sigma}^2 = 1/n$ RSS and depends on the current model. As the variance is estimated in all models, we can use the number of regression parameters p instead of $p + 1$ without influencing the selection.
The two terms included in AIC have contrary effects regarding selection: the negative log–likelihood or the residual sum of squares, respectively, becomes smaller when the model gets more complex and/or more variables are added. In the same case, the value of the second term measuring the complexity of the model increases. The opposite is true for the other way round: the simpler the model, the larger the value of the negative log–likelihood and the smaller the value of the second term. Hence, with these two terms AIC holds the balance between over– and underfitting.
The formula (3.3) mentioned above was developed for maximum likelihood inference, i.e. the assumed models g are likelihood functions. In structured additive models we perform penalised maximum likelihood inference, so that the assumed models g are now penalised likelihoods. In this context, a derivation of an information criterion based on the Kullback–Leibler distance is given by Shibata (1989). He calls the resulting criterion RIC. In this thesis, we will also refer to the criterion as AIC because the general form includes both cases: maximum and penalised maximum likelihood estimation. In the general form the AIC has the formula (compare Hastie & Tibshirani (1990))

$$\text{AIC} = -2\,l(\theta|y) + 2\,\text{tr}(\mathbf{H}) = -2\,l(\theta|y) + 2\,df_{total}, \tag{3.5}$$

where the hat matrix $\mathbf{H}$ is the matrix that projects the data y on the fitted values, i.e. $\hat{y} = \mathbf{H}y$. In the case of a non–Gaussian response, $\mathbf{H}$ is the matrix evaluated at the last iteration of the scoring algorithm, i.e. $\hat{\eta} = \mathbf{H}\tilde{y}$. In the following, we refer to the quantity $df_{total} := \text{tr}(\mathbf{H})$ as degrees of freedom of the model. In maximum likelihood inference the quantity $\text{tr}(\mathbf{H})$ is equal to the number of regression parameters. More details regarding the calculation of degrees of freedom are described in section 3.3 of this chapter.

3.2.2 Improved AIC

The bias–correction term $2df_{total}$ of the AIC is not sufficient if the degrees of freedom are large compared to the number of observations n. In this case, it is better to use a corrected version of AIC, the improved AIC described by Hurvich, Simonoff & Tsai (1998) for the context of smoothing parameter selection. It is developed for Gaussian response variables but can also be used for other response distributions (compare Burnham & Anderson (1998)). In comparison to AIC, the improved AIC contains an additional bias–correction

term:

$$\text{AIC}_{imp} = \text{AIC} + \frac{2df_{total}(df_{total}+1)}{n - df_{total} - 1}. \tag{3.6}$$

Burnham & Anderson (1998) give an approximate rule when the improved AIC should be used: It should be used when the ratio $n/df_{total} < 40$ for the most complex model considered for selection.

3.2.3 Bayesian Information Criterion (BIC)

The Schwarz Criterion or Bayesian Information Criterion (BIC) was derived by Schwarz (1978). A derivation of BIC can also be found in Cavanaugh & Neath (1999). The BIC originates from a Bayesian context. Suppose, we have two different models M_i, $i = 1, 2$, which are assumed with a priori probabilities $p(M_1)$ and $p(M_2)$. The priors for the regression coefficients are in this case defined conditional on the model by $p(\theta_i|M_i)$. With Bayes's theorem one gets the posterior probability for each model by

$$p(M_i|y) = \frac{p(y|M_i)p(M_i)}{p(y|M_1)p(M_1) + p(y|M_2)p(M_2)},$$

where the expression $p(y|M_i)$ is the marginal likelihood of the model and is calculated as

$$p(y|M_i) = \int p(y|\theta_i, M_i)p(\theta_i|M_i)d\theta_i.$$

The term $p(y|\theta_i, M_i)$ is the likelihood function for the parameters θ_i. In order to answer the question which of the two models is superior, one can use the Bayes factor (see Kass & Raftery (1995))

$$B_{12} = \frac{p(y|M_1)}{p(y|M_2)},$$

which supports M_1 if $B_{12} > 1$. In the case of equal prior probabilities $p(M_i)$, the Bayes factor is identical to the ratio of posterior odds. The BIC is a rough approximation to the Bayes factor and allows to avoid the specification of prior probabilities. In certain settings, model selection with BIC is even equal to selection based on bayes factors (see Kass & Raftery (1995) for more details). The formula of BIC is

$$\text{BIC} = -2l(\theta|y) + \log(n) \cdot p, \tag{3.7}$$

where p is again the number of parameters and n the number of observations. BIC has a consistency property: If the candidate models include the true model that generated the

data, BIC will identify this model with probability one for $n \to \infty$.
In the context of structured additive regression models we compare models which have the same number of regression parameters but differ in the amount of smoothness. In order to account for these differences, we again replace the number p with the number df_{total}. This leads to formula

$$\text{BIC} = -2l(\theta|y) + \log(n) \cdot df_{total}. \tag{3.8}$$

3.2.4 Generalised Cross Validation (GCV)

GCV is short for generalised cross–validation and is not an information or likelihood based criterion like the three previous ones. A derivation for normal distributed response can be found in Hastie & Tibshirani (1990) for instance. Suppose we have a model with a normal distributed response only containing one nonlinear function, i.e.

$$y_i = f(x_i) + \varepsilon_i,$$

with i.i.d. error terms $\varepsilon_i \sim N(0, \sigma^2)$. In this case, the hat (or smoother) matrix $\mathbf{H}$ projecting the data y on the fitted values, i.e. $\hat{y} = \mathbf{H}y$, is given by

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda\mathbf{P})^{-1}\mathbf{X}',$$

where $\mathbf{X}$ is the respective design and $\mathbf{P}$ the penalty matrix.
In order to determine an appropriate value for the smoothing parameter λ one can use cross-validation with leaving out one observation at a time. That means, the criterion

$$\text{CV} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_\lambda^{-i}(x_i))^2 \tag{3.9}$$

is minimised over λ, where $\hat{f}_\lambda^{-i}(x_i)$ was estimated without observation (y_i, x_i). Function $\hat{f}_\lambda^{-i}(x_i)$ can be directly calculated from matrix $\mathbf{H}$ without estimating all n different models through

$$\hat{f}_\lambda^{-i}(x_i) = \sum_{j=1, j\neq i}^{n} \frac{\mathbf{H}_{ij}}{1 - \mathbf{H}_{ii}} y_j.$$

Using this relationship, formula (3.9) can be equivalently written as

$$\text{CV} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_\lambda(x_i)}{1 - \mathbf{H}_{ii}}\right)^2.$$

By replacing the diagonal elements $\mathbf{H}_{ii}$ by their average value $\text{tr}(\mathbf{H})/n$ one obtains the generalised cross–validation

$$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_\lambda(x_i)}{1 - \text{tr}(\mathbf{H})/n}\right)^2.$$

In a structured additive model with several terms and with possibly heteroscedastic errors the more general formula is

$$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n} w_i\left(\frac{y_i - \hat{\eta}_i}{1 - \text{tr}(\mathbf{H})/n}\right)^2, \tag{3.10}$$

where $\mathbf{H}$ again represents the hat matrix for the entire model and additionally includes weight matrix $\mathbf{W}$ (compare section 3.3).
In the case of a non–normal response, GCV can be adapted by using the residual sum of squares based on the last step of the scoring algorithm (see Wood (2006a)):

$$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n} d_i\left(\frac{(\tilde{y}_i - \hat{\eta}_i)^2}{1 - \text{tr}(\mathbf{H})/n}\right)^2, \tag{3.11}$$

with IWLS–weights d_i and working response $\tilde{y}_i$.
Alternatively, GCV can be adapted using residuals appropriate for the respective context. One possibility is to use squared Pearson residuals (see e.g. Fahrmeir & Tutz (2001)). Another possibility is to use deviance residuals (see Hastie & Tibshirani (1990)) which lead to

$$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n}\frac{D_i(\hat{\mu}_i|y_i)}{(1 - \text{tr}(\mathbf{H})/n)^2}, \tag{3.12}$$

where the deviance is defined as (see Fahrmeir & Tutz (2001))

$$D_i(\hat{\mu}_i|y_i) = -2\phi\sum_{i=1}^{n} l_i(\hat{\mu}_i|y_i) - l_i(\hat{\mu}_{max,i}|y_i).$$

The term $l_i(\hat{\mu}_{max,i}|y_i)$ denotes the biggest possible value resulting from the saturated model. Often, all observations have different values in the covariates and $\hat{\mu}_{max,i} = y_i$. If several observations have exactly the same values in all covariates, $\hat{\mu}_{max,i}$ is the mean of the respective response values.
For the selection algorithms described in this chapter it is possible to use either GCV from formula (3.11) or the one from formula (3.12).
A modified version of GCV selecting more parsimonious models is introduced in Kim & Gu

(2004). Here, the degrees of freedom of the model are multiplied by an additional factor $\alpha > 1$ which results for Gaussian response in formula

$$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n} w_i \left(\frac{y_i - \hat{\eta}_i}{1 - \alpha \cdot \text{tr}(\mathbf{H})/n}\right)^2. \tag{3.13}$$

For non–Gaussian response variables formulas (3.11) or (3.12) are changed accordingly. Based on the results of simulation studies, Kim & Gu (2004) suggest to choose a value in the range $[1.2, 1.4]$ for α.
In the case of a normal distributed response variable, each of the four criteria AIC, AIC_{imp}, BIC and GCV can be brought into the general form

$$\text{criterion} = \log(\hat{\sigma}^2) + \psi(df_{total}), \tag{3.14}$$

where the function ψ indicates a penalty term. Table 3.1 gives an overview of the functions ψ and figure 3.1 illustrates the resulting curves in dependence on the ratio of the degrees of freedom to the number of observations. This helps to explain the different performance of the criteria. BIC has a strong penalty that is outdone by AIC_{imp} and GCV only if the ratio df/n is near one. AIC_{imp} and GCV both have nonlinear penalties increasing more strongly for high values of df/n, where AIC_{imp} always has the stronger penalty. For a small ratio $df/n < 0.2$, or alternatively for $n \to \infty$, AIC, AIC_{imp} and GCV are almost equivalent.

Criterion	Penalty ψ
AIC	$2df/n$
AIC_{imp}	$2df/n + 2df(df+1)/(n(n-df-1))$
BIC	$\log(n)df/n$
GCV	$-2\log(1-df/n)$

Table 3.1: Penalty functions ψ for the selection criteria AIC, AIC_{imp}, BIC and GCV.

3.2.5 Mean Squared Error of Prediction (MSEP)

Both GCV and AIC can be seen as estimates for the error of prediction when using the log–likelihood or the residual sum of squares as loss–function (compare Hastie, Tibshirani & Friedman (2001)). Considering the normal–response model

$$y_i = f(x_i) + \varepsilon_i$$

with $i = 1, \ldots, n$ again, both GCV and AIC are estimates of the following global prediction–oriented measure:

$$\text{MSEP} = \frac{1}{n}\sum_{i=1}^{n} E(Y_i^* - \hat{f}(x_i))^2, \tag{3.15}$$

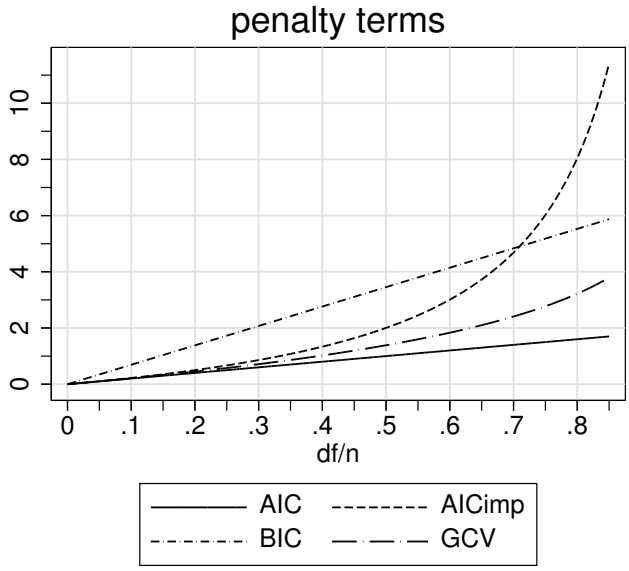

Figure 3.1: Penalty terms $\psi(df)$ for the selection criteria AIC, AIC_{imp}, BIC and GCV in dependence on the ratio df/n. Here, $n = 1000$ is used for the number of observations.

where Y_i^* are new, independent observations at covariate values x_i.
A different approach to estimate this MSEP (mean squared error of prediction) is by splitting the data into two parts (see Hastie, Tibshirani & Friedman (2001)): a training set and a test set. The training set is used to calculate the parameter estimates whereas the observations in the test set represent the new observations Y_i^* and are used to calculate MSEP. Suppose, we have m observations in the test set, then MSEP can be estimated using the formula

$$\text{MSEP} = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{f}^{training}(x_i))^2. \tag{3.16}$$

In the case of non–normal response variables, the residual sum of squares is replaced by the deviance resulting in

$$\text{MSEP} = \frac{1}{m} \sum_{i=1}^{m} D_i(\hat{\mu}_i | y_i). \tag{3.17}$$

The split–up of the original data set is carried out randomly with a few restrictions due to specific features of structured additive models:
For P–splines and one–dimensional random walks, the basis functions are defined locally on the range between minimum and maximum $[x_{min}, x_{max}]$ (compare section 2.2.3). This complicates the prediction of function evaluations $f(x)$ for values of x outside the interval

$[x_{min}, x_{max}]$. To overcome this problem, we make sure that both values x_{min} and x_{max} are in the training set.
Another problem occurs with Markov random fields and/or random effects. Here, one regression parameter is estimated for every region or group. In the case of random effects, it is only possible to estimate the parameter of a certain group if this group is represented in the training set. Therefore, the training set contains at least one observation of each group. A similar problem arises with Markov random fields: In principle, it is possible to estimate a parameter for a region without observations by the average of all neighbours. However, leaving out one region in the training set changes the neighbourhood structure in comparison to the complete data set and therefore the training set contains at least one observation of every region.
The split–up of the original data requires a relatively large number of observations. But unlike the previously described criteria, MSEP does not require the calculation of the degrees of freedom of the models.

3.2.6 Cross Validation

Like MSEP, cross validation is a direct estimate for prediction–error as defined by formula (3.15). And similarly, the original data is split up in several parts. With our algorithms, 5–fold and 10–fold cross validation is available, i.e. the data set is split into five or ten parts, respectively. But generally, every number up to the number of observations n, resulting in leave–one–out cross–validation (compare the section about GCV), is possible for the number of different parts. The split–up of the original data is carried out randomly in such a way that the resulting parts are disjunct. That means, every observation is contained in only one part. As far as possible, each part gets the same number of observations.
Let $m = 5, 10$ denote the number of parts and n_i the number of observations for part i. Then, the criteria CV_5 or CV_{10} can be calculated as

$$\mathrm{CV}_m = \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n_i} (y_{ij} - \hat{f}^{-i}(x_{ij}))^2, \tag{3.18}$$

where $\hat{f}^{-i}(x_{i,j})$ is the estimate without using the i–th part. So, the estimation is always carried out by using $m - 1$ parts whereas validation is performed by using the omitted part. This is repeated m times by always omitting another part. In the case of non–normal response, we again replace the residual sum of squares by the deviance

$$\mathrm{CV}_m = \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n_i} D_{ij}^{-i}(\hat{\mu}_{ij}|y_{ij}). \tag{3.19}$$

Like MSEP, cross validation has the advantage that the calculation of the degrees of freedom of the models is not necessary. But it also requires a relatively large number of observations for the partitioning of the data. The restrictions imposed for choosing the training set for MSEP cannot be observed here because of the split–up in m disjunct parts. In contrast to MSEP, cross validation has a high computational effort because all models have to be estimated five or ten times, respectively.
The estimation of models based on the m training sets (each consisting of $m-1$ parts) is carried out by defining m weight variables where the weights are set to zero if the respective observation is not in the training set. This allows to perform estimation for different training sets by only changing the weight variables without having to define new design matrices. The calculation of MSEP is handled in a similar way based on one weight variable with zero entries indicating the observations from the test set.

3.3 Degrees of freedom in STAR models

In the previous section, we already mentioned the concept of degrees of freedom of a model. The calculation of degrees of freedom is required with four of the selection criteria (AIC, AIC_{imp}, BIC and GCV) in order to account for the complexity of a model. In this section, we will describe a few details regarding this number.
The degrees of freedom, in the context of additive models sometimes alternatively called equivalent degrees of freedom, are calculated by

$$df_{total} = \text{tr}(\mathbf{H}),$$

where the so–called hat matrix $\mathbf{H}$ projects the response y on the fitted values $\hat{y}$, i.e. $\hat{y} = \mathbf{H}y$. In the case of a non–Gaussian response, $\mathbf{H}$ is evaluated at the last iteration of the scoring algorithm, i.e. $\hat{\eta} = \mathbf{H}\tilde{y}$.
There are two special cases in which the calculation of $\text{tr}(\mathbf{H})$ is simple: in the case of a linear model the trace of $\mathbf{H}$ is equal to the number of regression coefficients p. In the case of a simple model merely containing one non–linear function, $\text{tr}(\mathbf{H})$ can be calculated directly through

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda\mathbf{P})^{-1}\mathbf{X}'\mathbf{W}, \tag{3.20}$$

with design matrix $\mathbf{X}$, penalty matrix $\mathbf{P}$ and weight matrix $\mathbf{W}$ containing either weights for a weighted Gaussian regression or for the last iteration of the IWLS algorithm. For non–linear functions of continuous covariates, matrix $\mathbf{H}$ is often called smoother matrix and the number $\text{tr}(\mathbf{H})$ equivalent degrees of freedom of the smoother (see Hastie & Tibshirani

(1990)). In the following, we will refer to the number tr($\mathbf{H}$) simply as degrees of freedom. The degrees of freedom are always positive and depend on the value of the smoothing parameter. The relationship between smoothing parameter and degrees of freedom is inverse: A large (small) smoothing parameter results in small (large) degrees of freedom.
In the following, we will use the term *smoother matrix* to denote matrices of the form (3.20) regardless of the type of nonlinear function used. Furthermore, we consider the parametric part of the predictor as one single linear function with an appropriate, unpenalised smoother matrix of the form (3.20), i.e. $\mathbf{P} = \mathbf{0}$.
More difficult than in the simple cases mentioned above is the calculation of tr($\mathbf{H}$) in a (structured) additive model with several non–linear functions or with a non–linear function in combination with categorical covariates. In this case, the hat matrix $\mathbf{H}$ containing entries for all regressors is unknown and so, of course, is tr($\mathbf{H}$). The reason is, that for estimation performed by backfitting algorithm (see section 2.3.2) or local scoring procedure (see section 2.3.3) the complete hat matrix is not needed. The estimation is carried out iteratively using only the individual smoother matrices $\mathbf{H}_j$ of the respective functions f_j. Additionally, building up the complete hat matrix $\mathbf{H}$ in structured additive models is often computationally very expensive. The inversion of a $p \times p$ matrix is necessary, where p is the total number of parameters. For a spatial function, for instance, the number of basis functions is equal to the number of regions and can easily amount to a few hundred. To overcome the problem of the unknown hat matrix, the degrees of freedom of the model are approximated by the sum of individual degrees of freedom (see Hastie & Tibshirani (1990)), i.e.

$$df_{total} = \sum_j df_j. \tag{3.21}$$

In the case of most non–linear functions individual degrees of freedom are calculated by

$$df_j = \text{tr}(\mathbf{H}_j) - 1, \tag{3.22}$$

where the subtracting of 1 accounts for the centering with respect to the intercept term in case of an univariate nonlinear function or with respect to the common linear effect in case of a varying coefficient. The value df_j lies in the range $[d_j - 1; p_j - 1]$ where d_j is the dimension of the null space of the respective penalty matrix or equivalently the rank deficiency. The number p_j indicates the number of regression coefficients.
The approximate degrees of freedom ignore dependencies between individual terms and are only true if $\mathbf{X}_i'\mathbf{X}_j' = 0$ for all $i \neq j$. However, the approximation (3.22) was examined by Buja, Hastie & Tibshirani (1989) who found it to provide good results compared to the true degrees of freedom. Figure 3.2 also compares the approximate and the true degrees of freedom for a model with two P–splines, each represented by 22 basis functions. It should

be noted that each plot in figure 3.2 shows the whole range of possible degrees of freedom but that, in real data sets, the individual degrees of freedom for P–splines seldom exceed the value $df_j = 7$. The approximation is very good in plot (a) where the two underlying covariates are uncorrelated and the number of observations $n = 100$ is distinctly higher than the maximal number of parameters ($p = 43$). The largest difference between approximated and true value amounts to 0.8 at $df_{true} \approx 35$. In the case of correlated underlying variables shown in plots (b) and (c), the approximation is a bit worse especially for large individual degrees of freedom. The approximation always overestimates the true number with the largest difference of 3.8 at $df_{true} \approx 31$. This is similar in plot (d) with a small number of observations $n = 50$, which is only slightly larger than the maximal number of parameters, but with uncorrelated underlying variables. Here, the approximation exceeds the true value only for large individual degrees of freedom. The largest difference amounts to 2.8 at $df_{true} \approx 35$.
Note that the approximation of the overall degrees of freedom does not work if the sum of the individual degrees of freedom is larger than the number of observations. The true degrees of freedom cannot exceed the number of observations n.

For non–Gaussian responses, both true and approximate degrees of freedom depend on the current model. The reason is that the hat matrix and the single smoother matrices depend on the IWLS weights. That means, a certain value for a smoothing parameter λ_j can result in different values for df_j if the modelling of other covariates is changed.
In the following, we will describe functions and constellations of functions where the simple approximation (3.22) performs poorly or is clearly wrong. For all functions not mentioned, the simple approximation (3.22) is used.

Fixed effects

As mentioned earlier in this section, the parametric part of the predictor is considered as a special type of function. The intercept term is included in the parametric part, i.e. every model automatically contains a parametric part. The individual degrees of freedom are simply the number of coefficients, i.e.

$$df_{fix} = \text{tr}(\mathbf{H}_{fix}) = p_{fix}. \tag{3.23}$$

I.i.d. Gaussian random effects

Consider now the simple predictor

$$\eta = \gamma_0 + f_{ran}(x)$$

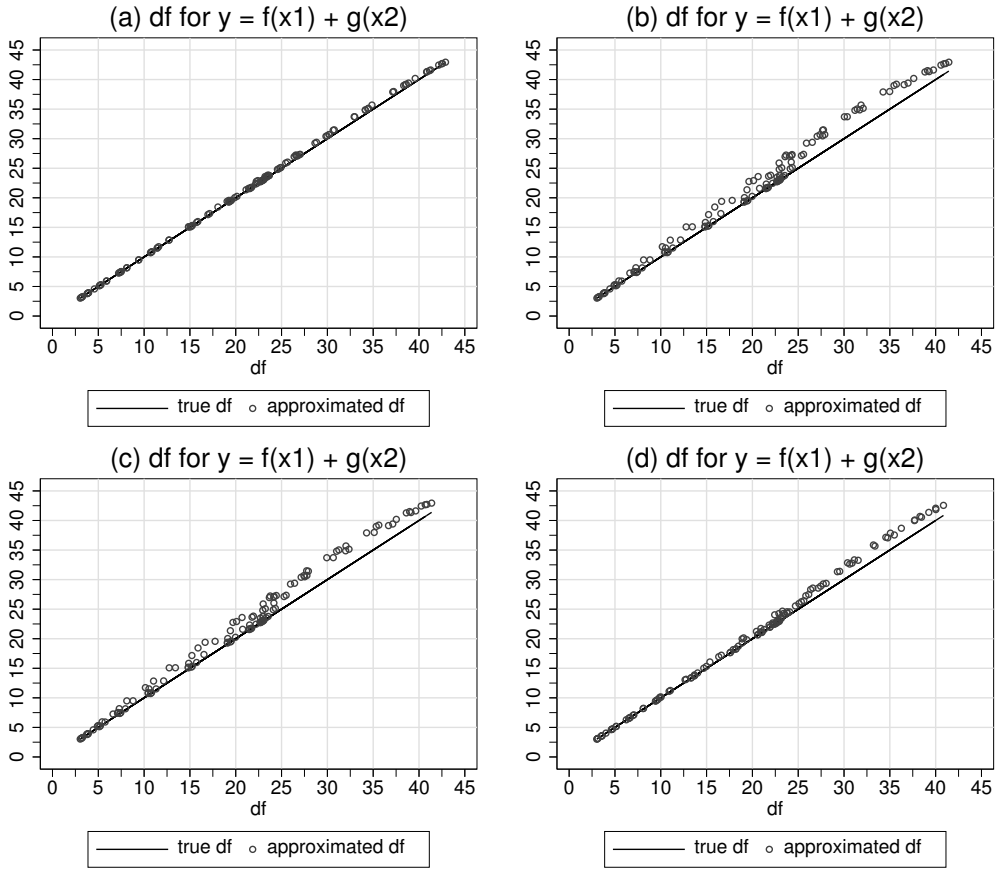

Figure 3.2: Approximated degrees of freedom versus true degrees of freedom for (a) a model with two P–splines each using 22 basis functions with number of observations $n = 100$. The two underlying variables are uncorrelated; (b) a model like in (a) but with positively correlated underlying variables ($\rho = 0.5$); (c) a model like in (a) but with negatively correlated underlying variables ($\rho = -0.5$); (d) a model like in (a) but with number of observations $n = 50$.

only containing an intercept term γ_0 and an i.i.d. random effect $f_{ran}(x)$. As mentioned in section 2.2.6, the null space of the penalty matrix is of dimension zero only containing the zero vector. That means, the function contains no unpenalised constant and is not centered. However, the function contains a penalised intercept term. Therefore, the separate calculation of the degrees of freedom of intercept term and random effect is not possible: For the unpenalised function, i.e. setting $\lambda = 0$, we get $\text{tr}(\mathbf{H}_{ran}) = p$, where p is the number of different groups. In the case of $\lambda \to \infty$, the vector of function evaluations becomes the zero vector, i.e. $\text{tr}(\mathbf{H}_{ran}) = 0$. So, $\text{tr}(\mathbf{H}_{ran})$ lies in the range of $[0; p]$. In contrast, the true degrees of freedom for the above model lie in the range $[1; p]$. The model always contains an intercept term, i.e. the minimal value is one. In the other extreme case, the predictor contains $p + 1$ unpenalised parameters but only p of them can be estimated freely. One

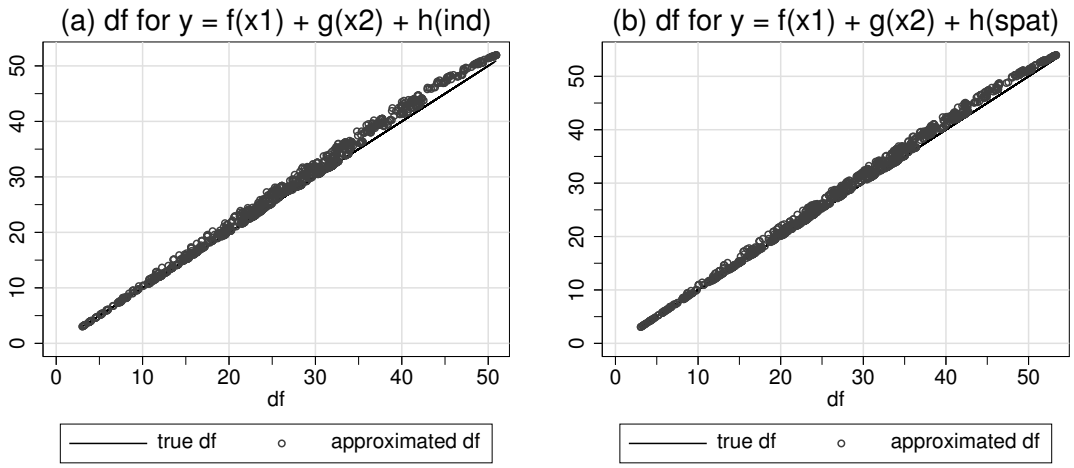

Figure 3.3: Approximated degrees of freedom versus true degrees of freedom for (a) two P–splines with 22 basis functions ($\rho = 0.3$) and an i.i.d. random effect for 10 groups with $n = 100$ (largest difference 3.5 at $df_{true} \approx 40$); (b) two P–splines with 22 basis functions ($\rho = 0.3$) and an MRF for 12 regions with $n = 200$ (largest difference 2.75 at $df_{true} \approx 40$).

parameter can always be calculated from all other parameters (compare section 2.2.2 about categorical covariates).
So, instead of basing the approximation solely upon $\text{tr}(\mathbf{H}_{ran})$, we use formula

$$df_{ran} = \text{tr}\left\{(\mathbf{X}_{ran}, \mathbf{1})\left[(\mathbf{X}_{ran}, \mathbf{1})'\mathbf{W}(\mathbf{X}_{ran}, \mathbf{1}) + \lambda_{ran}\ \text{diag}(\mathbf{I}_p, 0)\right]^{-1}(\mathbf{X}_{ran}, \mathbf{1})'\mathbf{W}\right\} - 1 \quad (3.24)$$

(for $\lambda_{ran} > 0$), where $\mathbf{1}$ is the vector containing value one only. The resulting values lie in the range of $[0; p-1]$. For the simple predictor used in this section, this formula even results in the true degrees of freedom because it takes into account the relationship between intercept term and nonlinear function.
For an efficient computation, formula (3.24) can equivalently be written as

$$df_{ran} = \sum_{k=1}^{p} \frac{-cn_k^3 + n_k^2 - 2cn_k^2\lambda_{ran} + n_k\lambda_{ran}}{(n_k + \lambda_{ran})^2} + n \cdot c - 1, \quad (3.25)$$

where $c = (n - \sum_{k=1}^{p} n_k^2/(n_k + \lambda_{ran}))^{-1}$ and $n_k = \sum_{1 \le i \le n: x_{ik}=1} w_i$ and $n = \sum_{k=1}^{p} n_k$. For an unweighted Gaussian regression model n_k is simply the number of observations in group k. The exact derivation of formula (3.25) is given in section B.1 of the appendix.
Figure 3.3 (a) shows the performance of the approximated degrees of freedom for a model with two P–splines and a random effect. Like in figure 3.2, the degrees of freedom are overestimated, especially for large true degrees of freedom. Note, that the number of observations $n = 100$ is unrealistically small compared to the maximum of $df = 52$.

Spatial effects

Now, we are going to examine spatial effects with decomposition in a smooth and an unstructured component with predictor

$$\eta = \gamma_0 + f_{str}(s) + f_{unstr}(s),$$

where the smooth function $f_{str}(s)$ is modelled by a Markov random field and the unstructured function $f_{unstr}(s)$ by an i.i.d. Gaussian random effect. In this case, the design matrices $\mathbf{X}_{str}$ for the smooth function and $\mathbf{X}_{unstr}$ for the random effect are exactly identical, i.e. $\mathbf{X}_{str} = \mathbf{X}_{unstr} = \mathbf{X}$. The difference between these two functions lies in the penalisation (compare sections 2.2.5 and 2.2.6), i.e. the penalty matrices differ. But for small values of the smoothing parameters, the penalty matrices hardly have any influence on the estimated functions. In this case, the smoother matrices of both functions are nearly identical or even equal for the extreme case of $\lambda_{str} = \lambda_{unstr} = 0$. Hence, the true degrees of freedom for the predictor above lie in the range of $[1; p]$ where the minimal value $df = 1$ can be obtained if both smoothing parameters tend towards infinity and the maximal value p is equal to the number of regions. The maximal value is obtained if the sum $f_{str}(s) + f_{unstr}(s)$ results in unpenalised estimates for all parameters.
In contrast, the individual degrees of freedom of both functions lie in the range of $[0; p-1]$ (by using formula (3.24) for the unstructured function). Hence, adding up the individual degrees of freedom results in a number much too high for small smoothing parameters. Instead, we calculate the degrees of freedom for both functions in one step using the combined design matrix $(\mathbf{X}, \mathbf{X})$ and the combined blockdiagonal penalty matrix $\mathbf{P}_{total} = \text{diag}(\lambda_{unstr}\mathbf{I}_p, \lambda_{str}\mathbf{P}_{str})$ as

$$df_{spat} = df_{str} + df_{unstr} = \text{tr}\left\{(\mathbf{X}, \mathbf{X})\left[(\mathbf{X}, \mathbf{X})'\mathbf{W}(\mathbf{X}, \mathbf{X}) + \mathbf{P}_{total}\right]^{-1}(\mathbf{X}, \mathbf{X})'\mathbf{W}\right\} - 1. \quad (3.26)$$

In order to account for the intercept term contained in the predictor, the value one is subtracted. By using the fact that both matrix $\mathbf{X}'\mathbf{X}$ and matrix $(\mathbf{X}'\mathbf{X} + \lambda_{unstr}\mathbf{I}_p)^{-1}$ are diagonal, formula (3.26) can be transformed into the computationally more efficient formula

$$df_{spat} = \underbrace{\text{tr}(\text{diag}(n_k)\cdot\mathcal{Z}) + \text{tr}(\text{diag}(n_k)\cdot\mathcal{Y}) - 1}_{df_{str}} + \underbrace{\text{tr}(\text{diag}(n_k)\cdot\mathcal{Y}) + \text{tr}(\text{diag}(n_k)\cdot\mathcal{X})}_{df_{unstr}}, \quad (3.27)$$

where the first two terms can be related to the structured and the last two terms to the unstructured spatial function. With $k = 1, \ldots, p$ and $n_k = \sum_{1 \le i \le n: x_{ik}=1} w_k$, matrix $\text{diag}(n_k) = \mathbf{X}'\mathbf{W}\mathbf{X}$. Matrices $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ are elements of the inverse matrix

$$\begin{pmatrix} \mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_{unstr}\mathbf{I}_p & \mathbf{X}'\mathbf{W}\mathbf{X} \\ \mathbf{X}'\mathbf{W}\mathbf{X} & \mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_{str}\mathbf{P}_{str} \end{pmatrix}^{-1} = \begin{pmatrix} \mathcal{X} & \mathcal{Y} \\ \mathcal{Y} & \mathcal{Z} \end{pmatrix}$$

with

$$\begin{aligned} \mathcal{Z} &= \left[\text{diag}\left(\frac{n_k \lambda_{unstr}}{n_k + \lambda_{unstr}}\right) + \lambda_{str}\mathbf{P}_{str}\right]^{-1}, \\ \mathcal{Y} &= -\mathcal{Z} \cdot \text{diag}\left(\frac{n_k}{n_k + \lambda_{unstr}}\right) \end{aligned}$$

and

$$\mathcal{X} = \text{diag}\left(\frac{1}{n_k + \lambda_{unstr}}\right)[\mathbf{I}_p - \text{diag}(n_k) \cdot \mathcal{Y}].$$

An exact derivation of formula (3.27) is given in section B.2 of the appendix. For the simple model only containing the intercept term and the two different spatial functions as mentioned above, formula (3.26) results in the true degrees of freedom.
Figure 3.3 (b) shows the performance of the approximated degrees of freedom for a model with two P–splines and a Markov random field. Like in figure 3.2, the degrees of freedom are overestimated, especially for large true degrees of freedom.

Seasonal Components

Here, we consider the predictor

$$\eta = \gamma_0 + f_{season}(t),$$

containing an intercept term and a seasonal effect with p seasons and period per. Similar to i.i.d. random effects, the null space of a seasonal component (obtained for $\lambda \to \infty$) contains no intercept term (compare section 2.2.4) but only $per - 1$ effect variables. In contrast, for $\lambda \to 0$, the seasonal component consists of p unpenalised dummy variables. This indicates that a seasonal component contains a penalised intercept term. Hence, $\text{tr}(\mathbf{H})$ lies in the range $[3; p]$ whereas the true degrees of freedom for the predictor above can take values from $[4; p]$. So again, the degrees of freedom for the seasonal component can not be calculated independently from the intercept term. Instead, we use formula

$$df_s = \text{tr}\left\{(\mathbf{1}, \mathbf{X}_s)\left[(\mathbf{1}, \mathbf{X}_s)'\mathbf{W}(\mathbf{1}, \mathbf{X}_s) + \lambda_s \text{ diag}(0, \mathbf{P}_{per})\right]^{-1}(\mathbf{1}, \mathbf{X}_s)'\mathbf{W}\right\} - 1 \quad (3.28)$$

$$= \sum_k n_k z_{kk} - \frac{1}{n}\sum_k n_k^2 z_{kk} - \frac{2}{n}\sum_{j>k} n_k n_j z_{jk} \quad (3.29)$$

for the calculation of the individual degrees of freedom df_s with $n_k = \sum_{1 \leq i \leq n: x_{ik}=1} w_i$ for $k = 1, \ldots, p$ and $n = \sum_{k=1}^{p} n_k$. The values z_{jk} are elements of matrix

$$\mathcal{Z} = \left[\begin{pmatrix} n_1 - n_1^2/n & -n_1 n_2/n & \ldots & n_1 n_p/n \\ -n_1 n_2/n & \ddots & & \vdots \\ \vdots & & \ddots & n_{p-1} n_p/n \\ -n_1 n_p/n & \ldots & n_{p-1} n_p/n & n_p - n_p^2/n \end{pmatrix} + \lambda_s \mathbf{P}_{per} \right]^{-1}.$$

The exact derivation of formula (3.29) is given in the appendix section B.3.

Varying coefficients

Among predictors including varying coefficients we have to distinguish between two situations. In the first situation we deal with a predictor of the kind

$$\eta = \gamma_0 + g_1(v_1)x + g_2(v_2)x + \gamma_x \cdot x.$$

In this case, the predictor is not identifiable (compare section 2.3.1) and the varying coefficients have to be centered with respect to the common linear effect of interacting variable x. That means that each varying coefficient loses one degree of freedom to the common linear effect. Hence, the general formula (3.22) can be used to calculate the individual degrees of freedom for both varying coefficients. The exception are random slopes based on i.i.d. Gaussian random effects where formula (3.24) has to applied.
In the second situation we consider the simpler predictor

$$\eta = \gamma_0 + g(v)x.$$

As this predictor contains only one varying coefficient modifying the effect of x, it is not necessary to center the varying coefficient here. That means, the formula for its degrees of freedom is both for random slopes and for other univariate functions given by

$$df_{vc} = \text{tr}(\mathbf{H}).$$

ANOVA type decomposition

Here, we consider a predictor containing only an ANOVA type interaction of two continuous variables x_1 and x_2, i.e.

$$\eta = \gamma_0 + f_1(x_1) + f_2(x_2) + f_{inter}(x_1, x_2) \tag{3.30}$$

as described in section 2.2.8. For this kind of predictor, the complete two–dimensional function $f(x_1, x_2) = f_1(x_1) + f_2(x_2) + f_{inter}(x_1, x_2)$ is estimated first. Only afterwards, the individual terms are calculated from the overall function. Therefore, the degrees of freedom are calculated in the same way: The degrees of freedom df_{all} for the overall function are the trace of the respective smoother matrix, i.e.

$$df_{all} = \text{tr}\left(\mathbf{X'WX}\left(\mathbf{X'WX} + \lambda\mathbf{P} + \frac{\lambda_1}{p}\mathbf{P}_{x_1} + \frac{\lambda_2}{p}\mathbf{P}_{x_2}\right)^{-1}\right) - 1,$$

where λ_1 and λ_2 are the smoothing parameters for functions f_1 and f_2, λ is the smoothing parameter for the interaction component and $\mathbf{X}$ is the tensor product design matrix. For further details regarding this formula compare section 2.2.8. The individual degrees of freedom df_j, $j = 1, 2$ of f_1 and f_2 are calculated using the one–dimensional smoother matrices, i.e.

$$df_j = \text{tr}(\mathbf{X'}_j\mathbf{W}\mathbf{X}_j(\mathbf{X'}_j\mathbf{W}\mathbf{X}_j + \lambda_j\mathbf{P}_j)^{-1}) - 1.$$

The degrees of freedom for the interaction component are then given by

$$df_{inter} = df_{all} - df_1 - df_2.$$

For the simple predictor (3.30), this formula results in the true overall degrees of freedom because it takes the dependencies between the individual terms into account. The true overall degrees of freedom lie between the sum of lower bounds of df_1 and df_2 for large smoothing parameters λ, λ_1 and λ_2 and the number $p^2 - 1$ for small smoothing parameters.

3.4 Algorithms for simultaneous selection of variables and degree of smoothness

In this section we will describe several algorithms for simultaneous selection of variables and the degree of smoothness in structured additive regression models. The simplest algorithm is the stepwise algorithm as implemented in the S–Plus routine *step.gam* and described in Chambers & Hastie (1992) or Hastie & Tibshirani (1990) for additive models. We will give a brief description of this method in the first part of this section. Afterwards, we will introduce a selection algorithm together with some modifications that is based on a mathematical optimisation algorithm, the coordinate descent method. All selection algorithms are designed to answer the questions from the introduction of this chapter. Hence, they are able to

- decide whether a particular covariate or term should be incorporated in the model,

- choose between a linear and non–linear function for a particular continuous variable,
- select the degree of smoothness, i.e. the smoothing parameter for each non–linear function,
- decide if there are complex interactions between certain covariates.

This is done by choosing a good model (according to a selection criterion) from a large set of possible models. The models are composed by choosing from a set of potentially important covariates or terms, where each covariate or term is again provided with a set of modelling alternatives. For the nonlinear modelling alternatives of a term, e.g. the j–th term, a certain number of smoothing parameters

$$\lambda_{j1} > \lambda_{j2} > \ldots > \lambda_{j,m_j}$$

is chosen according to predefined degrees of freedom

$$df_{j1} < df_{j2} < \ldots < df_{j,m_j}. \tag{3.31}$$

That means, the algorithms perform a grid search and do not treat smoothing parameters as continuous. In addition to the possibilities for a nonlinear function defined through smoothing parameters, some other modelling alternatives can be considered, like e.g. excluding the variable or term from the model or using a linear effect. These alternatives depend on the term type and are listed in table 3.2 together with the range of degrees of freedom possible for the respective nonlinear function. The possibility of removing a term from the model (coinciding with $df_j = 0$) is not mentioned in table 3.2 as this alternative exists for each term type. It is possible to decide for each variable or term whether this alternative should be considered or whether the respective term must be included in the predictor. Likewise, the representation by a linear effect which is possible with some terms can be eliminated. For each variable or term, the modelling alternatives are ordered according to their degrees of freedom leading to a list of the form (3.31).
Some specifics for the different term types regarding the choice of modelling alternatives are given in the last column of table 3.2 with some further details given here:

1. In some cases, the smallest degree of freedom possible for the nonlinear function is smaller than the degree of freedom of an extra alternative. Then, the extra alternative has to be correctly positioned between nonlinear alternatives. For example, for a P–spline with first order penalty it is possible to estimate a nonlinear function with $df_j < 1$, whereas $df_j = 1$ corresponds to the linear effect. In this case, the linear effect is positioned between the nonlinear alternatives with $df_j < 1$ and those with $df > 1$.

2. For a two–dimensional P–spline the selection algorithms offer the possibility to use a linear interaction term of the form

$$\gamma_{1,2} \cdot (x_1 - \bar{x}_1)(x_2 - \bar{x}_2). \tag{3.32}$$

Here, the centered covariates $x_1 - \bar{x}_1$ and $x_2 - \bar{x}_2$ are used for the reason shown in figure 3.4: The form of the surface depends on the values of the two covariates. When using the centered variables, the surface is fixed and thus independent of linear transformations of the original covariates. Note that the linear interaction (3.32) is not the limit for a two–dimensional P–spline with a second order penalty and $\lambda \to \infty$.

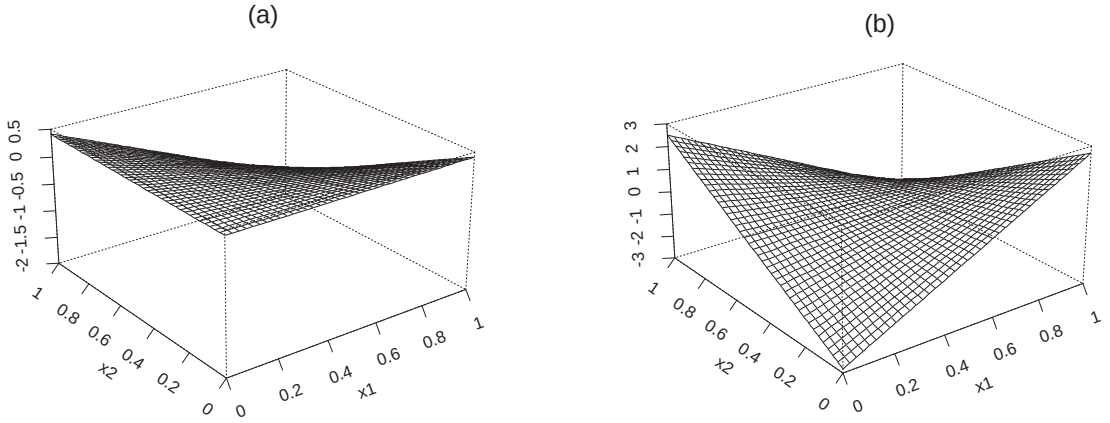

Figure 3.4: Shown are (a) the linear interaction using the non–transformed covariates and (b) the linear interaction using centered covariates.

3. An ANOVA type decomposition is controlled by three smoothing parameters λ, λ_1 and λ_2 in the overall penalty matrix $\mathbf{P}_{comp} = \lambda\mathbf{P} + \lambda_1/p\mathbf{P}_{x_1} + \lambda_2/p\mathbf{P}_{x_2}$. Hence, the degrees of freedom depend on all three smoothing parameters. But the choice of modelling alternatives is carried out separately for the three parameters: The interaction component is mainly controlled by λ and for the determination of values for λ according to predefined degrees of freedom we set $\lambda_1 = \lambda_2 = 0$. Additionally, the interaction component can be a linear interaction of the form (3.32) or can be removed from the model.

The values for λ_1 and λ_2 are determined independently for the respective one–dimensional P–splines, i.e. for the case of a model with main effects only. For the main effects, the alternatives of using a linear effect or removing the term from the model are also possible.

There are, however, some restrictions regarding the extra modelling alternatives (linear fit or exclusion) which have to be considered during the selection process. These are mainly due to the definition of the ANOVA type decomposition where the main

effects are extracted from the overall surface. This, for instance, does not allow the modelling of a main effect by an additional linear effect if the overall surface is also estimated. Altogether, the function type used for the interaction component must never be more complex than the least complex function type used for the main effects. This leads to the following consequences:

- the interaction component may only be included in the model if the model contains both main effects;
- the interaction component cannot be nonlinear if one main effect is linear;
- the main effects cannot be removed from the model if the interaction component is included;
- the main effects cannot be modelled through a linear fit if the interaction component is nonlinear.

4. For varying coefficients $f_j = g(v_j)x$ the modelling alternatives and the respective degrees of freedom depend on whether f_j is identifiable or has to get centered with respect to x. This is considered in table 3.2.

5. The centering of a non–identifiable varying coefficient f_j with respect to variable x has a consequence for the selection process: variable x is automatically included in the model if the varying coefficient f_j is included (even if x was not included in the model before).

When describing the selection algorithms in the next sections, we will use the fact that each possible model is uniquely determined by the combination of modelling alternatives for all covariates and terms, i.e. by vector $(df_1, \ldots, df_q)$. For each function f_j (depending on the function type, compare table 3.2), the selection algorithms can choose between some or all of the alternatives 'removing the term from the model', 'using a linear effect' or 'using a nonlinear function', i.e. the vector of function evaluations $\mathbf{f}_j|df_j$ is estimated by

$$\hat{\mathbf{f}}_j|df_j = \left\{ \begin{array}{l} 0 \\ \hat{\gamma}_j\, x_j \\ \mathbf{X}_j\hat{\boldsymbol{\beta}}_j \end{array} \right\} = \left\{ \begin{array}{ll} 0 & \text{, if } df_j = 0 \\ x_j(x_j'x_j)^{-1}x_j'\tilde{r} & \text{, if } df_j = 1 \\ \mathbf{X}_j(\mathbf{X}_j'\mathbf{X}_j + \lambda_j\mathbf{P}_j)^{-1}\mathbf{X}_j'\tilde{r} & \text{, if } df_j \leftrightarrow \lambda_j, \end{array} \right. \tag{3.33}$$

where $\tilde{r}$ denotes the respective partial residuals. The expression $df_j \leftrightarrow \lambda_j$ indicates the unique relationship between df_j and λ_j. For an ANOVA type decomposition the formula is analogue but determined by the degrees of freedom of all three components.

3.4.1 Stepwise Algorithm

As already mentioned before, an important condition for the stepwise algorithm is the hierarchical ordering of the modelling alternatives for every covariate. Starting from a user–specified basis model, the algorithm changes the modelling of one variable at a time. In doing so, it tries out both adjoining alternatives (from the ordered list (3.31)), i.e. the next complex and the next smooth function. Afterwards, it goes back to the basis model. This process is repeated for each covariate, and only afterwards the basis model is changed. The best among all new models calculated during this one iteration is chosen to become the new basis model. This process is repeated until the new basis model is worse than the old one. In this case, the search is finished and the old basis model is the best model found. The evaluation of the models is based upon a selection criterion.
One modification is to use not only the next but also the second to next or even more neighbouring alternatives.

Stepwise Algorithm

1. Initialisation
   For $j = 1, \ldots, q$:
   Choose a set of modelling alternatives for covariate (or term) x_j as described in the paragraphs above, i.e.
   $$df_{j,1} < \ldots < df_{j,m_j}.$$

2. Starting model
   Choose a starting model, i.e. choose a modelling alternative $df_{j,k_j^{(0)}}$ for each covariate (or term) x_j, where $k_j^{(0)} \in \{1, \ldots m_j\}$. The starting model consists of the set of modelling alternatives given by
   $$\left(df_{1,k_1^{(0)}}, df_{2,k_2^{(0)}}, \ldots, df_{q,k_q^{(0)}}\right)$$
   Estimate this model and calculate the selection criterion $C^{(0)}$.
   Set $r = 1$.

3. Iteration
   For $j = 1, \ldots, q$:

   - For variable x_j try the alternative that is next in complexity, i.e.
     replace $df_{j,k_j^{(r-1)}}$ with $df_{j,k_j^{(r-1)}+1}$ (if existing) which leads to the model
     $$\left(df_{1,k_1^{(r-1)}}, \ldots, df_{j,k_j^{(r-1)}+1}, \ldots, df_{q,k_q^{(r-1)}}\right).$$
     Calculate the selection criterion C_{j+}.

- For variable x_j try the alternative that is next in smoothness, i.e. replace $df_{j,k_j^{(r-1)}}$ with $df_{j,k_j^{(r-1)}-1}$ (if existing) which leads to the model

$$\left(df_{1,k_1^{(r-1)}}, \ldots, df_{j,k_j^{(r-1)}-1}, \ldots, df_{q,k_q^{(r-1)}}\right).$$

  Calculate the selection criterion C_{j-}.

- Go back to the old alternative $df_{j,k_j^{(r-1)}}$ again, i.e. go back to the basis model.

Determine the minimum value amongst $C_{1+}, \ldots, C_{q+}, C_{1-}, \ldots, C_{q-}$ and assign it to $C^{(r)}$. Additionally determine the model, i.e. determine variable x_j and modelling alternative $df_{j,k_j^{(r-1)}+1}$ or $df_{j,k_j^{(r-1)}-1}$, that produced $C^{(r)}$. Use this model which is indicated by

$$\left(df_{1,k_1^{(r)}}, \ldots, df_{j,k_j^{(r)}}, \ldots, df_{q,k_q^{(r)}}\right)$$

as the new basis model. Set $r = r + 1$.

4. Termination
   Step 3. is repeated until the best model of the current iteration is worse than the basis model of this iteration, i.e. until $C^{(r)} > C^{(r-1)}$.
   The best model found is the model belonging to $C^{(r-1)}$.

3.4.2 Algorithms based on the Coordinate Descent Method

The coordinate descent method is a multidimensional optimisation algorithm based on repeated one–dimensional minimisations. Like the stepwise algorithm, the coordinate descent method starts with a user–specified basis model. It also changes the modelling of one covariate or term at a time, but it always checks all possible alternatives. The old modelling of the respective covariate or term is at once replaced by the best alternative. That means, the basis model is changed after each component and is replaced by the currently best model. During one iteration, the algorithm passes through all covariates and terms always using the same sequence. The search is finished if during one entire iteration the model does not change any more.
This process is also shown in figure 3.5 for two continuous variables x_1 and x_2. The upper plot (a) shows the AIC–function in dependence on the individual degrees of freedom df_1 and df_2. The minimum is indicated by the black dot. The lower plot (b) shows a contour plot for the same AIC–function together with the progression of the search. The search starts in the direction of x_1 finding the minimum after two iterations. In contrast to the stepwise algorithm, the order of the variables may influence the progression. This is shown in plot (c). Here, the search starts in the direction of x_2 and needs only 1.5 iterations to

find the minimum.
First, before coming to some modified versions, we will describe the basic coordinate descent algorithm in detail.

Basic algorithm or *exact search*

1. Initialisation
   For $j = 1, \ldots, q$:
   For covariate x_j choose a set of modelling alternatives as described in the paragraphs above, i.e.
   $$df_{j,1} < \ldots < df_{j,m_j}$$

2. Starting model
   Choose a starting model, i.e. choose a modelling alternative $df_{j,k_j^{(0)}}$ for each covariate x_j, where $k_j^{(0)} \in \{1, \ldots m_j\}$. The starting model is given by the set of modelling alternatives
   $$\left(df_{1,k_1^{(0)}}, df_{2,k_2^{(0)}}, \ldots, df_{q,k_q^{(0)}}\right).$$
   Estimate this model and calculate the selection criterion $C^{(0)}$.
   Set $r = 1$.

3. Iteration
   For $j = 1, \ldots, q$:

   For $k \in \{1, \ldots m_j\}, k \neq k_j^{(r-1)}$:
   Estimate the model that replaces $df_{j,k_j^{(r-1)}}$ with $df_{j,k}$, i.e. the model indicated by
   $$\left(df_{1,k_1^{(r)}}, \ldots, df_{j-1,k_{j+1}^{(r)}}, df_{j,k}, df_{j+1,k_{j+1}^{(r-1)}}, \ldots, df_{q,k_q^{(r-1)}}\right)$$
   Change the basis model by replacing $df_{j,k_j^{(r-1)}}$ with $df_{j,k_j^{(r)}}$ minimising the selection criterion over all modelling alternatives for x_j. The new basis model is given by the set
   $$\left(df_{1,k_1^{(r)}}, \ldots, df_{j,k_j^{(r)}}, df_{j+1,k_{j+1}^{(r-1)}}, df_{q,k_q^{(r-1)}}\right).$$

4. Termination
   Repeat step 3. until the modelling alternatives of all covariates do not change.

Modifications

The problematic part of the basic algorithm or *exact search*, as we will call it from now on, is the third step (step 3.). For each covariate or term, the algorithm has to try all modelling

alternatives in order to find the best possibility. In doing so, the algorithm uses backfitting or local scoring procedure, respectively, to estimate every model. This process is very time consuming. In order to overcome this problem, we introduce some modifications of the basic algorithm.

1. *Adaptive search*
   This selection method can not only be seen as a modification of the basic coordinate descent algorithm but also as an adaptive backfitting algorithm instead. We want to introduce the algorithm from the backfitting point of view. A very similar algorithm called BRUTO was already presented by Hastie, Tibshirani & Buja (1994). Like the basic algorithm, the adaptive search starts from a basis model with user specified modelling alternatives for each independent variable or term. This model is estimated via backfitting or local scoring procedure leading to the predictor

   $$\hat{\eta} = \hat{\gamma}_0 + \hat{f}_{1,k_1^{(0)}}(x_1) + \hat{f}_{2,k_2^{(0)}}(x_2) + \ldots + \hat{f}_{q,k_q^{(0)}}(x_q).$$

   In the formula, the effect of each covariate or term is expressed through a function $\hat{f}_{j,k_j^{(0)}}(x_j)$, where removing the variable from the model can be expressed by $\hat{f}_{j,1}(x_j) \equiv 0$ and the linear effect by $\hat{f}_{j,2}(x_j) = \hat{\gamma}_j x_j$ (compare formula (3.33)).
   After estimating the basis model, the algorithm alternately runs through all independent variables and terms, each time updating the respective function estimate $\hat{f}_j$ by basing it on the current partial residuals $y - \hat{\eta} + \hat{f}_{j,k_j^{(r-1)}}$. This is a similar process as is used by the backfitting algorithm. In contrast to the backfitting algorithm, the degree of smoothness of the function is not fixed. Instead, all modelling alternatives $k_j \in \{1, \ldots, m_j\}$ are checked and the alternative $df_{j,k_j^{(r)}}$ currently minimising the selection criterion is chosen for the update. Note, that the intercept term should be adjusted when trying the zero function $df_j = 0$ or the fixed effect $df_j = 1$. With nonlinear functions, the intercept is adjusted automatically.
   For ANOVA type decompositions according to 2.2.8.2 this process has to be changed slightly. Here, the main effects are extracted from the estimated overall surface rather than being estimated as extra components. The surface estimator uses penalty matrix $\mathbf{P}_{comp} = \lambda\mathbf{P} + \lambda_1/p\mathbf{P}_{x_1} + \lambda_2\mathbf{P}_{x_2}$ including all three smoothing parameters. Hence, if one of the smoothing parameters λ, λ_1 or λ_2 is to be chosen, the respective smoothing parameter in $\mathbf{P}_{comp}$ is changed and the overall surface is reestimated. If the selection method decides that a nonlinear interaction component is not necessary, the two main effects are selected and estimated as separate components in the usual way as described above.
   The process described in the paragraphs above is repeated until the modelling of all covariates does not change during three successive iterations. The number three

accounts for changes that could arise due to the improving of function estimates even if there had been no changes in the last iteration. Afterwards, the algorithm switches to backfitting or local scoring procedure in order to obtain the correct penalised maximum likelihood estimates.
With non–Gaussian responses, additional to the process described above, the scale parameter (if unknown) and the IWLS weights are updated after each iteration, i.e. after the algorithm has once passed through all covariates and terms. This process mimics the local scoring procedure with the difference that the local scoring procedure updates scale parameter and IWLS weights only after the convergence of the inner backfitting algorithm.
In contrast to the terminating condition mentioned above, there could be thought of two possibilities as terminating condition: Either the search algorithm could continue until there are no changes in the estimated regression coefficients. But with fixed modelling alternatives, this variation is exactly identical to backfitting or local scoring procedure, just needing more time. The other alternative would be to continue minimising the selection criterion. However, with most criteria this process would be equivalent to maximising the unpenalised log–likelihood and, therefore, would not result in penalised maximum likelihood estimates.
The adaptive search can be interpreted as a modification of the basic coordinate descent algorithm. Thereby, the way of choosing the modelling alternative of one covariate or term is regarded as an approximate one–dimensional minimisation method. The approximation lies in the mere updating of the respective function by formula (3.33) without adjusting all other terms, whereas the exact search always fits the whole model. Moreover, it has to be accepted that the value of the selection criterion can get worse during the process. This is due to the adaptation of the function estimates to the penalised log–likelihood caused by the backfitting updates whereas the selection criteria include the unpenalised log–likelihood.

2. *Adaptive/exact search*
This modification is a combination of the exact and the adaptive search that is intended to combine the advantages of both versions. Here, the adaptive search is performed first. Afterwards, based on the model selected by the adaptive search, an exact search follows. The aim is to select a good model in a short time by the adaptive search. Based on this good model, the exact search is supposed to need only very few iterations to correct errors that are possibly made because of the approximations during the first search. With this process, the combined algorithm is supposed to need less time than the exact search alone but to arrive at the same or a very similar model.

3. *Approximate search*
   This modification is very similar to the adaptive search. The choice of the modelling alternative for each covariate or term is performed exactly as with the adaptive search, i.e. by only updating the estimate of the respective function. The difference to the adaptive search is that, after the choice of the modelling alternative, the approximate search at once estimates this new model using either backfitting algorithm or local scoring procedure. Moreover, the old basis model also is at once replaced by the new model, but only if the new model is better than the old one. This ensures that the selection criterion always improves during the process.
   In simulation studies, the results achieved by the approximate and the adaptive search were nearly identical. Additionally, both methods needed about the same time to select and estimate the models. Hence, we do not use this approximate search in the rest of this thesis.

term	**linear** $\beta' x$	**linear VC** $(\alpha + \beta v)x$	**nonlinear** $[df_{min}; df_{max}]$	**remark**
linear effect	$df = 1$	-	-	-
factor variable ($k+1$–categorical covariate)	$df = k$	-	-	The k dummy– or effect variables are either all included or all excluded from the model.
MRF, i.i.d. random effect	-	-	$[0, p-1]$	-
Seasonal component	-	-	$[per-1, p-1]$	-
RW1, RW1 P–spline	$df = 1$	-	$[0; p-1]$	The linear effect has to be correctly positioned.
RW2, RW2 P–spline	$df = 1$	-	$[1; p-1]$	-
2–dim. RW1 P–spline	$df = 1$	-	$[0; p^2-1]$	The linear effect has to be correctly positioned.
2–dim. RW2 P–spline	$df = 1$	-	$[3; p^2-1]$	Compare remark 2 on page 75.
ANOVA type decomposition	$df = 1$	-	$[2p-2; p^2-1]$	When choosing the values for λ of the interaction component, we set $\lambda_1 = 0$ and $\lambda_2 = 0$; whereas for choosing λ_1 and λ_2 we set $\lambda = \infty$.
identifiable Varying coefficients:				
MRF, i.i.d. random effect	$df = 1$	-	$[1, p]$	The random slope includes an unpenalised linear effect.
i.i.d. random effect with option `nofixed`	-	-	$[0, p]$	The linear effect is penalised.
RW1, RW1 P–spline	$df = 1$	$df = 2$	$[1; p]$	The linear VC has to be correctly positioned.
RW2, RW2 P–spline	$df = 1$	$df = 2$	$[2; p]$	–
2–dim. RW1 P–spline	$df = 1$	$df = 2$	$[1; p^2]$	The linear VC has to be correctly positioned.
2–dim. RW2 P–spline	$df = 1$	$df = 2$	$[4; p^2]$	Compare remark 2 on page 75.
non–identifiable Varying coefficients:				
MRF, i.i.d. random effect	-	-	$[0, p-1]$	–
RW1, RW1 P–spline	-	$df = 1$	$[0; p-1]$	The linear VC has to be correctly positioned.
RW2, RW2 P–spline	-	$df = 1$	$[1; p-1]$	–
2–dim. RW1 P–spline	-	$df = 1$	$[0; p^2-1]$	The linear VC has to be correctly positioned.
2–dim. RW2 P–spline	-	$df = 1$	$[3; p^2-1]$	Compare remark 2 on page 75.
ANOVA type decomposition	-	$df = 1$	$[2p-2; p^2-1]$	See above.

Table 3.2: Overview of modelling alternatives for different term types. The alternative $df = 0$ of removing the term from the model is available for each term type (not included in the table).

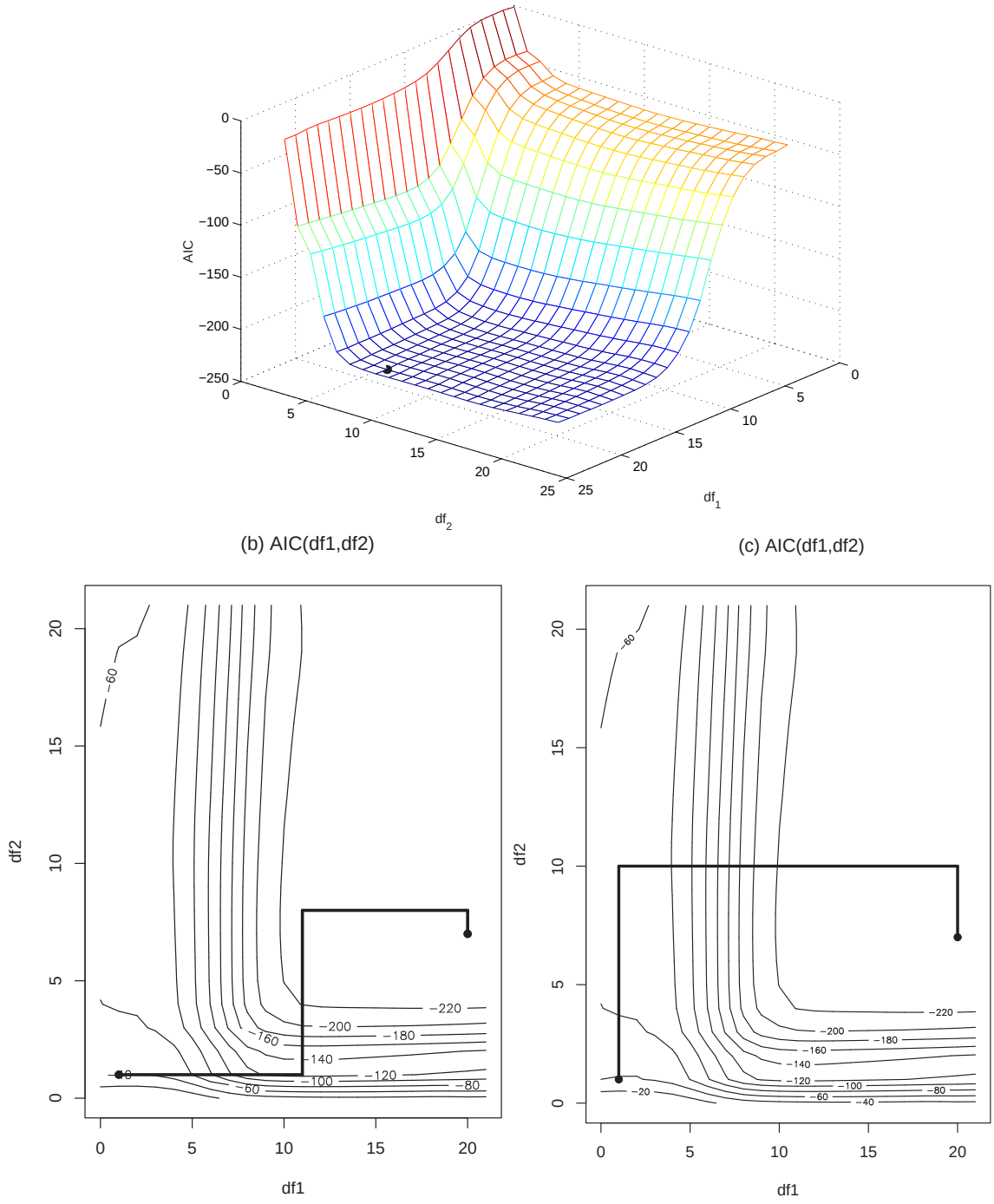

Figure 3.5: (a) shows the AIC as a two–dimensional function of $df_1 = tr(H_1)$ and $df_2 = tr(H_2)$. The black dot indicates the minimum value. (b) shows the coordinate descent method for the AIC function in (a). The algorithm works along the directions of variables x_1 and x_2. After two iterations, it finds the minimum. (c) shows the coordinate descent method with a switched order of variables, i.e. it starts in the direction of x_2. With this order, the algorithm finds the minimum after merely 1.5 iterations. (Nevertheless, it has to complete the second iteration in reality.)

Chapter 4

Structured Additive Multinomial Logit Models

In this chapter, we consider extensions of chapters 2 and 3 to multinomial logit models. The first section deals with inference in these models when dependent variables and smoothing parameters are fixed. The second part describes adjustments regarding the selection algorithms and their components, e.g. the calculation of degrees of freedom.

4.1 Model specification and Inference

In this section, we describe the estimation of regression coefficients in multinomial logit models with fixed covariates and smoothing parameters. More details can be found in Fahrmeir & Tutz (2001) for instance. Multinomial logit models are a special case of multivariate exponential families. We consider here a multinomial distributed response variable Y with $k+1$ different possible outcomes which are labelled by $1, \ldots, k+1$ for simplicity. At first, we consider the case of one trial per observation, i.e. we have

$$Y|\eta \sim M(1, (\pi^{(1)}, \ldots, \pi^{(k)})'),$$

where η denotes the predictor with fixed covariates and smoothing parameters. In an alternative representation, the response variable Y is written as a vector $y = (y^{(1)}, \ldots, y^{(k)})'$ of k indicator variables $y^{(s)}$ given by

$$y^{(s)} = \begin{cases} 1 & , \text{if } Y = s \\ 0 & , \text{otherwise.} \end{cases}$$

The vector $\pi = (\pi^{(1)}, \ldots, \pi^{(k)})'$ contains the probabilities for observing categories $1, \ldots, k$, i.e. we have

$$P(Y = s) = P(y^{(s)} = 1) = \pi^{(s)}.$$

In order to ensure identifiability, the last category $k+1$ serves as reference category with respective probability given by $P(Y = k+1) = 1 - \sum_{s=1}^{k} \pi^{(s)}$.
Using the vector notation of y, it is also possible to consider the more general case with several trials $m \geq 1$ per observation. Similar to binomial data (described in section 2.3.3.1), we use the scaled multinomial distribution in this case. That means, the response variables $y^{(s)}$ denote the relative frequencies of trials with outcome s, i.e. we have

$$(y^{(1)}, \ldots, y^{(k)})'|\eta \sim \frac{1}{m} M(m, (\pi^{(1)}, \ldots, \pi^{(k)})').$$

The vector of conditional expectations $\mu = E(y|\eta)$ is equal to the probability vector, i.e.

$$\mu = (\mu^{(1)}, \ldots, \mu^{(k)})' = (\pi^{(1)}, \ldots, \pi^{(k)})' = \pi.$$

Like in the univariate case, the model specification for the multinomial logit model is based on two different assumptions:

1. **Distributional assumption**

   Given the predictor values η_i, the response variables y_i, $i = 1 \ldots, n$ are conditionally independent. The density of vector y_i can be written in form of a multivariate exponential family, i.e.

   $$f(y_i|\theta_i, \phi, w_i) = \exp\left\{\frac{y_i'\theta_i - b(\theta_i)}{\phi} w_i + c(y_i, \phi, w_i)\right\},$$

   with scale parameter $\phi = 1$ and weights $w_i = m_i$ where m_i denotes the number of trials per observation. The natural parameter θ_i is here

   $$\theta_i = \left[\ln\left(\frac{\pi_i^{(1)}}{1 - \sum_{s=1}^{k} \pi_i^{(s)}}\right), \ldots, \ln\left(\frac{\pi_i^{(1)}}{1 - \sum_{s=1}^{k} \pi_i^{(s)}}\right)\right]'$$

   and function $b(\theta)$ is given by

   $$b(\theta) = -\ln\left(1 - \sum_{s=1}^{k} \pi^{(s)}\right).$$

2. **Structural assumption**

   The conditional expectation $\mu_i = E(y_i|\eta_i)$ is related to the multivariate predictor $\eta_i = (\eta_i^{(1)}, \ldots, \eta_i^{(k)})'$ by

   $$\mu_i = h(\eta_i) \text{ or } \eta_i = g(\mu_i).$$

The multinomial logit model uses the natural link function

$$\eta = g(\pi) = (g_1(\pi), \ldots, g_k(\pi))' = \theta = \left[\ln\left(\frac{\pi^{(1)}}{1 - \sum_{s=1}^{k} \pi^{(s)}} \right), \ldots, \ln\left(\frac{\pi^{(1)}}{1 - \sum_{s=1}^{k} \pi^{(s)}} \right) \right]'$$

and the resulting response function

$$\pi = h(\eta) = (h_1(\eta), \ldots, h_k(\eta))' = \left(\frac{\exp(\eta^{(1)})}{1 + \sum_{s=1}^{k} \exp(\eta^{(s)})}, \ldots, \frac{\exp(\eta^{(k)})}{1 + \sum_{s=1}^{k} \exp(\eta^{(s)})} \right)'.$$

Here again, we have the relation $\mu = \partial b(\theta)/\partial\theta$. With $\phi = 1$ the conditional covariance matrix for observation i is given by

$$Cov(y_i|\eta_i) = \frac{1}{w_i} \frac{\partial b(\theta_i)}{\partial\theta\partial\theta'}$$

with

$$\frac{\partial b(\theta_i)}{\partial\theta\partial\theta'} = \begin{pmatrix} \pi_i^{(1)}(1-\pi_i^{(1)}) & -\pi_i^{(1)}\pi_i^{(2)} & \ldots & -\pi_i^{(1)}\pi_i^{(k)} \\ -\pi_i^{(1)}\pi_i^{(2)} & \pi_i^{(2)}(1-\pi_i^{(2)}) & & \vdots \\ \vdots & & \ddots & \vdots \\ -\pi_i^{(1)}\pi_i^{(k)} & \ldots & -\pi_i^{(k-1)}\pi_i^{(k)} & \pi_i^{(k)}(1-\pi_i^{(k)}) \end{pmatrix} = \frac{\partial h(\eta_i)}{\partial\eta} \quad (4.1)$$

The multivariate predictor η_i for the i–th observation can be written as the product of a design matrix $\mathbf{X}_i$ and a parameter vector $\boldsymbol{\beta}$, i.e.

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

where the design matrix is of dimension $k \times p$ and the parameter vector has length p. The number p is here the overall number of parameters, i.e. $p = \sum_{s=1}^{k} p^{(s)}$ with $p^{(s)}$ indicating the number of parameters for the s–th component $\eta^{(s)}$ of the predictor. The numbers $p^{(s)}$ and the dependent variables can be different for the single components $\eta^{(s)}$. The design matrix is given by

$$\mathbf{X}_i = \begin{pmatrix} x_i^{(1)'} & & & \\ & x_i^{(2)'} & & \\ & & \ddots & \\ & & & x_i^{(k)'} \end{pmatrix},$$

where $x_i^{(s)'}$ contains the covariate values for the component $\eta^{(s)}$. Accordingly, the parameter vector $\boldsymbol{\beta} = (\beta^{(1)'}, \ldots, \beta^{(k)'})'$ contains one subvector for each component. That means, each component has its own regression coefficients what also allows to perform variable selection

separately for each component $\eta^{(s)}$. The overall design matrix $\mathbf{X}$ for all n observations is of order $nk \times p$ and takes the form

$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{pmatrix}.$$

Accordingly, vector $y = (y_1', \ldots, y_n')'$ of length nk is the vector containing all n response variables and $\eta = (\eta_1', \ldots, \eta_n')'$ the overall predictor.
Similar to univariate generalised STAR models, the estimation of the unknown regression parameters can be based on the individual smoother matrices of the individual components so that the computation of the overall design matrix $\mathbf{X}$ is not necessary. The respective estimation algorithm is a modification of the Local Scoring procedure (compare section 2.3.3.2) for multinomial logit models and was presented by Abe (1999). It computes IWLS weights separately for each component and uses the backfitting algorithm to estimate regression parameters. The design matrices used in the formulas below are therefore identical to the design matrices in univariate response models.

Local Scoring procedure

1. Initialisation:
   For $s = 1, \ldots, k$: Set (e.g.) $\hat{\gamma}^{(s,0)} = 0$ and $\hat{\beta}_j^{(s,0)} = 0$ for $j = 1, \ldots, q^{(s)}$.
   Set $r = 1$.

2. For $s = 1, \ldots, k$: Calculation of weight matrix and dependent variable:

$$\begin{aligned}
\mathbf{W}^{(s,r-1)} &= \text{diag}(d_1^{(s,r-1)}, \ldots, d_n^{(s,r-1)}) \\
\eta_i^{(s,r-1)} &= \hat{f}_1^{(s,r-1)}(x_{i1}) + \ldots + \hat{f}_{q^{(s)}}^{(s,r-1)}(x_{iq^{(s)}}) + u_i'\hat{\boldsymbol{\gamma}}^{(s,r-1)} \\
\mu_i^{(s,r-1)} &= h(\eta_i^{(s,r-1)}) \\
\theta_i^{(s,r-1)} &= \eta_i^{(s,r-1)} \\
d_i^{(s,r-1)} &= w_i \frac{\partial h(\eta_i^{(s,r-1)})}{\partial \eta^{(s)}} = w_i \, Var(y_i^{(s)} | \eta_i^{(s,r-1)}) = w_i \, \pi_i^{(s,r-1)}(1 - \pi_i^{(s,r-1)}) \\
\tilde{y}_i^{(s,r-1)} &= \eta_i^{(s,r-1)} + \left(\frac{\partial h(\eta_i^{(s,r-1)})}{\partial \eta^{(s)}} \right)^{-1} (y_i^{(s)} - \mu_i^{(s,r-1)})
\end{aligned}$$

3. Calculation of the weighted least squares estimates using backfitting

(a) Initialisation:
Set $r' = 0$.
For $s = 1, \ldots, k$: Set $\hat{\gamma}^{(s,r')} = \hat{\gamma}^{(s,r-1)}$ and $\hat{\beta}_j^{(s,r')} = \hat{\beta}_j^{(s,r-1)}$ for $j = 1, \ldots, q^{(s)}$.
Set $r' = 1$.

(b) For $s = 1, \ldots, k$:
Calculation of

$$\hat{\boldsymbol{\gamma}}^{(s,r')} = (\mathbf{U}'^{(s)}\mathbf{W}^{(s)}\mathbf{U}^{(s)})^{-1}\mathbf{U}'^{(s)}\mathbf{W}^{(s)}\left(\tilde{y}^{(s)} - \sum_{j=1}^{q^{(s)}} \hat{f}_j^{(s,r'-1)}\right)$$

and for $j = 1, \ldots, q^{(s)}$:

$$\hat{\boldsymbol{\beta}}_j^{(s,r')} = (\mathbf{X}'^{(s)}_j\mathbf{W}^{(s)}\mathbf{X}_j^{(s)} + \lambda_j^{(s)}\mathbf{P}_j^{(s)})^{-1}\mathbf{X}'^{(s)}_j\mathbf{W}^{(s)}res_j^{(s,r')}$$

using the current partial residuals

$$res_j^{(s,r')} = \tilde{y}^{(s)} - \mathbf{U}^{(s)}\hat{\boldsymbol{\gamma}}^{(s,r')} - \sum_{l=1}^{j-1} \hat{f}_l^{(s,r')} - \sum_{l=j+1}^{q^{(s)}} \hat{f}_l^{(s,r'-1)}.$$

(c) For $s = 1, \ldots, k$:
Centering of the nonlinear functions $\hat{f}_j^{(s,r')} = \mathbf{X}_j^{(s)}\hat{\boldsymbol{\beta}}_j^{(s,r')}$ for $j = 1, \ldots, q^{(s)}$:

$$\hat{f}_j^{(c,s,r')} = \hat{f}_j^{(s,r')} - \bar{f}_j^{(s,r')}$$

and adjustment of the intercept term, i.e.

$$\hat{\gamma}_0^{(s,r')} = \hat{\gamma}_0^{(s,r')} + \sum_{j=1}^{q^{(s)}} \bar{f}_j^{(s,r')}$$

or of the common linear effect for varying coefficients.
Set $r' = r' + 1$.

(d) Repeating of (b) and (c) until there are no changes in the estimated parameters.

4. The process terminates if the changes in all parameters are sufficiently small, otherwise set $r = r + 1$ and go back to 2.

4.2 Simultaneous selection of variables and smoothing parameters

In this section we describe the extensions for the selection procedures of chapter 3 to multinomial logit models. As already mentioned in the last section, we consider several,

in general k, response categories where each category has an own predictor with its own parameter values. Therefore, each predictor can also have its own covariates and its own smoothing parameters. Hence, the selection algorithms described in this chapter select dependent variables and smoothing parameters separately for each category. The selection depends on one of the selection criteria described in chapter 3.2 based on the deviance. First, we describe the calculation of degrees of freedom.

4.2.1 Degrees of freedom

Like in univariate models, the true degrees of freedom are calculated using the overall generalised hat matrix from the last iteration of the IWLS algorithm (compare Fahrmeir & Tutz (2001)), i.e.

$$df_{total} = \text{tr}(\mathbf{H}),$$

where $\mathbf{H}$ is the matrix that projects the working response $\tilde{y}$ on the fitted values $\hat{y}$, i.e. $\hat{y} = \mathbf{H}\tilde{y}$. The vector y is the $nk \times 1$ vector containing all observations for all categories. Again, similar to the univariate case, the overall hat matrix $\mathbf{H}$ is difficult to compute. The degrees of freedom for a model are therefore approximated using formula

$$df_{total} = \sum_{s=1}^{k} \sum_{j=1}^{q^{(s)}} df_j^{(s)}. \tag{4.2}$$

The individual degrees of freedom $df_j^{(s)}$ are calculated from the respective smoother matrix $\mathbf{H}_j^{(s)}$ as described in chapter 3 for the univariate case.
In contrast to the univariate case, however, formula (4.2) not only ignores the dependencies between individual terms of one category but also the dependencies between the categories. Matrix (4.1) shows that the covariances of all pairs of categories are unlike zero.

4.2.2 Stepwise Algorithm

In the multivariate case the stepwise algorithm works essentially as for univariate responses. One iteration comprises trying out new models for each category and each term. But every iteration is divided into several parts. The first part contains all terms belonging to the first predictor, the second one all terms belonging to the second predictor and so on. Every part of one iteration is then treated like a complete iteration in the univariate case. That means, after trying out new possible predictors for the first category, the actual basis model is immediately replaced by the best among these models (if this best model is better than the old basis model). Then the algorithm continues with the second category using the

new basis model that was determined after having completed the selection for the first predictor. In this way, all categories are passed alternately always proceeding with the first category after having completed the last (k–th) one. This process continues until the basis model is not replaced during one entire iteration.

4.2.3 Algorithms based on the Coordinate Descent Method

In the multivariate case, the algorithms based on the coordinate descent method work nearly exactly as for univariate models. In the case of the adaptive search, this is possible because the estimation algorithm used for calculating the individual IWLS estimates is also the backfitting algorithm.
Both algorithms, exact and adaptive search, run alternately through all categories starting with the predictor of the first category. For each category, the respective predictor is improved by running once through all terms as described in section 3.4 for univariate responses. Afterwards, both algorithms proceed with the predictor of the next component. When using the adaptive search, the IWLS weights are updated after each category. When the algorithms have completed the last (k–th) category they continue with the first predictor again. This process is repeated until there are no changes in the model during one (exact search) or three successive (adaptive search) iterations. One iteration comprises here all terms of all individual predictors.
In the multivariate case, it is of course also possible to perform the exact search after having completed the adaptive search in order to get the adaptive/exact search as an additional selection procedure.

Chapter 5

Construction of conditional and unconditional credible intervals

In this chapter we describe methods for the construction of credible intervals for nonlinear functions and for regression coefficients of parametric terms. The credible intervals can be conditional or unconditional. Conditional means that the model is considered as fixed and only the regression coefficients show variation, whereas unconditional intervals incorporate the uncertainty induced by model selection. Generally, credible intervals for nonlinear effects are an important visual tool when plotted around the estimated function. They help to detect regions with a higher variability which is often due to few data points.

5.1 Conditional credible intervals

In this section we describe an approach for the construction of credible intervals for regression parameters of linear effects and for nonlinear functions which are conditional on the model selected by one of the selection algorithms of chapter 3. All selection algorithms described there use the backfitting algorithm for the estimation of regression parameters. The backfitting algorithm is a modular algorithm based on the individual smoother matrices. That means, the overall hat matrix is not needed and, therefore, not known. However, the overall hat matrix would be needed for a direct calculation of credible intervals. Out of this reason we calculate conditional credible intervals using a hybrid MCMC approach: first, a model is selected by one of the selection algorithms and, afterwards, MCMC techniques are used to construct credible intervals conditional on this selected model. Thereby, smoothing parameters and scale parameter are set fixed to the values estimated or chosen by the selection algorithm. Hence, the joint posterior distribution of regression parameters

for linear effects and vectors of nonlinear functions is given by

$$\begin{aligned} & p(\boldsymbol{\gamma}, \mathbf{f}_1, \ldots, \mathbf{f}_q | y, \hat{\phi}, \widehat{\mathbf{df}}_0, \widehat{df}_1, \ldots, \widehat{df}_q) \\ \propto \; & L(y|\hat{\phi}, \gamma, \mathbf{f}_1, \ldots, \mathbf{f}_q, \widehat{\mathbf{df}}_0, \widehat{df}_1, \ldots, \widehat{df}_q) \prod_{j=1}^{q} p(\mathbf{f}_j | \hat{\phi}, \widehat{df}_j). \end{aligned} \tag{5.1}$$

Here, the degrees of freedom $\widehat{df}_j$, $j = 1, \ldots, q$, represent the modelling alternative or degree of smoothness chosen for the respective function f_j, whereas $\widehat{\mathbf{df}}_0 = (\widehat{df}_{0,1}, \ldots, \widehat{df}_{0,f})$ is the vector summarising the degrees of freedom selected for the linear effects. Hence, the vector $(\widehat{\mathbf{df}}_0, \widehat{df}_1, \ldots, \widehat{df}_q)$ uniquely specifies the selected model among all different possible models. For many nonlinear functions, the selection algorithms can choose between removing the term from the model, using a linear effect or a nonlinear function, i.e. $\mathbf{f}_j | \widehat{df}_j$ can be expressed by

$$\mathbf{f}_j | \widehat{df}_j = \begin{cases} 0 & , \text{ if } \widehat{df}_j = 0 \\ \gamma_j \, x_j & , \text{ if } \widehat{df}_j = 1 \\ \mathbf{X}_j \boldsymbol{\beta}_j & , \text{ else,} \end{cases}$$

where the usual prior assumptions (compare chapter 2) are made regarding the coefficients $\boldsymbol{\beta}_j$ or γ_j.

MCMC simulation techniques create a Markov chain with the joint posterior (5.1) as stationary distribution. This is achieved by repeatedly drawing random numbers which, at least after a convergence phase, can be considered as random numbers from the joint posterior (5.1). The random numbers can be used to estimate certain quantities of the posterior distribution, like e.g. its mean or even its density function. In our case, we use the random numbers for the construction of credible intervals.

The way in which the random numbers are drawn depends on the type of the response variable y, i.e. one distinguishes between Gaussian responses and non–Gaussian responses from an exponential family, where the Gaussian case is easier to deal with. In both cases, random numbers are not drawn directly from the joint distribution of all functions but are obtained by alternately drawing from the full conditional posterior distributions of one function conditional on all others, i.e. by drawing from $p(\mathbf{f}_j|\cdot) = p(\mathbf{f}_j | y, \hat{\phi}, \gamma, \mathbf{f}_k, k \neq j, \widehat{\mathbf{df}}_0, \widehat{df}_1, \ldots, \widehat{df}_q)$ and $p(\boldsymbol{\gamma}|\cdot)$. For nonlinear functions which are not removed from the model, this is achieved by drawing from the full conditional of the coefficients $\boldsymbol{\beta}_j$ or γ_j and calculating $\mathbf{f}_j$ afterwards.

In the Gaussian case the joint posterior (5.1) of all functions conditional on variance parameter and degrees of freedom is multivariate Gaussian with known parameters. Here, a direct calculation of credible intervals would be possible but would require the overall hat matrix. Hence, we use the Gibbs sampler (compare Green (2001)) which alternately draws

random samples for the individual functions from their full conditionals. That means we again get a modular algorithm which uses the sparse structures of the individual smoother matrices similarly to the backfitting algorithm. For the full conditional of regression parameters for linear effects we get a multivariate Gaussian distribution with expectation and covariance matrix given by

$$E(\boldsymbol{\gamma}|\cdot) = (\mathbf{U}'\mathbf{W}\mathbf{U})^{-1}\mathbf{U}'\mathbf{W}(y - \tilde{\eta}_0) \text{ and } Cov(\boldsymbol{\gamma}|\cdot) = \hat{\sigma}^2(\mathbf{U}'\mathbf{W}\mathbf{U})^{-1}. \tag{5.2}$$

The regression parameters of nonlinear functions also possess multivariate Gaussian full conditionals with

$$E(\boldsymbol{\beta}_j|\cdot) = (\mathbf{X}_j'\mathbf{W}\mathbf{X}_j + \lambda_j\mathbf{P}_j)^{-1}\mathbf{X}_j'\mathbf{W}(y - \tilde{\eta}_j) \text{ and } Cov(\boldsymbol{\beta}_j|\cdot) = \hat{\sigma}^2(\mathbf{X}_j'\mathbf{W}\mathbf{X}_j + \lambda_j\mathbf{P}_j)^{-1}. \tag{5.3}$$

Vectors $\tilde{\eta}_j = \eta - \mathbf{X}_j\boldsymbol{\beta}_j$ and $\tilde{\eta}_0 = \eta - \mathbf{U}\boldsymbol{\gamma}$ are used to construct the respective partial residuals. For details on the drawing of random samples from the full conditionals compare Lang & Brezger (2004) and Rue (2001).

In the non–Gaussian case the form of the joint posterior (5.1) is unknown. Hence, a direct calculation of credible intervals is not possible. Moreover, the form of the individual full conditionals is also unknown so that the Gibbs sampler can no longer be used. Instead, we use a Metropolis–Hastings–algorithm based on IWLS proposals. IWLS proposals were first introduced by Gamerman (1997) and adapted to the context of structured additive regression models by Brezger & Lang (2006).

Suppose, we want to update the function vector $\mathbf{f}_j$. This is achieved by updating the respective regression coefficients $\boldsymbol{\beta}_j$, where $\boldsymbol{\beta}_j^c$ is the current value of the chain. With the Metropolis–Hastings–algorithm, a random sample for $\boldsymbol{\beta}_j$ is created by drawing a proposed vector $\boldsymbol{\beta}_j^p$ from a proposal density $q(\boldsymbol{\beta}_j^c, \boldsymbol{\beta}_j^p)$ which may depend on the current value $\boldsymbol{\beta}_j^c$. The new value $\boldsymbol{\beta}_j^p$ is accepted as new state of the chain with a certain probability $\alpha(\boldsymbol{\beta}_j^c, \boldsymbol{\beta}_j^p)$. If it is not accepted the current state of the chain is used once more as the new value.

The idea of IWLS proposals is to use a multivariate Gaussian distribution as proposal density whose mean and covariance matrix are calculated using one step of the IWLS algorithm. That means, mean and covariance matrix of the Gaussian proposal are analogue to formulas (5.2) and (5.3) where $\hat{\sigma}^2$ is replaced by the general scale parameter $\hat{\phi}$, y by the working response $\tilde{y}$ and matrix $\mathbf{W}$ contains the current IWLS weights based on $\boldsymbol{\beta}_j^c$. The proposed value $\boldsymbol{\beta}_j^p$ is accepted as new value with probability

$$\alpha(\boldsymbol{\beta}_j^c, \boldsymbol{\beta}_j^p) = \min\left(1, \frac{p(\boldsymbol{\beta}_j^p|\cdot)q(\boldsymbol{\beta}_j^p, \boldsymbol{\beta}_j^c)}{p(\boldsymbol{\beta}_j^c|\cdot)q(\boldsymbol{\beta}_j^c, \boldsymbol{\beta}_j^p)}\right),$$

where $p(\boldsymbol{\beta}_j|\cdot)$ is the full conditional for $\boldsymbol{\beta}_j$.

Usually, MCMC techniques need a certain number of iterations in order to converge to the

stationary distribution. The samples from this so called burn–in phase are not used for inference. In our case, the mode of the joint posterior (5.1) has already been calculated by the selection algorithm so that the mode can be used as starting value for the Markov chain. Hence, the Markov chain already starts in its stationary distribution so that a burn–in phase is not necessary. Nevertheless, an analysis of the MCMC output, e.g. of the sampling paths should be performed in order to ensure that no problems have occurred.
The marginal credible intervals for regression cofficients and nonlinear functions regarding significance level α are calculated by using the empirical quantiles $q(\alpha/2)$ and $q(1-\alpha/2)$ of the respective random samples. For a nonlinear function f_j the credible bands are calculated pointwise, i.e. the credible interval for each observation point x_{ij} is computed seperately by using the quantiles of function evaluations at this point.

5.2 Unconditional credible intervals

Model selection can be considered as a kind of estimation procedure (compare Burnham & Anderson (1998)) what is distinct in the following comparison: Estimation of regression parameters means choosing a certain value for each parameter based on some criterion, like e.g. the log–likelihood. This is similar for model selection: Based on one of the selection criteria we choose a certain modelling alternative for each term and certain values for the corresponding regression coefficients. In both cases, the result depends on the current data set. With another sample, the result very likely will be different: In the case of the mere parameter estimation we will get other values for the estimated parameters and with model selection we will get a different best model (and also other estimates for the regression parameters). Hence, when constructing credible bands or intervals we should not only consider the uncertainty in the estimation of regression parameters but also the uncertainty due to model selection. Otherwise, the credible intervals can get too narrow leading to undercoverage.
In the context of this thesis we are mainly interested in constructing credible intervals for regression parameters and nonlinear functions which consider model selection uncertainty. Besides, we are interested to examine the stability of the modelling for individual covariates and terms: Is there a clearly best modelling alternative or should other possibilities also be considered and which are these possibilities?
There are already various approaches for considering model selection uncertainty in (generalised) linear models. Most approaches go beyond the scope of this section and lead to averaged estimates that are obtained by averaging the estimates from several good models. Many approaches for *Model Averaging* are Bayesian like the approach of Geweke (1996) for linear models which is shortly described in section 3.1.1. Here, indicator variables are used

to indicate whether a certain covariate is included in the model or not. Hence, MCMC samples of the regression coefficients can be considered as being obtained from different models and their quantities, like e.g. the mean, as model averaged estimates. An approach for splines based on indicator variables was presented by Yau, Kohn & Wood (2003).
Another approach known as *Bayesian Model Averaging* is, amongst others, described in Raftery, Madigan & Hoeting (1997), Hoeting, Madigan, Raftery & Volinsky (1999) or Clyde & George (2004). Here, the posterior distribution of the parameters θ given the data y is a weighted sum of the posterior distributions of different models M_j, i.e.

$$p(\theta|y) = \sum_j p(\theta|y, M_j)p(M_j|y),$$

where the weights $p(M_j|y)$ are the posterior probabilities for the different models M_j. If the model space is large, the evaluation of this distribution requires the computation of large integrals and sums. Therefore, Occam's window (compare Madigan & Raftery (1994)) restricts the model space to models whose posterior probability is higher than some threshold value. Other approaches use MCMC samplers that can jump between the parameter spaces of different models M_j, e.g. the reversible jump MCMC approach introduced by Green (1995) or the MCMC model composition (MC^3) algorithm described in Madigan & York (1995).
Frequentist approaches for model averaging are often based on bootstrap resampling as described in Burnham & Anderson (1998) or in Augustin, Sauerbrei & Schumacher (2005) for the special case of survival models with a linear predictor. Here, model selection is repeated for each bootstrap sample and model averaged estimates can be obtained by averaging the estimates from all selected models. For an overview and a theoretical background on bootstrap methods compare the monographs of Efron & Tibshirani (1993), Davison & Hinkley (1997) or Shao & Dongsheng (1995).
Bootstrap methods are also frequently used for the construction of credible bands for nonlinear functions. An overview of different bootstrap approaches for the construction of credible intervals is given in Carpenter & Bithell (2000). For smoothing splines, Wang & Wahba (1995) compare bootstrap based credible intervals to Bayesian intervals. Further issues special to the construction of confidence bands for penalised splines are described in Kauermann, Claeskens & Opsomer (2006).
As bootstrap methods have already been used both for the construction of credible bands for nonlinear functions on the one hand and for investigating model selection uncertainty on the other hand, we use bootstrap based methods for the purposes of this chapter. This means that the model selection process is bootstrapped, i.e. a model is selected for each bootstrap data set by using one of the selection algorithms of chapter 3. At first we used pairwise resampling for the construction of bootstrap data sets. This is described in Burn-

ham & Anderson (1998) in combination with bootstrapping of the model selection process. Here, a certain number of bootstrap data sets with as many observations n as the original data set is created by sampling randomly with replacement from the observations of the original data set. However, this approach led to considerable difficulties. The main problem was that the selection algorithms performed badly for the bootstrap data sets and often selected models with many degrees of freedom and rough functions. This is due to the many identical observations in the bootstrap data sets. The formulas for selection criteria like AIC include the number of observations and, as it turned out, for a good performance of these criteria, observations have to be grouped as far as possible. But the grouping of identical observations in the bootstrap data sets would mean to hurt the assumption of using n independent observations. Out of these reasons, we rejected this approach.
Hence, we use parametric bootstrap where the covariates are considered as fixed and only the response vectors are changed. With this approach, adequate models are selected for the bootstrap data. However, there arose a further problem: the credible bands for nonlinear functions based on bootstrap samples are heavily biased. This problem is also mentioned by Kauermann, Claeskens & Opsomer (2006) for instance. The reason is that the estimates of nonlinear functions including a penalty term are biased. This bias is underestimated by bootstrap and thus enlarged. The approach described in Wood (2006c) for the context of smoothing parameter selection avoids this problem. Hence, we adapt this approach to the wider context of a simultaneous selection of variables and degree of smoothness.
The approach of Wood (2006c) is based on the idea that, in a fully Bayesian approach, the joint posterior distribution of the regression parameters for linear effects and vectors of nonlinear function evaluations on the one hand and the degrees of freedom on the other hand can be decomposed as

$$\begin{aligned} & p(\gamma, \mathbf{f}_1, \ldots, \mathbf{f}_q, \mathbf{df}_0, df_1, \ldots, df_q | y) \\ = \; & p(\gamma, \mathbf{f}_1, \ldots, \mathbf{f}_q | \mathbf{df}_0, df_1, \ldots, df_q, y) \cdot p(\mathbf{df}_0, df_1, \ldots, df_q | y), \end{aligned} \tag{5.4}$$

where vector $(\mathbf{df}_0, df_1, \ldots, df_q)$ uniquely specifies all different possible models. The estimation of this joint posterior distribution would require complicated MCMC techniques, e.g. based on indicator variables for each term similar to the approach of Geweke (1996) for linear models. The selection algorithms described in chapter 3 yield an estimated model which is indicated by vector $(\widehat{\mathbf{df}}_0, \widehat{df}_1, \ldots, \widehat{df}_q)$. Hence the idea of Wood (2006c) is to replace df_j by $\widehat{df}_j$ in formula (5.4), thus using the distribution of the frequentist estimates for the degrees of freedom. The unknown distribution $p(\widehat{\mathbf{df}}_0, \widehat{df}_1, \ldots, \widehat{df}_q)$ can be estimated via bootstrap methods so that, actually, we deal with the approximation

$$\begin{aligned} & p(\gamma, \mathbf{f}_1, \ldots, \mathbf{f}_q, \widehat{\mathbf{df}}_0, \widehat{df}_1, \ldots, \widehat{df}_q | y) \\ \approx \; & p(\gamma, \mathbf{f}_1, \ldots, \mathbf{f}_q | \widehat{\mathbf{df}}_0, \widehat{df}_1, \ldots, \widehat{df}_q, y) \cdot \hat{p}(\widehat{\mathbf{df}}_0, \widehat{df}_1, \ldots, \widehat{df}_q). \end{aligned} \tag{5.5}$$

Here, using bootstrap methods means that the model selection process is bootstrapped, i.e. we construct bootstrap response variables $y^{(k)}$, $k = 1 \ldots, B$, and repeat the selection procedure for each $y^{(k)}$. For the simulation of bootstrap responses we use parametric bootstrap (compare Efron & Tibshirani (1993)). This means, we make a distributional assumption regarding the response vector y and use this distribution in combination with the estimated conditional expectations $\hat{\mu}$ of the original response for the simulation of new response variables. In the context of generalised regression models which are also based on a distributional assumption, parametric bootstrap seems to be an appropriate approach. The individual responses $y_i^{(k)}$, $i = 1, \ldots, n$, $k = 1, \ldots, B$, are chosen randomly using a certain distribution D with expectation $\hat{\mu}_i$, scale parameter $\hat{\phi}$ and weight w_i, i.e.

$$y_i^{(k)} \sim D(\hat{\mu}_i, \hat{\phi}, w_i).$$

For instance, for a Gaussian response we get $y_i^{(k)} \sim N(\hat{\mu}_i, \hat{\sigma}^2/w_i)$.
Repeating the selection process $B+1$ times leads to different selected models. Some models are selected more often, other models are never selected. Thus, we get the estimated distribution $\hat{p}(\widehat{\mathbf{df}}_0, \widehat{df}_1, \ldots, \widehat{df}_q)$ by using the relative frequencies of the different models. Models which are selected frequently are more likely to be good models. Similarly, for the individual covariates or terms, the estimated marginal distribution $\hat{p}(\widehat{df}_j)$ can be obtained by using the relative frequencies of the modelling alternatives. This gives a hint as to how stable the respective term is regarding the alternative chosen for the original data y: Some variables or terms are quite stable and only a few similar modelling alternatives are selected. Others are not so stable and more different alternatives are selected with similar frequencies. Hence, bootstrapping offers a sensitivity analysis for model selection.
Apart from the frequency distribution for the different models, we are mainly interested in credible intervals for regression parameters and nonlinear functions. To obtain these credible intervals, Wood (2006c) suggests to combine the bootstrapping of the selection process with MCMC techniques that are used conditional on the selected models like in section 5.1. That means, we draw random numbers for the regression parameters and nonlinear functions conditional on each of the $B+1$ selected models. By using this approach we get random samples that are, at least approximately, from the joint posterior distribution of regression parameters and degrees of freedom.
Here, it is possible that the chosen degrees of freedom for f_j are $df_j = 0$ meaning that the function was removed from the model. In this case, we use a point mass at zero for the sampling of function evaluations, i.e.

$$p(\mathbf{f}_j | \widehat{df}_j = 0) = \begin{cases} 1 & , \mathbf{f}_j = 0 \\ 0 & , \text{else} \end{cases}$$

Altogether, the combined algorithm works as follows:

Algorithm for the construction of unconditional credible intervals

1. Select and estimate a model based on the original data y resulting in estimates $\hat{\boldsymbol{\gamma}}^{(0)}, \hat{\boldsymbol{\beta}}_1^{(0)}, \ldots, \hat{\boldsymbol{\beta}}_q^{(0)}$ for the regression parameters, $\hat{\eta}^{(0)}$ and $\hat{\mu}^{(0)}$ for linear predictor and conditional expectations, and $\widehat{\mathbf{df}}_0^{(0)}, \widehat{df}_1^{(0)}, \ldots, \widehat{df}_q^{(0)}$ for the modelling alternatives.

2. Use the *conditional* approach from section 5.1 for the sampling of random numbers for the regression parameters $\boldsymbol{\gamma}$ and nonlinear functions $\mathbf{f}_j$, $j = 1, \ldots, q$, conditional on the selected model resulting in s samples each.

3. For $k = 1, \ldots, B$, do:

   (a) Simulate a bootstrap response vector $y^{(k)}$ based on the estimates $\hat{\mu}^{(0)}$ by using the distribution assumed for the response y and using the estimate $\hat{\phi}^{(0)}$.

   (b) Select end estimate a model based on the bootstrap response $y^{(k)}$ leading to estimates $\widehat{\mathbf{df}}_0^{(k)}, \widehat{df}_1^{(k)}, \ldots, \widehat{df}_q^{(k)}$ for the modelling alternatives.

   (c) Use the *conditional* approach from section 5.1 for the sampling of random numbers for the regression parameters $\boldsymbol{\gamma}$ and nonlinear functions $\mathbf{f}_j$, $j = 1, \ldots, q$, conditional on the selected model $(\widehat{\mathbf{df}}_0^{(k)}, \widehat{df}_1^{(k)}, \ldots, \widehat{df}_q^{(k)})$ but using the original response y. This results in s samples each.

4. Construct credible intervals for parameters $\boldsymbol{\gamma}$ and nonlinear functions $\mathbf{f}_j$, $j = 1, \ldots, q$, by determining the empirical quantiles to level α of all $(B + 1)s$ MCMC samples.

This combined approach has considerable advantages compared to a simple bootstrap algorithm. As already mentioned, bootstrap estimates of nonlinear functions are usually biased. This problem is solved here, because the regression parameters (after the selection process is finished) are always estimated (and sampled) using the original data y instead of the bootstrap responses $y^{(k)}$. The variables $y^{(k)}$ are merely used for selection.
Furthermore, we do not need to repeat the bootstrapping of the selection process very often. Using $B = 99$ resulting in 100 different models is sufficient to get an estimate for the probability distribution of different models. With a simple bootstrap algorithm, we would have to use the bootstrapping of the selection process in order to obtain enough samples for the calculation of credible intervals. Hence, we would have to use $B \approx 1000$, each time repeating the selection process what is very time consuming.

Chapter 6

Variable and smoothing parameter selection with BayesX

All algorithms introduced in this thesis are implemented in the programming language C++ within the statistical software package *BayesX*. Apart from the approaches for variable and smoothing parameter selection, *BayesX* provides estimation of generalised STAR models either by fully Bayesian inference based on MCMC techniques or by empirical Bayesian inference based on restricted maximum likelihood estimation (REML). An overview of these methods and their usage in *BayesX* can be found in Brezger, Kneib & Lang (2005a) or in the BayesX manuals (Brezger, Kneib & Lang (2005b)), especially the reference and methodological manuals. *BayesX* is free of charge and available at

`http://www.stat.uni-muenchen.de/~bayesx`

together with the manuals mentioned above.
In this chapter, we demonstrate the usage of *BayesX* in combination with the selection algorithms presented in chapters 3–5 on the basis of the Belgian car insurance data from the application in section 8.1. For the general structure of *BayesX* and basic commands, like e.g. the handling of data sets and maps, compare the *BayesX* manuals.
BayesX is object–oriented and the syntax for generating a new object is

`>` *objecttype objectname*

where *objecttype* is the type and *objectname* is the user–defined name of the new object. The Belgian car insurance data is stored in the external ASCII–file `c:\data\carinsurance.raw`. It can be read into *BayesX* by creating a *dataset object* named `d` via the command

```
> dataset d
```

and by storing the data in object `d` using the `infile` command of dataset objects, i.e.

```
> d.infile using c:\data\carinsurance.raw
```

Based on the data, it is possible to estimate a spatially correlated effect for the Belgian districts. For this purpose we need geographical information of Belgium, i.e. the boundaries of the districts, in order to compute the neighbourhood structure. *BayesX* stores the geographical map in a *map object* created with command

```
> map m
```

and, afterwards, reads in the information contained in the external boundary file `c:\data\belgium.bnd` by using the `infile` command for *map objects*:

```
> m.infile using c:\data\belgium.bnd
```

BayesX automatically computes the neighbourhood structure.
In order to perform a variable and smoothing parameter selection in *BayesX*, we start with creating a *stepwisereg object* which we simply call `s`:

```
> stepwisereg s
```

The next step is to specify the output directory and a basis filename for the files containing the estimation results. This is done via the `outfile` command of *stepwisereg objects*:

```
> s.outfile = c:\results\car
```

Now, all results files created by *BayesX* after the selection process are stored in the directory '`c:/results`' and their names begin with the characters '`car`'. If the user does not specify an output directory, the results files are written to the subdirectory '`output`' of the installation directory. In this case, the name of the *stepwisereg object*, i.e. '`s`' in our example, is used as base filename.
The selection is performed using the `regress` command for *stepwisereg objects*. Its general structure is

`> s.regress` *depvar* $= term_1 + term_2 + \ldots + term_r$ [`weight` *weightvar*] [`if` *expression*] [, *options*] `using d`

where *depvar* is the dependent variable, i.e. the logarithmic claim size in our example, and $term_1$, etc. specifies the type of function for the respective covariate (compare tables 6.5 and 6.6). An intercept term is automatically included in the model and is not specified by the user. The part `using d` indicates that data stored in *dataset object* `d` is used for the selection. In the Belgian car insurance example we want to perform a variable and smoothing parameter selection using the dependent variable *logs* (logarithmic claim size),

weight variable *nclaims* (number of claims) and independent variables *ageph* (policyholder's age), *bm* (bonus–malus score) and *s* (gender). A simple linear model based on these variables can be selected and estimated by command

```
> s.regress logs = s + ageph + bm weight nclaims,
  criterion=AIC_imp family=gaussian using d
```

But as we want to investigate whether the continuous variables *ageph* and *bm* possess nonlinear effects, we have to specify the semiparametric predictor

$$\eta = \gamma_0 + \gamma_s s + f_{ageph}(ageph) + f_{bm}(bm),$$

where the two nonlinear functions are represented by P–splines. The selection for this semiparametric predictor can be performed using the command

```
> s.regress logs = s + ageph(psplinerw2,dfmin=2,dfmax=16,number=15) +
                   bm(psplinerw2,dfmin=2,dfmax=16,number=15) weight nclaims,
  criterion=AIC_imp family=gaussian using d
```

For the selection, there are several global options available whose meanings are described in the following list. Possible values and default values are given in tables 6.3 and 6.4.

`algorithm` specifies the selection method that is to be used.

`steps` defines the maximum number of iterations that can be used during the selection process. If the value `steps` is reached before the selection process is finished, the process stops and the results of the current model are written to the results files. If that happens, a warning is written to the output window. By setting `steps=0` it is possible to estimate a certain model without performing a selection.

`criterion` specifies the selection criterion that is to be used.

`proportion` If the selection is based on a criterion using a training and a validation data set, i.e. on `MSEP`, `proportion` defines the fraction of the original data used as training data.

`startmodel` defines the model that is used as basis model.

`number` defines the number of different smoothing parameters to be used for the nonlinear terms. This number can be overwritten using the local option `number`.

`trace` specifies how detailed the output in the *output window* will be.

`CI` specifies if confidence intervals are to be calculated. The default value is `CI=none` so that no confidence intervals are obtained. `CI=MCMCselect` yields confidence intervals which are estimated by MCMC techniques conditional on the selected model, i.e. scale parameter and smoothing parameters are fixed on the values chosen by the preceding selection algorithm. Unconditional confidence intervals can be obtained by `CI=MCMCbootstrap` where several models are selected on the basis of bootstrap samples. For each of the selected models samples are drawn by MCMC techniques conditional on the respective model and based on the original data set. `CI=bootstrap` yields unconditional confidence intervals by selecting many models on the basis of bootstrap samples.

`bootstrap-samples` defines the number of bootstrap samples used for `CI=bootstrap` or `CI=MCMCbootstrap`.

`iterations` defines the number of MCMC iterations used for `CI=MCMCselect` or `CI=MCMCbootstrap`. With `CI=MCMCbootstrap`, option `iterations` specifies the total number of iterations, i.e. the sum of iterations used for the individual conditional MCMC estimations. Here, `iterations` is divided equally between the individual conditional estimations so that the number of iterations used for one model is `iterations / (bootstrap + 1)`. Hence, `iterations` should be chosen appropriately.

`step` is a thinning parameter and specifies that only every `step`–th MCMC–sample is used for the calculation of credible intervals with `CI=MCMCselect` or `CI=MCMCbootstrap`. Since the samples are correlated, the thinning out of MCMC samples is used to obtain approximately independent samples.

`burnin` defines the number of MCMC iterations used for the burn–in phase at the beginning of each conditional MCMC estimation. Hence it is meaningful for `CI=MCMCbootstrap` and `CI=MCMCselect`. The burn–in phase usually is needed to achieve convergence of the Markov chain regarding its stationary (i.e. the posterior) distribution. In our case, the initial estimates for each conditional MCMC estimation are the posterior mode estimates. That means, the Markov chain already starts in its stationary distribution. Hence, the burn–in phase usually is not needed here and it is possible to define `burnin=0` what saves a lot of computing time.

`level1` defines the first significance level for confidence intervals.

`level2` defines the second significance level for confidence intervals.

`predict` By specifying `predict` an additional results file is created containing estimates for the predictor and for the conditional expectation of the response variable.

`family` specifies response distribution and link function.

`reference` specifies the reference category for multinomial logit models.

The commands for specifying different term types for univariate covariates are listed in table 6.5. Possibilities for interactions and the respective commands are shown in table 6.6. For all term types, there are various options which are described below. In the following, we will refer to these options as local options (in constrast to the global options affecting the whole selection process). Possible values for the local options are described in table 6.7 whereas table 6.8 gives a short overview of possible combinations of function terms and local options.

`dfmin` Option `dfmin` defines the smallest possible degree of freedom for a nonlinear function (besides the linear effect). Hence, the largest smoothing parameter is calculated according to `dfmin`. Possible values depend on the number of regression parameters and on the prior distribution (compare section 3.3). In order to avoid numerical problems the smoothing parameter may not become larger than 10^9. If a value larger than 10^9 would be obtained, `dfmin` is repeatedly enlarged by (`dfmax` - `dfmin`) / `number` (and `number` is reduced by one) until $\lambda < 10^9$. Additionally, this ascertains that `dfmin` is redefined to a possible value.

`dfmax` Option `dfmax` defines the largest possible degree of freedom for a nonlinear function. Hence, the smallest smoothing parameter is calculated according to `dfmax`. Possible values depend on the number of regression parameters and on the prior distribution (compare section 3.3). In order to avoid numerical problems the smoothing parameter may not become smaller than 10^{-9}. If a value smaller than 10^{-9} would be obtained, `dfmax` is repeatedly reduced by (`dfmax` - `dfmin`) / `number` (and `number` is reduced by one) until $\lambda > 10^{-9}$. Additionally, this ascertains that `dfmax` is redefined to a possible value.

`dfstart` Option `dfstart` defines the complexity of the function used in the base model. This option is only meaningful if `startmodel=userdefined` is specified. In this case, the default value for `dfstart` is either the fixed effect, if possible, or otherwise the degree of freedom nearest to one.

`logscale` This option causes the smoothing parameters to lie on a logarithmic scale instead of being specified according to equidistant degrees of freedom. In this case, only the smallest and largest smoothing parameters are calculated according to `dfmin` and `dfmax`. This option is only meaningful if option `sp` is not specified (see below).

`df_accuracy` This option specifies the maximal absolute difference in terms of degrees of freedom that is allowed when calculating smoothing parameters according to user–specified degrees of freedom.

`sp` Option `sp` causes the smoothing parameters to be chosen directly according to values specified by options `spmin`, `spmax` and `spstart`. All other values are chosen according to a logarithmic scale. (Options `dfmin`, `dfmax` and `dfstart` are ignored.)

`spmin` This option defines the smallest smoothing parameter but is only valid if `sp` is specified.

`spmax` Option `spmax` defines the largest smoothing parameter but is only valid if `sp` is specified.

`spstart` This option is only meaningful if `startmodel=userdefined` and `sp` are specified. It defines the smoothing parameter used for the base model. Note, that `spstart` can not only take positive values but can also take the values `spstart=0` for excluding the function in the base model and `spstart=-1` for using the fixed effect.

`number` `number` specifies the number of different smoothing parameters (besides the linear effect and exclusion from the model). For `number=0` the global option `number` is used.

`forced_into` This option drops the possibility to exclude the function from the model. That means the respective function is always included in the model.

`nofixed` This option drops the possibility to use a linear fit. Hence, only possibilities for a nonlinear effect and for the removal from the model remain.

`center` `center` has to be specified with varying coefficients if the coefficients must get centered with regard to the interacting variable, i.e. if there are several varying coefficients modifying the same interacting variable. Hence, `center` is only meaningful for varying coefficients and random slopes. The interacting variable has to be specified as separate term.

`coding`	Option `coding` is only meaningful for factor variables. It determines whether dummy variables (`coding=dummy`) or effect variables (`coding=effect`) are used to represent the factor.
`reference`	Option `reference` is again only meaningful for factor variables. It defines the value for the reference category.
`degree`	Specifies the degree of B-spline basis functions.
`nrknots`	Specifies the number of inner knots for a P-spline term.
`monotone`	Option `monotone` specifies additional constraints for univariate P–spline terms. Possible are the estimation of an unrestricted function, a monotonically increasing or decreasing function (i.e. positive/negative first derivative) or a convex or concave function (i.e. positive/negative second derivative). Note, that both type and direction of the constraint have to be defined by the user and are not determined by the selection algorithm.
`gridsize`	The option `gridsize` can be used to restrict the number of points (at the x-axis) for which estimates are computed. By default, estimates are computed at every distinct covariate value in the data set (indicated by `gridsize=-1`). This may be relatively time consuming in situations where the number of distinct covariate values is large. If `gridsize=nrpoints` is specified, estimates are computed on an equidistant grid with `nrpoints` knots.
`period`	The period of the seasonal effect can be specified with option `period`. The default is `period=12` which corresponds to monthly data.
`map`	The map object for a spatial function is defined by option `map`.

Some information about the progression of the selection algorithm and some results are shown in the *output window* whereas other results are only available from external ASCII–files. The *output window* shows all specified covariates and terms together with the respective number of different smoothing parameters and the way in which they were specified. Furthermore, even by specifying option `trace=trace_off`, starting model and final model are shown together with the respective values of the selection criterion. The total number of iterations is also given in the output. By using option `trace=trace_on`, the *output window* additionally shows every model that was tried during iterations. Default value `trace=trace_half` reduces the output to the starting models of the individual iterations. With `trace=trace_off`, the information given in the *output window* is

```
STEPWISE OBJECT s: stepwise procedure

GENERAL OPTIONS:

 Performance criterion: AIC_imp

 RESPONSE DISTRIBUTION:
 Family: Gaussian
 Number of observations: 18139

OPTIONS FOR STEPWISE PROCEDURE:

 OPTIONS FOR FIXED EFFECTS TERM: s

 Startvalue of the 1. startmodel is the fixed effect

 OPTIONS FOR NONPARAMETRIC TERM: ageph

 Minimal value for the smoothing parameter: 2.0480375
 This is equivalent to degrees of freedom: approximately 16, exact 16.0369
 Maximal value for the smoothing parameter: 62500
 This is equivalent to degrees of freedom: approximately 2, exact 1.95119
 Number of different smoothing parameters with equidistant degrees of freedom: 15
 Startvalue of the 1. startmodel is the fixed effect

 OPTIONS FOR NONPARAMETRIC TERM: bm

 Minimal value for the smoothing parameter: 1.0240375
 This is equivalent to degrees of freedom: approximately 16, exact 16.0487
 Maximal value for the smoothing parameter: 45000
 This is equivalent to degrees of freedom: approximately 2, exact 2.02502
 Number of different smoothing parameters with equidistant degrees of freedom: 15
 Startvalue of the 1. startmodel is the fixed effect


STEPWISE PROCEDURE STARTED

 Startmodel:

 LOGS = const + s + ageph + bm
 AIC_imp = 14821.315

 ------------------------------------------------------------------------

 Final Model:

 LOGS = const + ageph(psplinerw2,df=5.96466,(lambda=666.043)) +
        bm(psplinerw2,df=4.96696,(lambda=1188.99))
 AIC_imp = 14757.465

 Used number of iterations: 4
```

The estimation results are stored in several external ASCII-files whose names start with the basis filename `car_`. The file `car_FixedEffects1.res` contains the estimated coefficients for the linear effects in tabular form, including the estimated intercept term and coefficients of factor variables. The results for linear effects are additionally shown in the *output window*. For each nonlinear function, e.g. for $f_{ageph}(ageph)$, there exists one file in form of a data frame, here called `car_f_ageph_pspline.res`, containing the function estimates at all distinct covariate values. The first lines of the file are

```
intnr  ageph    pmean    pqu2p5  pqu10  pmed  pqu90  pqu97p5  pcat95  pcat80
1       18   -0.0003835   0       0     0      0      0        0       0
2       19   -0.0226083   0       0     0      0      0        0       0
3       20   -0.0445334   0       0     0      0      0        0       0
4       21   -0.0659971   0       0     0      0      0        0       0
```

Column *pmean* contains the function estimates. Columns *pqu2p5* to *pcat80* are only meaningful if credible intervals are constructed. In this case, columns *pqu2p5* and *pqu97p5* build the credible interval corresponding to `level1=95`, whereas *pqu10* and *pqu90* belong to the credible interval with `level2=80`. Columns *pcat95* and *pcat80* indicate whether the credible interval is strictly negative (-1), contains zero (0) or is strictly positive (1) with (posterior) probabilities of nominal levels 95% and 80%. The first column *intnr* is merely a parameter index. These results files can be read into any general purpose statistics software (e.g. STATA, R, S-plus) to further analyse and/or visualise the results. The names of the respective files are shown in the *output window*. BayesX has also some facilities for the plotting of nonlinear and spatial functions. The respective commands `plotnonp` and `drwamap` are described in the manuals.
Additional to the files containing estimated effects, there are files containing information about the progression of the selection: the file `car_models.raw` displays the models chosen after every iteration (i.e. after having passed once through all variables and terms). Its contents are

```
step   AIC_imp   model
0   14821.315     LOGS = const + s + ageph + bm
1   14757.645     LOGS = const + ageph(psplinerw2,df=5.96466,(lambda=666.043)) +
                         bm(psplinerw2,df=4.96696,(lambda=1188.99))
...
3   14757.468     LOGS = const + ageph(psplinerw2,df=5.96466,(lambda=666.043)) +
                         bm(psplinerw2,df=4.96696,(lambda=1188.99))
4   14757.465
B   14757.464
```

In this example, variable *s* has been removed from the model during the first iteration, whereas the effects of *ageph* and of *bm* are modelled by nonlinear effects. Column *step*

shows the number of the current iteration with *step=0* indicating the starting model. The information *step=B* is peculiar to the adaptive search where the final model is estimated by backfitting after the selection process is finished what usually changes the value of the selection criterion once more. The largest number of *steps* indicates the total number of iterations. Using this file, it is possible to detect changes in the model that were made during an iteration. Furthermore, it is possible to observe the changes of the selection criterion during the selection process using file `car_criterium.raw` which displays

```
step  var   AIC_imp
0      0   14821.315
0      1   14820.56
0      2   14770.105
0      3   14757.645
1      0   14757.645
1      1   14757.645
1      2   14757.485
1      3   14757.485
...
4      0   14757.465
B      0   14757.464
```

This file displays the current value of the selection criterion after the respective covariate or term was updated. Variable *step* again indicates the number of iterations whereas column *var* gives the number of the covariates / terms. In each iteration, *var=0* indicates the starting model.

If option `predict` is specified, BayesX creates a file `car_predictmean.raw` containing estimates for the predictor η_i in column *linpred* and for the conditional expectations of the response μ_i in column *mu*. If `CI=MCMCbootstrap` is specified the file contains the estimates for the original data in columns *linpred* and *mu* and, additionally, contains average estimates for η_i and μ_i calculated from the samples of all selected models (columns *average_linpred* and *average_mu*). Then, the first lines of `car_predictmean.raw` are given by

```
logs     s  ageph bm nclaims linpred average_linpred   mu   average_mu  sat_dev
11.086   1   50    5   1      9.8551    9.85614      9.8551   9.8561    0.74395
8.7470  -1   28    9   1      9.9052    9.90306      9.9052   9.9031    0.65828
8.7470   1   26   11   1      10.016    10.0125      10.016   10.013    0.79044
```

If unconditional confidence bands were constructed by using options `CI=MCMCboostrap` or `CI=bootstrap`, BayesX creates one additional results file for each nonlinear term and for the linear effects. Those files contain the possible degrees of freedom for the term together with the frequency distribution, i.e. the number of bootstrap samples in which the individual degrees of freedom were selected plus the degree of freedom selected for the original data. For the P–spline effect of *ageph* the file is called `car_f_ageph_pspline_df.res` and contains

```
df_value  sp_value  frequency  selected
4.04862    3447.83     3         -
4.99322    1438.86    26         -
6.03377    632.898    48         +
6.98883    325.333    14         -
7.97817    173.922     1         -
9.96079    56.4291     2         -
11.033     32.1334     2         -
11.9617    19.9828     2         -
13.0304    11.5881     1         -
```

BayesX automatically creates a file `car_model_summary.tex` summarising the most important results which can be compiled using LaTeX. Among the displayed results are graphics for the nonparametric and spatial effects. These graphics are also created automatically and stored in postscript format. The effect of *ageph*, for example, is contained in file `car_f_ageph_pspline.ps`.
When credible intervals are constructed by using one of the hybrid MCMC methods (`CI=MCMCselect` or `CI=MCMCbootstrap`), *BayesX* stores the MCMC samples for the regression parameters of linear effects and for the nonlinear function evaluations. These samples can be obtained using the post estimation command

`s.getsample`

and used for an analysis of the sampling paths. For further information regarding the command `getsample` and the analysis of MCMC output compare the BayesX manuals.

6.1 Specific commands for multinomial logit models

The commands for multinomial logit models differ slightly from the commands for univariate response models. Here, we explain the specifics of these commands: For multinomial logit models, there are two different commands in order to perform a variable and smoothing parameter selection. If the data consists of observations with merely one trial per observation, the dependent variable Y is supposed to specify the chosen category, e.g. $Y \in \{1, \ldots, k+1\}$. In this case, a selection can be performed using the `regress` command like for univariate response variables:

`> s.regress` $Y = term_1 + term_2 + \ldots + term_r$ [`if` *expression*] [, *options*] `using d`

Here, an important option is `reference` specifying the category that is to be chosen as reference category. A weight variable is not allowed with `regress`.
The second possibility is given by the command `mregress`. Here, it is possible to deal with

grouped data with several trials per observation. In this case, *BayesX* needs k response variables, e.g. $Y_1, \ldots, Y_k$, each specifying the numbers of cases in which the respective category was chosen. One category, here $k+1$, serves as reference. The command is

```
> s.mregress Y_1 = term_11 + term_12 + ... + term_1r:
             Y_2 = term_21 + term_22 + ... + term_2r:
             ⋮
             Y_k = term_k1 + term_k2 + ... + term_kr
[weight weightvar] [if expression] [, options] using d
```

The weight variable defines the number of trials per observation. The command `mregress` assumes the same fixed effects for each of the categories and, regarding all other effects, it requires the same number of terms for all categories but not necessarily the same terms. The global and local options are the same as for the `regress` command and local options can be individually specified for each term and category.
With both commands, BayesX creates one results file for each nonlinear term (in every category) containing the estimated effects like in the univariate case. For the linear effects, there exists one results file per category containing all respective parameter estimates. The names of results files for the first category are identical to the names used for univariate response models, e.g. `s_f_varname_pspline.res` for the P–sline effect of variable *varname*. For the j–th category with $j = 2, \ldots, k$ the names additionally contain number j and the P–spline effect of variable *varname* is stored in file `s_f_varname_j_pspline.res`.

global option	type	default	values	description
`algorithm`	string	cdescent1	cdescent1	adaptive search
			cdescent2	exact search
			cdescent3	adaptive/exact search
			stepwise	stepwise algorithm
`criterion`	string	AIC_imp	GCV	GCV (based on deviance residuals, i.e. (3.12) for non–Gaussian, (3.10) for Gaussian response)
			GCVrss	only meaningful for non–Gaussian response: GCV (3.11) based on residual sum of squares
			AIC	AIC
			AIC_imp	improved AIC
			BIC	BIC
			MSEP	MSEP
			CV5	5–fold cross validation
			CV10	10–fold cross validation
			AUC	area under the ROC curve (only for binary response)

Table 6.3: Possible global options for stepwisereg objects.

global option	type	default	values	description
steps	numeric (integer)	100	{0; 10000}	maximum number of iterations
proportion	numeric (real)	0.75	(0; 1)	for MSEP (see description in text)
startmodel	string	empty	empty full userdefined	empty model containing the intercept term only most complex possible model base model specified by the user; default: linear model
number	numeric (integer)	20	{1; 50}	number of smoothing parameters
trace	string	trace_half	trace_on trace_half trace_off	output shows every new model during iterations output shows the starting models of all iterations no output except starting and final model
CI	string	none	none MCMCselect MCMCbootstrap bootstrap	no confidance intervals conditional MCMC confidance bands unconditional confidance bands based on bootstrap and MCMC unconditional MCMC confidance intervals based on bootstrap
bootstrap-samples	numeric (integer)	99	{0; 10000}	number of bootstrap samples
iterations	numeric (integer)	20000	{1; 10000000}	total number of MCMC iterations
step	numeric (integer)	20	{1; 1000}	thinning parameter for MCMC samples
burnin	numeric (integer)	0	{0; 500000}	number of MCMC iterations used for each burnin phase
level1	numeric (real)	95	[40; 99]	first significance level
level2	numeric (real)	80	[40; 99]	second significance level
predict	boolean	false	false true	no estimates for predictor / expectations of response estimates for predictor and expectations are obtained
family	string	logit	gaussian binomial binomialprobit poisson gamma multinomial	Gaussian distribution with identity link Binomial distribution with logit link Binomial distribution with probit link Poisson distribution with log link Gamma distribution with log link Multinomial distribution with logit link
reference	numeric (real)	0	[−10000; 10000]	reference category for multinomial logit models

Table 6.4: Possible global options for stepwisereg objects.

Type	Syntax example	Description
offset	`offs(offset)`	Variable `offs` is an offset term.
linear effect	`W1` `W1(linear)`	Linear effect for `W1`.
factor	`F1(factor)`	Effect of categorical variable `F1`
first or second order random walk	`X1(rw1)` `X1(rw2)`	Nonlinear effect of `X1`.
P-spline	`X1(psplinerw1)` `X1(psplinerw2)`	Nonlinear effect of `X1`.
seasonal prior	`time(season)`	Varying seasonal effect of `time`.
Markov random field	`region(spatial,map=m)`	Spatial effect of `region` where `region` indicates the region an observation belongs to. The boundary information and the neighbourhood structure are stored in *map object* `m`.
Two dimensional P-spline	`region(geosplinerw1,map=m)` `region(geosplinerw2,map=m)`	Spatial effect of `region` by estimating a two dimensional P-spline based on the regions' centroids. The centroids are stored in *map object* `m`.
random intercept	`grvar(random)`	I.i.d. Gaussian random effect of group indicator `grvar`.

Table 6.5: Overview over different model terms for stepwisereg objects.

Type of interaction	Syntax example	Description

Varying coefficient term	`X1*X2(rw1)`
	`X1*X2(rw2)`
	`X1*X2(psplinerw1)`
	`X1*X2(psplinerw2)`
		Effect of `X1` varies smoothly over the range of the continuous covariate `X2`.

random slope	`X1*grvar(random)`	The regression coefficient of `X1` varies with respect to the unit- or cluster–index `grvar`.
Geographically weighted regression	`X1*region(spatial,map=m)`	Effect of `X1` varies geographically. Covariate `region` indicates the region an observation belongs to.
Two dimensional surface	`X1*X2(pspline2dimrw1)` `X1*X2(pspline2dimrw2)`	Two dimensional surface for the continuous covariates `X1` and `X2`.

ANOVA type decomposition	`X1*X2(psplineinteract)`
	`+ X1(psplinerw?)`
	`+ X2(psplinerw?)`
		ANOVA type decomposition for continuous covariates `X1` and `X2`. For the univariate P–splines `rw1` and `rw2` are possible.

Table 6.6: Possible interaction terms for stepwisereg objects.

local option	type	default	values	description
`dfmin`	numeric (real)	–	compare section 3.3	minimal degree of freedom
`dfmax`	numeric (real)	–	compare section 3.3	maximal degree of freedom
`dfstart`	numeric (real)	1	$\{0,1\} \cup [dfmin; dfmax]$	degree of freedom used in the basis model
`logscale`	boolean	false	false true	equidistant degrees of freedom smoothing parameters on a logarithmic scale
`df_accuracy`	numeric (real)	0.05	$[0.01, 0.5]$	accuracy for computing *sp* from predefined *df*
`sp`	boolean	false	false true	smoothing parameters are specified in terms of *df* smoothing parameters are directly specified
`spmin`	numeric (real)	10^{-4}	$[10^{-6}; 10^{8}]$	minimal smoothing parameter
`spmax`	numeric (real)	10^{4}	$[10^{-6}; 10^{8}]$	maximal smoothing parameter
`spstart`	numeric (real)	–	$\{-1, 0\} \cup [10^{-6}; 10^{8}]$	smoothing parameter for the base model
`number`	numeric (integer)	0	$\{0; 100\}$	number of different smoothing parameters
`forced_into`	boolean	false	false true	term may be excluded from the model term may not be excluded from the model
`nofixed`	boolean	false	false true	Possibility of linear fit A linear fit is not possible
`center`	boolean	false	false true	no centering of varying coefficient terms centering of varying coefficient terms
`coding`	string	dummy	dummy effect	dummy coding of categorical variables effect coding of categorical variables
`reference`	numeric (real)	1	$(-100; 100)$	reference category
`degree`	numeric (integer)	3	$\{0; \ldots; 5\}$	degree of B–spline basis functions
`nrknots`	numeric (integer)	20	$\{5; \ldots; 500\}$	number of inner knots for a P–spline term
`monotone`	string	unrestricted	unrestricted increasing decreasing convex concave	no constraint on the spline function monotonically increasing function monotonically increasing function convex function, i.e. positive second derivative concave function, i.e. negative second derivative
`gridsize`	numeric (integer)	-1	$\{-1; 10; \ldots; 500\}$	Specifies the number of data points for output
`period`	numeric (integer)	12	$\{2; \ldots; 72\}$	period for a seasonal effect
`map`	*map object*	–	–	specifies the map object used

Table 6.7: Possible local options for stepwisereg objects. Note, that boolean options are specified without supplying a value.

	linear	factor	rw1 rw2	season	psplinerw1 psplinerw2	spatial	random	geosplinerw1 geosplinerw2	pspline2dimrw1 pspline2dimrw2 psplineinteract
dfmin	—	—	real	real	real	real	real	real	real
dfmax	—	—	real	real	real	real	real	real	real
dfstart	integer	integer	real	real	real	real	real	real	real
logscale	—	—	boolean	boolean	boolean	boolean	boolean	boolean	boolean
sp	—	—	boolean	boolean	boolean	boolean	boolean	boolean	boolean
spmin	—	—	real	real	real	real	real	real	real
spmax	—	—	real	real	real	real	real	real	real
spstart	—	—	real	real	real	real	real	real	real
number	—	—	integer	integer	integer	integer	integer	integer	integer
forced_into	boolean	boolean	boolean	boolean	boolean	boolean	boolean	boolean	boolean
nofixed	—	—	boolean	—	boolean	boolean	boolean	boolean	boolean
center	boolean	—	boolean	—	boolean	boolean	boolean	boolean	boolean
coding	—	string	—	—	—	—	—	—	—
reference	—	real	—	—	—	—	—	—	—
degree	—	—	—	—	integer	—	—	integer	integer
nrknots	—	—	—	—	integer	—	—	integer	integer
monotone	string	—	—	—	string	—	—	—	—
gridsize	—	—	—	—	integer	—	—	—	integer
period	—	—	—	integer	—	—	—	—	—
map	—	—	—	—	—	*map object*	—	*map object*	—

Table 6.8: Terms and options for stepwisereg objects. For possible values to each of the local options compare table 6.7. Note, that boolean options are specified without supplying a value.

Chapter 7

Simulation Studies

In this chapter we present the results of several simulation studies that aim at testing the performance of the selection algorithms described in chapters 3–5, especially of the adaptive search. All simulation studies address the following questions:

- How accurate is the performance regarding selection of relevant covariates and terms? That means, do the algorithms select the important covariates that have an influence on the response and omit irrelevant covariates without an influence?
- How well works the selection of smoothing parameters? That means, do the estimated functions possess a good fit towards their true underlying function? In the case of a linear effect, we like to see, whether the selection algorithms recognise the linear form and avoid a nonlinear modelling of the respective function.
- All algorithms are supposed to minimise the selection criterion. So we are interested to see which of the algorithms obtain the smallest values.
- The computing time differs considerably between the selection approaches. Hence the last topic is to compare the times each of the algorithms needed to estimate all replications.

To answer these questions, we show the results of the following approaches:

- Stepwise algorithm
- Adaptive search
- Adaptive/exact search
- Exact search

- Fully Bayesian approach based on MCMC techniques (see e.g. Fahrmeir & Lang (2001a), Lang & Brezger (2004) and Brezger & Lang (2006)). This approach serves as a benchmark and estimates the true model, i.e. only the variance parameters have to be estimated but the covariates are fixed. Linear functions are estimated by fixed effects.
- Selection with the mgcv package (see Wood (2006b)).

7.1 Simulation of an additive model

The first simulation study is an additive model, i.e. only continuous covariates are available. Whether the algorithms can select an important variable and estimate its effect appropriately often depends on the strength of influence the respective variable has on the response (compare Burnham & Anderson (1998)). For this reason, we used two different classes of functions: the functions in the first class have a large range of values (the distance between minimum and maximum amounts to 2.0) and thus a strong influence on the response whereas the functions in the second class have only a small range (the distance between minimum and maximum amounts to 0.6) and a weak influence. Altogether, we used six different types of functions where each functional form imposes other difficulties for the selection. Every functional form was used twice: once with a strong influence and once with a weak influence. All twelve functions are shown in figure 7.1. The predictor for this additive model is given by

$$\eta = \sum_{j=1}^{12} f_j(x_j).$$

The underlying covariates x_1 to x_{12} were chosen uniformly from the interval $[-3; 3]$ but rounded to two decimal places afterwards. Furthermore, we usually used 18 additional covariates without an influence on the response which were chosen in the same way. All covariates were chosen independently of each other.
For the simulation study we created $R = 250$ replications with $n = 700$ observations each which are based on the following distributional assumptions:

- Gaussian model with response $y_i \sim N(\eta_i, \sigma^2)$ with $\sigma^2 = 1$;
- Binomial logit model with $m = 3$ repeated binary observations, i.e. $y_i \sim B(3, \pi_i)$, where $\pi_i = \frac{\exp(\eta_i)}{1+\exp(\eta_i)}$;
- loglinear Poisson model, i.e. $y_i \sim Po(\lambda_i)$, with $\lambda_i = \exp(\eta_i/2)$;

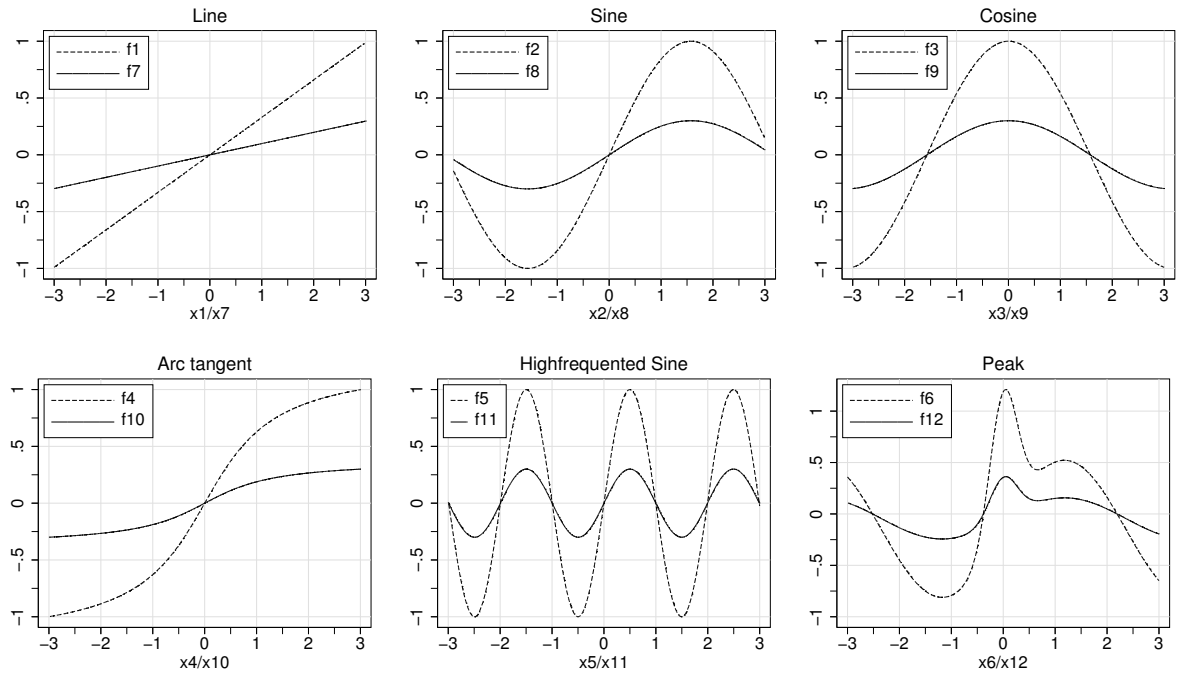

Figure 7.1: True functions for the simulation of the additive model. The two functions contained in the same plot are of the same functional form but with different ranges.

- loglinear Gamma model with $y_i \sim \Gamma(\mu_i, \nu)$ with $\mu_i = \exp(\eta_i)$ and $\nu = 2$.

In order to prevent too extreme predictor values, we used only 8 additional covariates for the Gamma model.
For each covariate we used a P–spline of third degree with a second order penalty and 22 basis functions. The possibilities were in each case the removal from the model, a linear effect or a nonlinear function with possible degrees of freedom $\{2, \ldots, 21\}$. As selection criterion we used AIC_{imp} for the continuous response models (Gaussian and Gamma model). For the discrete response models (Poisson and logit model) we used GCV based on deviance residuals (since AIC_{imp} was especially derived for Gaussian responses as described in Hurvich, Simonoff & Tsai (1998)). Generally, AIC and BIC yielded worse results (not shown).
In order to compare the estimation results of different approaches we examined the different approaches regarding the following aspects:

- We examined the number of wrongly identified variables, i.e. either relevant variables that were removed from the model or irrelevant variables that were added to the model. Additionally, we also analysed the individual numbers of wrongly omitted variables and wrongly added variables.

- We analysed the number of replications in which the linear effects were correctly identified.
- For each individual function f_j, $j = 1, \ldots, 12$, we calculated an average estimate by

$$\bar{f}_j(x_j) = \frac{1}{250} \sum_{i=1}^{250} \hat{f}_{ij}(x_j).$$

  If the variable was removed from the model in replication i the respective function estimate $\hat{f}_{ij}$ was set to zero. For the comparison of linear and nonlinear estimates the linear functions were centered in the same way as the nonlinear functions.
- Additionally we calculated for each function f_j logarithmic empirical mean squared errors (MSE) given by

$$\log(\text{MSE}(f_j)) = \log\left(\frac{1}{m}\sum_{i=1}^{m}\left(\hat{f}_j(x_{ij}) - f_j(x_{ij})\right)^2\right),$$

  where m denotes the number of different values of the underlying covariate x_j. The logarithmic empirical MSE was also calculated for the predictor η using the same formula.
  In this simulation study we often want to compare estimated functions that are of the same functional form but have unequal ranges. In this case we use a logarithmic relative MSE defined as

$$\log(\text{relMSE}(f_j)) = \log\left(\frac{\sum_{i=1}^{m}\left(\hat{f}_j(x_{ij}) - f_j(x_{ij})\right)^2}{\sum_{i=1}^{m}\left(f_j(x_{ij})\right)^2}\right).$$

- For comparison of the obtained values of the selection criterion we used the ratio

$$\text{CR}_i = \frac{\text{C}_i - \min_j(\text{C}_{ij})}{\text{C}_i^{(0)} - \min_j(\text{C}_{ij})}, \tag{7.1}$$

  where C_i denotes the value of the selection criterion that was achieved for the i–th replication by the respective selection method, $\min_j(\text{C}_{ij})$ denotes the best value achieved for the i–th replication among all four selection methods and $\text{C}_i^{(0)}$ denotes the value for the model containing an intercept term only (from now on called *empty model*) and thus yielding the worst value possible. This ratio serves at judging if the models selected by different selection algorithms differ distinctly or if the difference is rather negligible. We use this ratio here because it is not possible to interpret absolute values or even absolute differences of the selection criteria (this is due to the

fact that constant factors are often omitted for the calculation of the criteria). Ratio (7.1) compares the actually achieved improvement (compared to the empty model) to the largest achieved improvement. Note however, that the value $\min(C_i)$ is not automatically the absolute minimum of the criterion function because the selection methods not always find this minimum.

7.1.1 Dependence on the starting model

With each of our selection algorithms the user has the option to specify the basis model from which the selection process starts. In this section we will examine the sensitivity of the selection process regarding the choice of the basis model. For this purpose we compared the results of the approaches

- *adaptive/empty*:
  adaptive search in combination with the empty basis model,
- *adaptive/linear*:
  adaptive search in combination with the linear basis model using a linear effect for each of the 30 available covariates,
- *adaptive/nonlinear*:
  adaptive search in combination with a nonlinear basis model using a function with $df = 10$ for each of the 30 available covariates,
- *stepwise/empty*:
  stepwise algorithm in combination with the empty basis model,
- *stepwise/linear*:
  stepwise algorithm in combination with the linear basis model,
- *stepwise/nonlinear*:
  stepwise algorithm in combination with a nonlinear basis model using a function with $df = 10$ for each of the 30 available covariates.

In order to detect how much the results of the selection algorithms depend on the chosen basis model it suffices to compare the distributions of ratio (7.1) for the respective values of AIC_{imp} and the distributions of the empirical logarithmic MSE for the predictor. These distributions are shown in figures 7.2 and 7.3, respectively. More detailed results are presented in section 7.1.3. The results shown in figures 7.2 and 7.3 lead to the following conclusions:

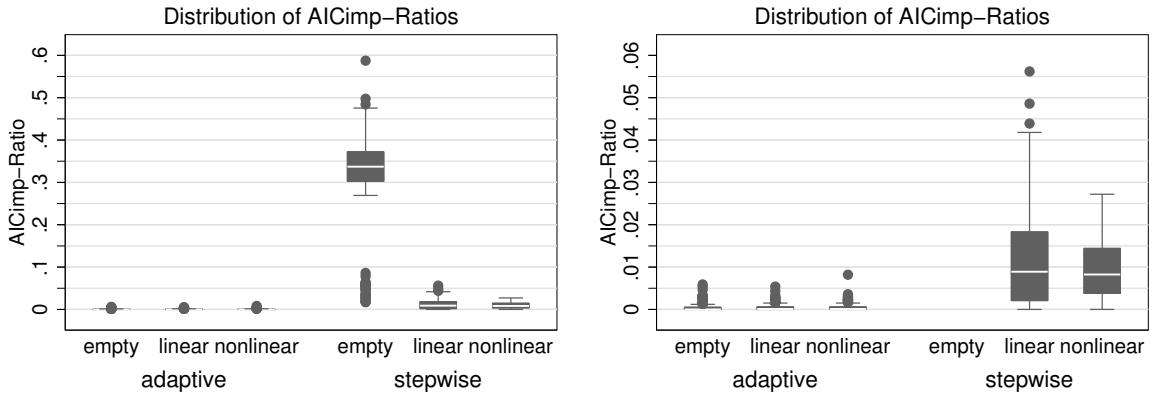

Figure 7.2: Distributions of ratio (7.1) for all different approaches (left plot) and without the results of stepwise/empty *(right plot) for a better comparison of the other results.*

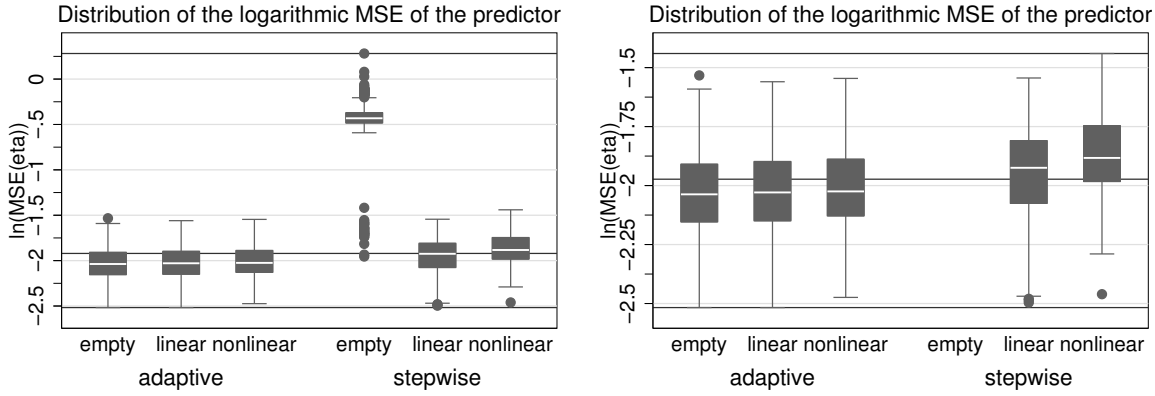

Figure 7.3: Distributions of $\log(MSE(\eta))$ *for all different approaches (left plot) and without the results of* stepwise/empty *(right plot) for a better comparison of the other results. The constant lines indicate the common minimum, median and maximum calculated over all approaches contained in the respective plot.*

- From figure 7.2 it is obvious that the values of AIC_{imp} achieved by the stepwise algorithm strongly depend on the chosen basis model. The *empty* model led to the distinctly worst results. Between the results of the adaptive search there is hardly any difference visible. This indicates that its results are sufficiently independent of the chosen basis model. Additionally, the results of the adaptive search are all distinctly better than those of the stepwise algorithm.
- Figure 7.3 shows that the results regarding the MSE values indicate the same pattern as described above for ratio (7.1).
- Table 7.1 shows the computing time each approach needed to perform the selection for all 250 replications. All adaptive approaches needed about the same computing

time whereas the computing time for the stepwise algorithm strongly depends on the chosen basis model. Additionally, the stepwise algorithm was in each case distinctly slower than the adaptive search.

In summarising these three conclusions, we can see that the results of the stepwise algorithm strongly depend on the chosen basis model whereas the results of the adaptive search are almost independent of the basis model. Based on these results, the linear model proved to be the best starting model for the stepwise algorithm since the empirical MSE took the lowest values and the selection was finished after a moderate time. Merely the values for ratio (7.1) were better for the *nonlinear* basis model. Hence, we use the linear model as basis model throughout the rest of this section.

7.1.2 Dependence on the order of the covariates

In section 3.4.2 we already mentioned that the order of the covariates can influence the progression of the selection algorithms based on the coordinate descent method. Hence, in this section, we want to examine if a different order of covariates changes the results, i.e. if other models are selected. For this purpose we used the adaptive search together with the four different versions:

- *adaptive*:
  The covariates are ordered according to their names. Hence, functions with a large effect are estimated first, then functions with a small effect and covariates without an effect are estimated last, i.e.
  $$\eta = \sum_{j=1}^{6} f_j + \sum_{j=7}^{12} f_j + \sum_{j=13}^{30} f_j.$$
- *order1*:
  Here, we changed the order of the covariates such that the unimportant variables were estimated at first and the functions with a large effect at last, leading to
  $$\eta = \sum_{j=13}^{30} f_j + \sum_{j=7}^{12} f_j + \sum_{j=1}^{6} f_j.$$
- *order2*:
  Here, we only changed the order of the important functions such that the functions with a small effect were estimated first, i.e.
  $$\eta = \sum_{j=7}^{12} f_j + \sum_{j=1}^{6} f_j + \sum_{j=13}^{30} f_j.$$

- *order3*:
  Here, the order of variables was chosen randomly.

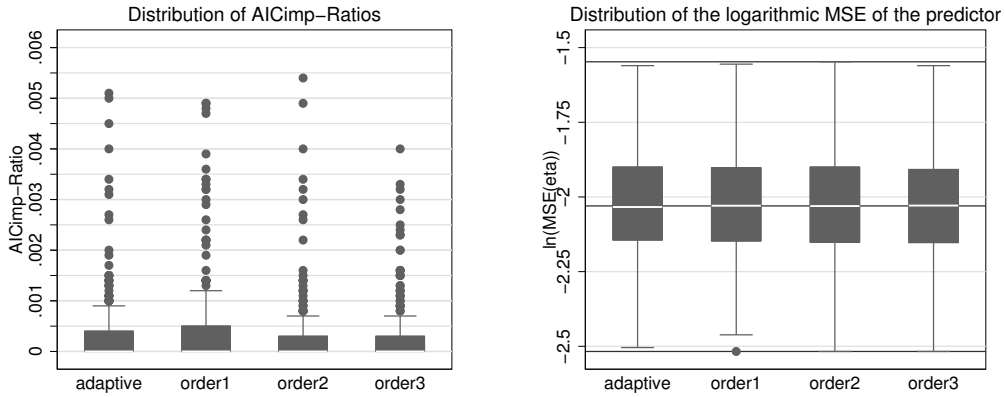

Figure 7.4: Distributions of ratio (7.1) (left plot) and distributions of $\log(MSE(\eta))$ *(right plot) for all different approaches.*

The results of this simulation study show that the finally selected model sometimes changes if the order of the covariates changes. The selected models are, however, very similar as is shown in figure 7.4 in terms of the distributions of ratio (7.1) and of the empirical MSE since these plots show practically no differences. A more thorough investigation of the results (not shown) shows that for many replications the same model is selected and that otherwise the modelling of some covariates merely differs by one or two degrees of freedom. Moreover, as can be concluded from the nearly identical distributions of ratio (7.1), there exists no ordering that is superior to the others.
Hence, as there are only small differences between the four versions, we use the ordering based on the number of terms as in version *adaptive* for the further results of the simulation study.

7.1.3 Detailed results

In this section we show the detailed results of the stepwise algorithm, the adaptive search, the adaptive/exact search and the exact search and compare them to the results achieved by the mgcv package. For the mgcv package we used GCV with $\alpha = 1.4$ (see section 3.2.4) as selection criterion and a smoothing spline with 22 basis functions for each covariate. The penalty for the smoothing splines included a small shrinkage component in order to be able to shrink unimportant terms towards zero and such perform a kind of variable selection. The estimates of the true model obtained by MCMC techniques serve as a benchmark in order to see what could ideally be achieved.

7.1.3.1 Gaussian distribution

algorithm	adaptive	adaptive/exact	exact	stepwise
linear	1:10	7:09	11:28	26:36
empty	1:00			3:14
nonlinear	1:05			143:15
mgcv	184:52			
MCMC	5:11			

Table 7.1: Gaussian distribution: Computing times in hours for all 250 replications each.

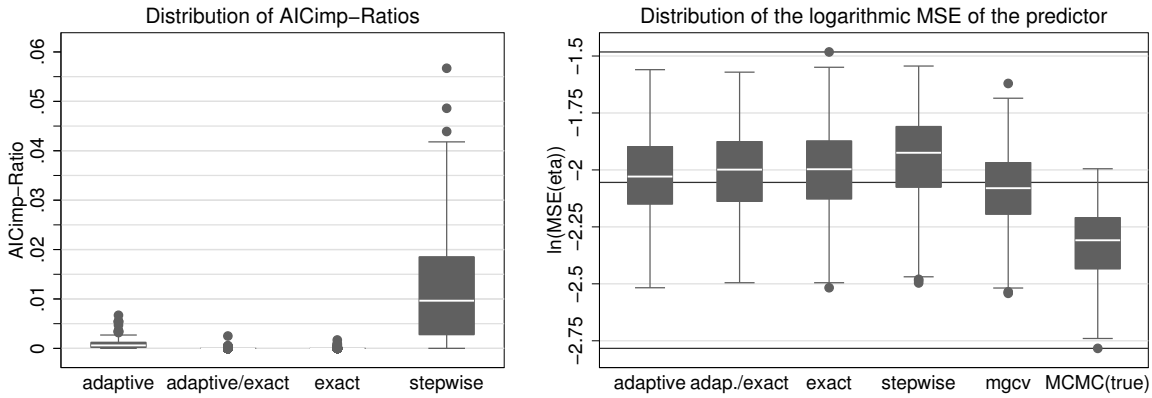

Figure 7.5: Gaussian distribution: The left plot shows the distributions of ratio (7.1) for AIC_{imp} values. The right plot compares the distributions of $\log(MSE(\eta))$. *Here, the constant lines indicate the common minimum, median and maximum calculated over all approaches.*

From the results of the Gaussian model we can draw the following conclusions:

- The stepwise algorithm produced worse results than the selection algorithms derived from the coordinate descent method. This applies to the results regarding the distribution of ratio (7.1), the distribution of the logarithmic MSE of the predictor (both shown in figure 7.5) and the number of wrongly identified variables (see figure 7.10). In contrast, the distributions of the logarithmic relative MSE of the individual functions (shown in figures 7.6 and 7.7) are mostly not distinguishable between the four selection algorithms. Here, the only exception is function f_{11} where the stepwise algorithm produced distinctly worse results.
- The three selection algorithms derived from the coordinate descent method achieved practically the same results regarding MSE values, number of wrongly identified variables and average function estimates (not shown). The values of ratio (7.1)

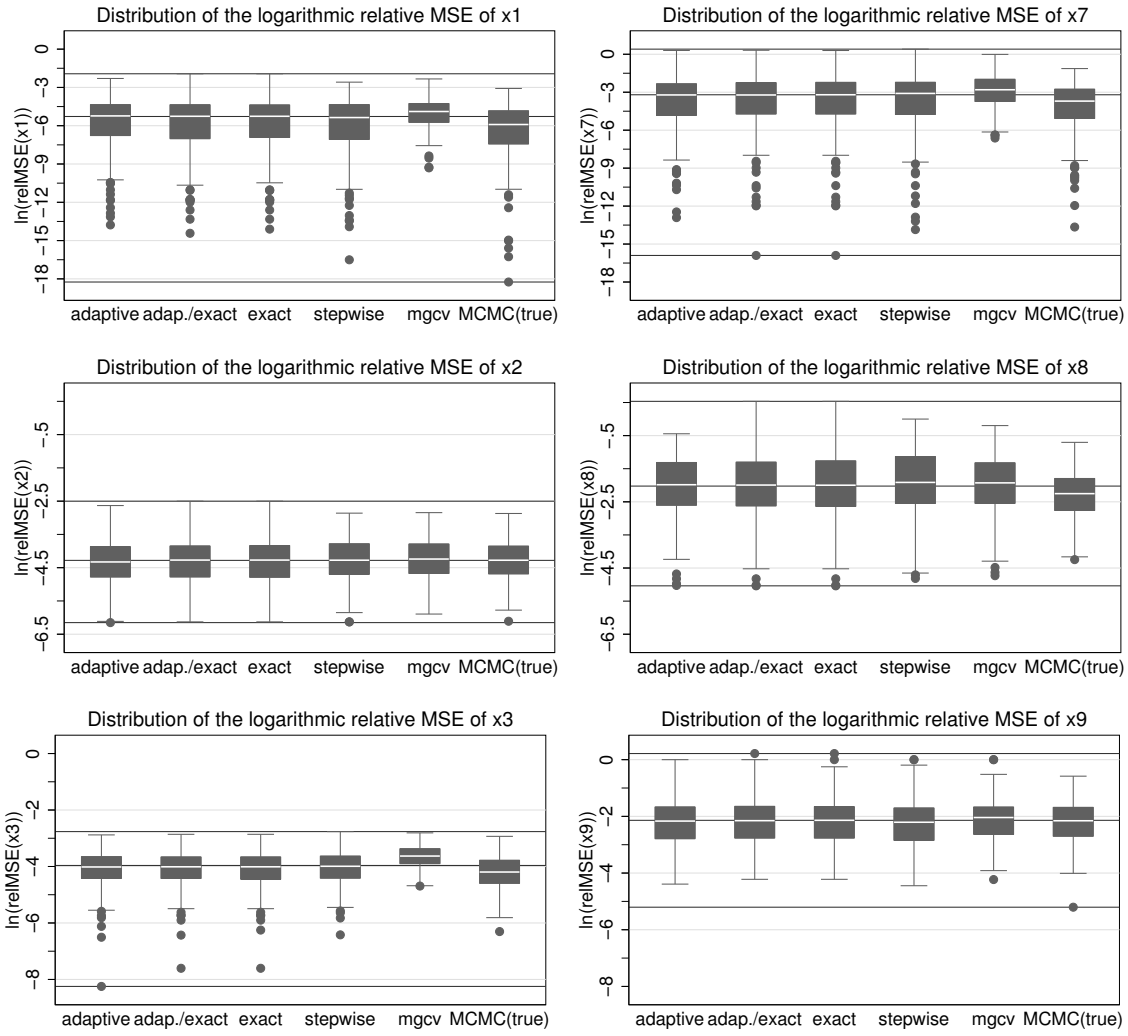

Figure 7.6: Gaussian distribution: Distributions of the logarithmic relative MSE for the individual functions. Each row compares the functions that are of the same functional form where the functions with a large influence are in the left column and the functions with a small influence in the right one. The constant lines indicate in each case the common minimum, median and maximum calculated over all algorithms.

were slightly larger for the adaptive search than for the other two methods but this difference is negligible since the largest value for the adaptive search only amounts to about 0.006. This means, that if $C_i^{(0)} - min_j(C_{ij}) = 100$, the difference between the value of the adaptive search and the minimum would be merely $C_i - min_j(C_{ij}) = 0.6$. The most important difference between these approaches is the time they needed to perform the selection for all 250 replications (compare table 7.1). The adaptive search is by far the most efficient approach and needed even considerably less time than the estimation of the true model by MCMC techniques.

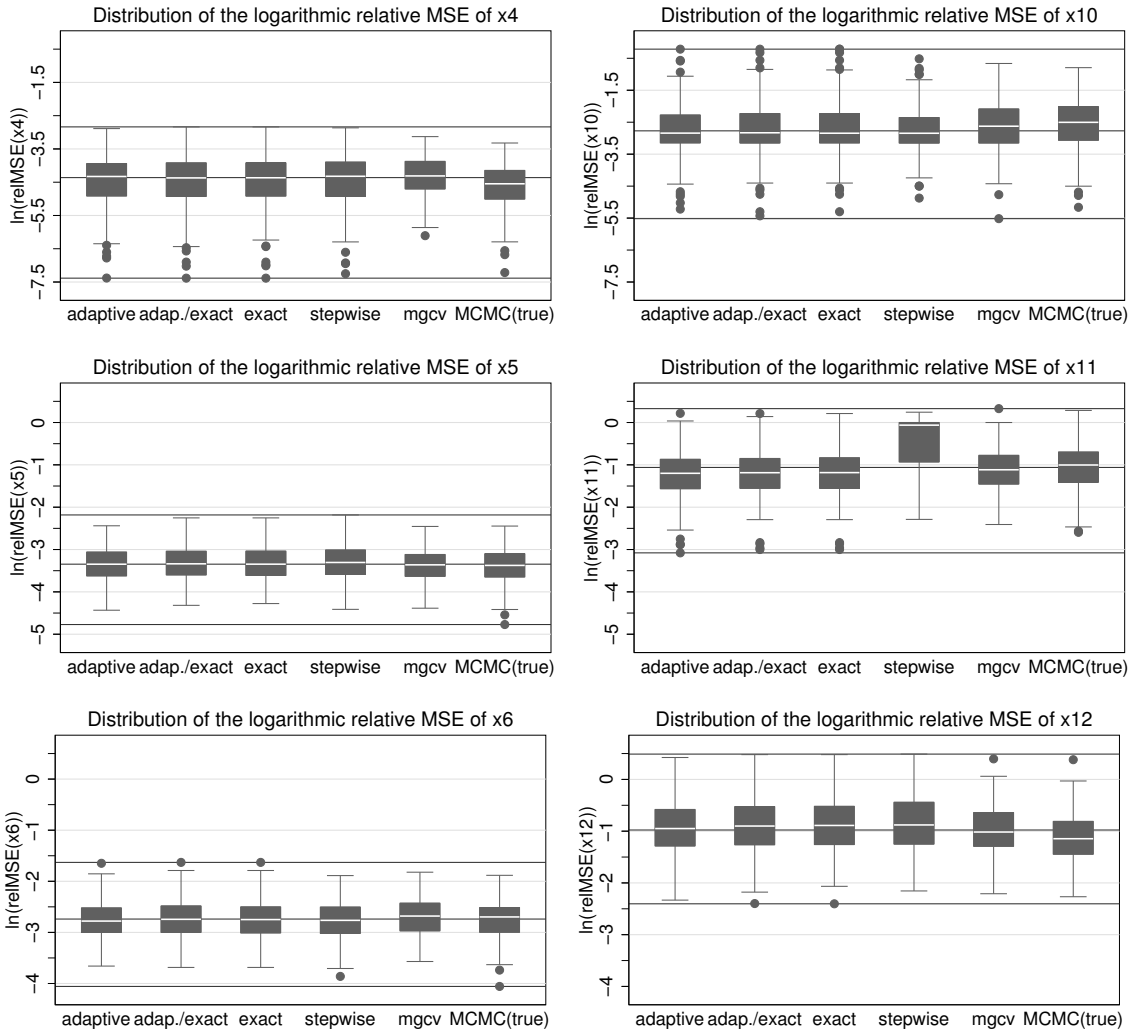

Figure 7.7: Gaussian distribution: Distributions of the logarithmic relative MSE for the individual functions. Each row compares the functions that are of the same functional form where the functions with a large influence are in the left column and the functions with a small influence in the right one. The constant lines indicate in each case the common minimum, median and maximum calculated over all approaches.

Hence, as the results of all coordinate descent methods are practically the same and the computing time of the adaptive search is considerably lower, the adaptive search is the most preferable selection algorithm.

- The distributions of the empirical MSE of the predictor (shown in figure 7.5) indicates that the estimates of the predictor for *MCMC (true)* conditional on the true predictor are superior to the estimates achieved by any of the selection algorithms. The results of mgcv are, however, only slightly better than those of the coordinate

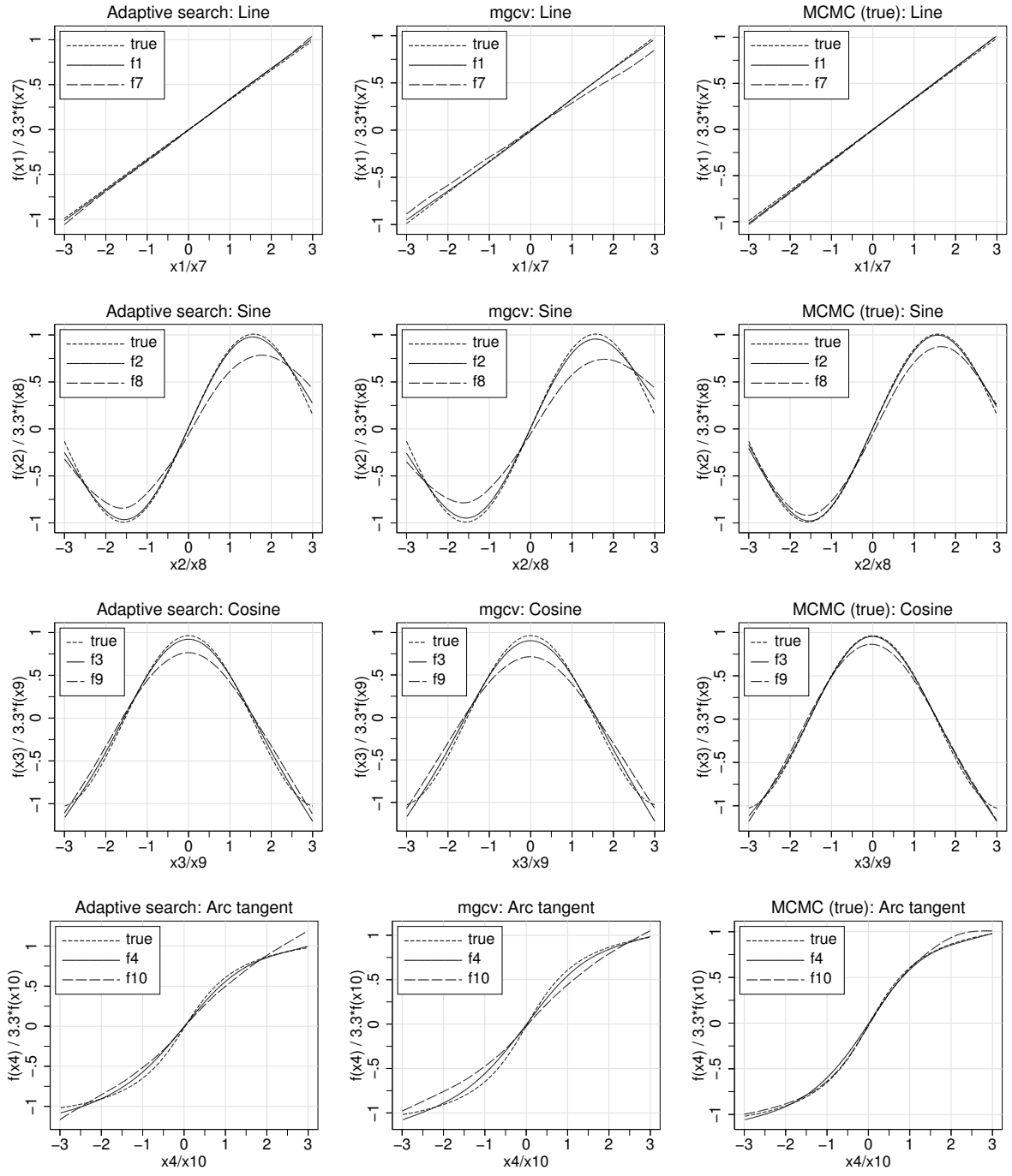

Figure 7.8: Gaussian distribution: Average estimated functions together with the true underlying functions for the adaptive search (left column), the mgcv package (middle) and the true model estimated by MCMC techniques (right column). By multiplying the weak functions with factor 3.3, both functions of the same type are plotted on the same scale.

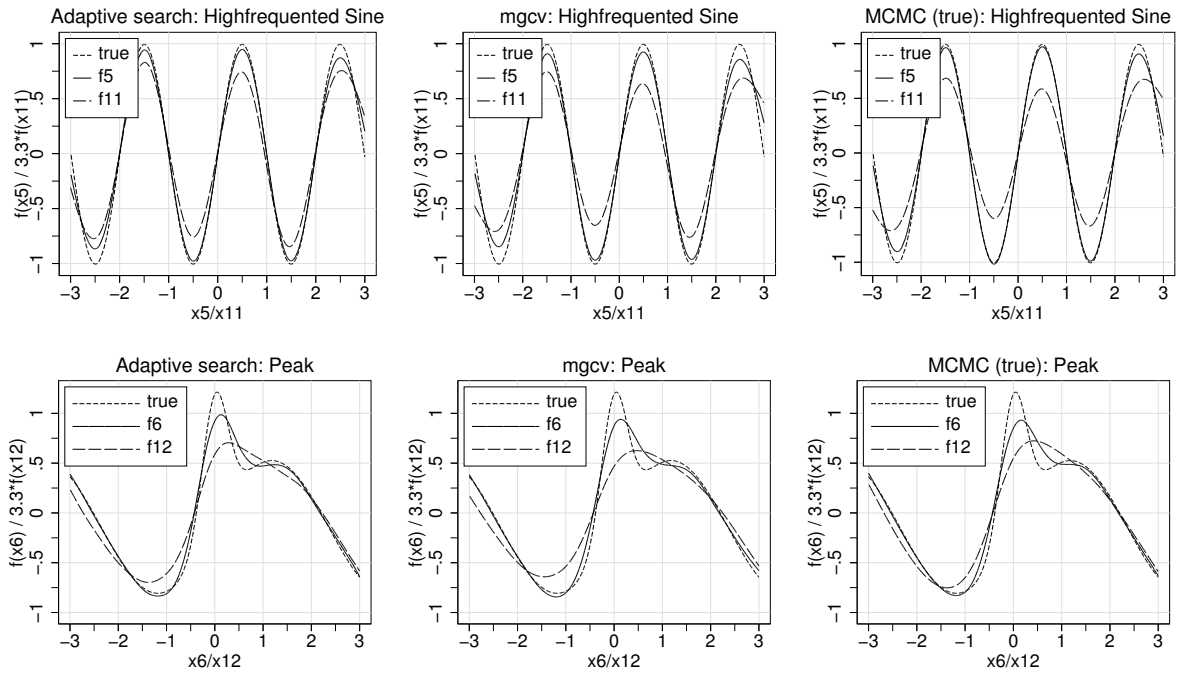

Figure 7.9: Gaussian distribution: Average estimated functions together with the true underlying functions for the adaptive search (left column), the mgcv package (middle) and the true model estimated by MCMC techniques (right column). By multiplying the weak functions with factor 3.3, both functions of the same type are plotted on the same scale.

descent methods.

In contrast, the average function estimates (shown in figures 7.8 and 7.9) and the distributions of the empirical relative MSE (compare figures 7.6 and 7.7) show merely small differences (apart from the stepwise algorithm). As expected, the smallest bias of individual functions was achieved conditional on the true model. The bias of the individual mgcv estimates is often slightly larger than for the adaptive search.

- When analysing the number of wrongly omitted covariates (figure 7.10), the coordinate descent methods show show comparable results to mgcv. In contrast, the number of wrongly identified variables is considerably larger for mgcv. This is due to the fact that mgcv treats smoothing parameters as continues and therefore can estimate functions with very small degrees of freedom that are, nevertheless, unequal to zero. The same could be observed for the number of replications in which the linear effects were correctly identified (not shown for mgcv). Here, mgcv hardly ever used an exactly linear effect.

  Altogether, the results achieved by the coordinate descent methods are as least as good as those achieved by mgcv. The biggest advantage of our approach is the com-

puting time for the 250 replications (compare table 7.1). The adaptive search needed a bit more than an hour whereas mgcv needed more than a week for the estimation of this complex model.

- In the introduction we mentioned that the performance of selection algorithms regarding individual covariates depends on the strength of influence of the respective effect. The average function estimates in figures 7.8 and 7.9 show that the weak functions are always more heavily biased than the strong functions, whereas some of the strong functions are nearly unbiased. A similar conclusion can be obtained from the distributions of the relative empirical MSE in figures 7.6 and 7.7. Here, the relative MSE takes much lower values for the functions with a large effect. Additionally, the deviation of the distribution is smaller in this case. The difference between strong and weak functions is especially distinct if the true effect is wiggly (functions f_5/f_{11} and f_6/f_{12}). Additionally, in all cases when important covariates were removed from the model, these functions were among those with a small effect.
  These results (regarding bias and MSE) show, however, that difficulties with the selection and estimation of functions with a small effect did not only occur with variable selection algorithms but also with MCMC techniques which only had to choose appropriate degrees of smoothness.

- The span of the average estimates of the null functions is always below 0.03 and in most cases even below 0.02. Average estimated null functions are not shown for the Gaussian distribution. But they are similar to the estimates obtained for the Gamma simulation shown in figure 7.12.

algorithm	x_1	x_7
adaptive	0.76	0.83
adaptive/exact	0.78	0.81
exact	0.78	0.80
stepwise	0.79	0.82

Table 7.2: Gaussian distribution: Portion of replications in which variables x_1 or x_7 were correctly modelled by a linear effect.

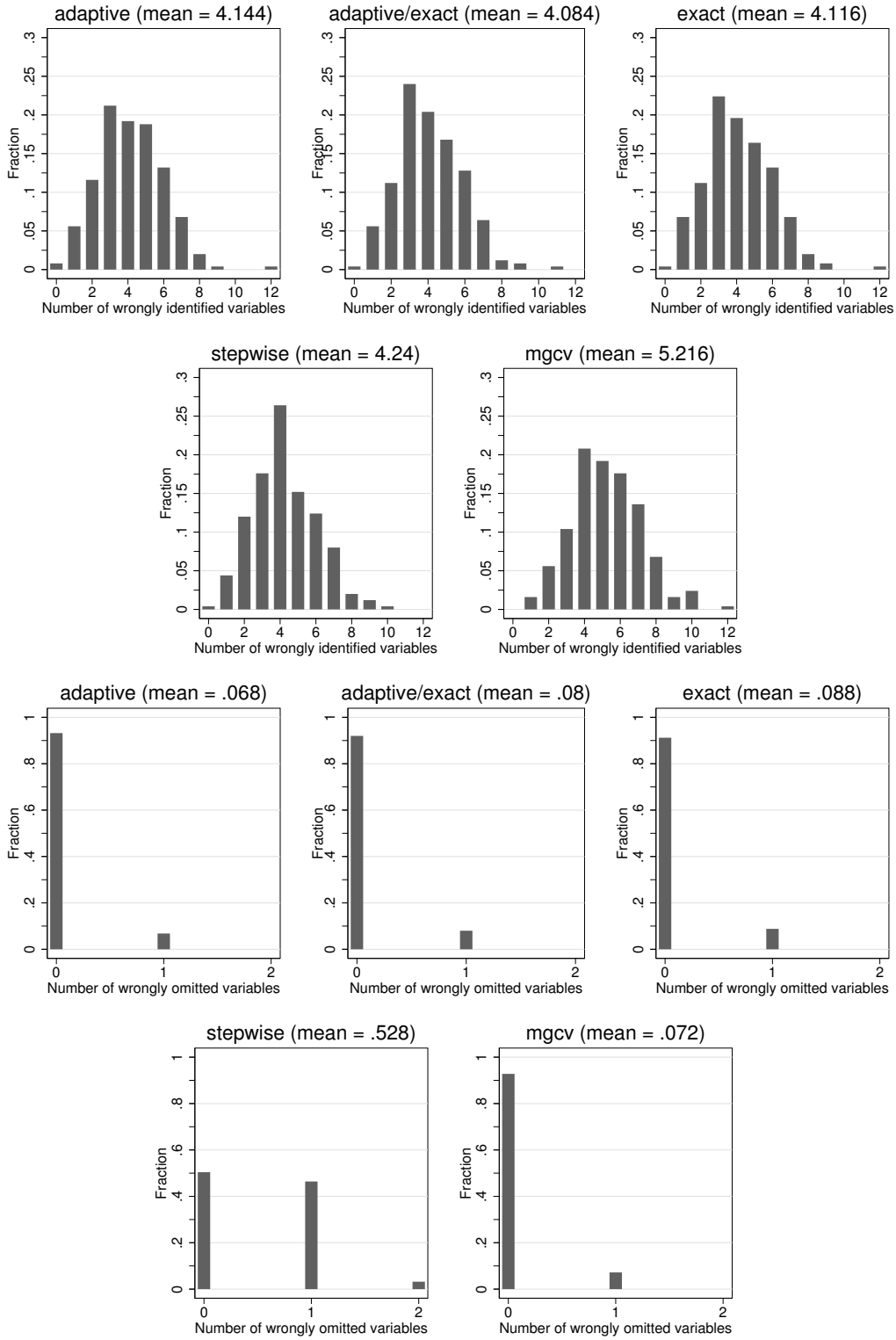

Figure 7.10: Gaussian distribution: Histograms for the distribution of the number of wrongly identified covariates (upper rows) and the number of wrongly omitted covariates (bottom rows). Wrongly identified means that both cases of mistakes are considered (i.e. relevant variables which were removed from the model or irrelevant variables which were included into the model).

7.1.3.2 Gamma distribution

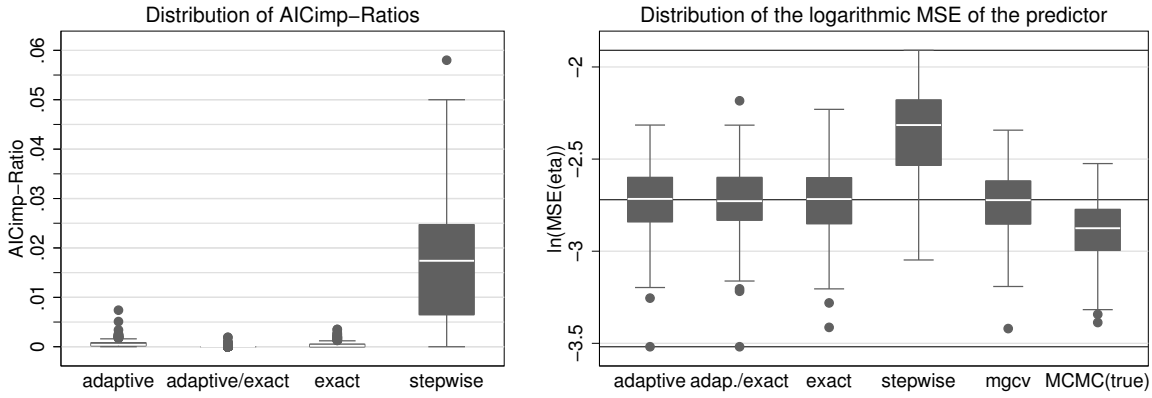

Figure 7.11: Gamma distribution: The left plot shows the distributions of ratio (7.1) for AIC_{imp} values. For a better comparison, the plot leaves out the extreme outlier (0.2) of the stepwise algorithm. The right plot compares the distributions of $\log(MSE(\eta))$*, where the extreme outlier (-0.29) of the stepwise algorithm is left out. The constant lines indicate the common minimum, median and maximum of all approaches excluding the outlier.*

algorithm	adaptive	adaptive/exact	exact	stepwise	mgcv
wrongly added	1.61	1.61	1.58	3.11	1.83
wrongly omitted	0.00	0.00	0.00	0.10	0.00
total	1.61	1.61	1.58	3.21	1.83

Table 7.3: Gamma distribution: Average numbers of wrongly identified variables.

The results obtained for the Gamma distributed response variables are essentially the same as for the Gaussian distribution. Therefore, we confine the results to the most important ones. Additionally, we show some figures not shown for the Gaussian simulation.

- In terms of ratio (7.1) for AIC_{imp} shown in figure 7.11, the difference between the stepwise algorithm and the other approaches is even greater than for the Gaussian simulation. There is no noteworthy difference between the algorithms derived from the coordinate descent method.
- Regarding the MSE of the predictor (figure 7.11) the stepwise algorithm performed worst. Between the other selection methods and mgcv there is no difference, whereas the true model (MCMC) achieved slightly better results.
- The results of the individual functions regarding average estimates and logarithmic relative MSE are very similar to the results shown in figures 7.8 to 7.9 and 7.6 to 7.7 for the Gaussian simulation. Hence, the respective conclusions apply here as well.
- Figure 7.12 exemplarily shows the average estimates and the empirical MSE for two of the eight null functions (for covariates x_{13} and x_{14}) for the adaptive search and

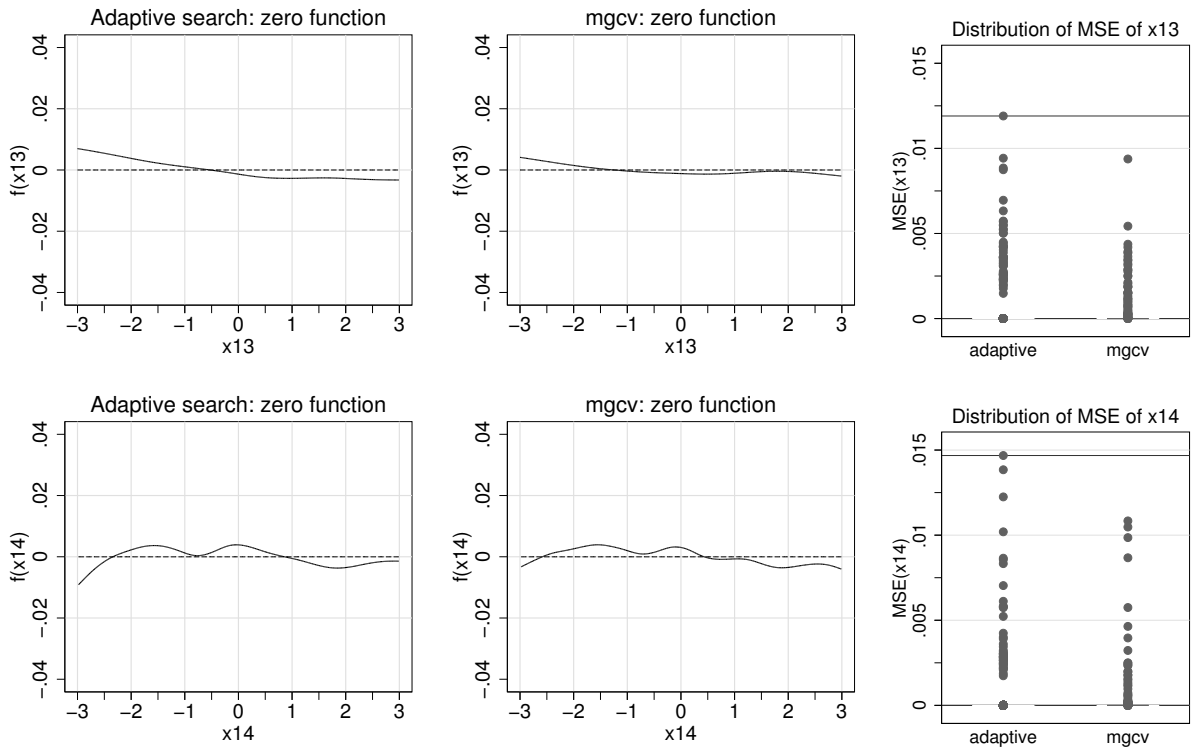

Figure 7.12: Gamma distribution: Average estimated functions (solid line) together with the true underlying null functions (dashed line) for adaptive search (left column) and mgcv package (middle column). The right plots compare the distributions of the empirical MSE for the same functions. The constant lines indicate the common maximum.

mgcv. There is no difference between the two approaches. The empirical MSE is equal to zero in at least 75% of replications indicating that the respective variable was correctly removed from the model. The average estimates are close to zero.

- Table 7.3 shows the average numbers of wrongly identified variables. The stepwise algorithm was the only approach that removed important variables from the model and has the highest number of mistakes. Between the other selection algorithms and mgcv there is no notable difference.
- The runtime the algorithms needed to select and estimate all 250 replications is shown in table 7.4. The results are also similar to the Gaussian distribution. The adaptive search was again by far the fastest approach by nearly identical other results.

algorithm	adaptive	adaptive/exact	exact	stepwise	mgcv	mcmc (true)
runtime	1:05	13:02	33:26	37:47	204:55	12:34

Table 7.4: Gamma distribution: Computing times in hours for all 250 replications.

7.1.3.3 Binomial distribution

For the Binomial simulation we also wanted to show results of mgcv for a comparison with our approaches. However, there sometimes occurred convergence problems so that we did not obtain results for all replications. Furthermore, mgcv needs nearly two hours for the estimation of one replication. For these reasons, we cannot show the results of mgcv here. The results of the Binomial simulation are in most respects comparable to the results of Gaussian and Gamma simulation. Therefore, we restrict to the most important results here. If not mentioned otherwise, the same conclusions apply as for Gaussian and Gamma simulation.

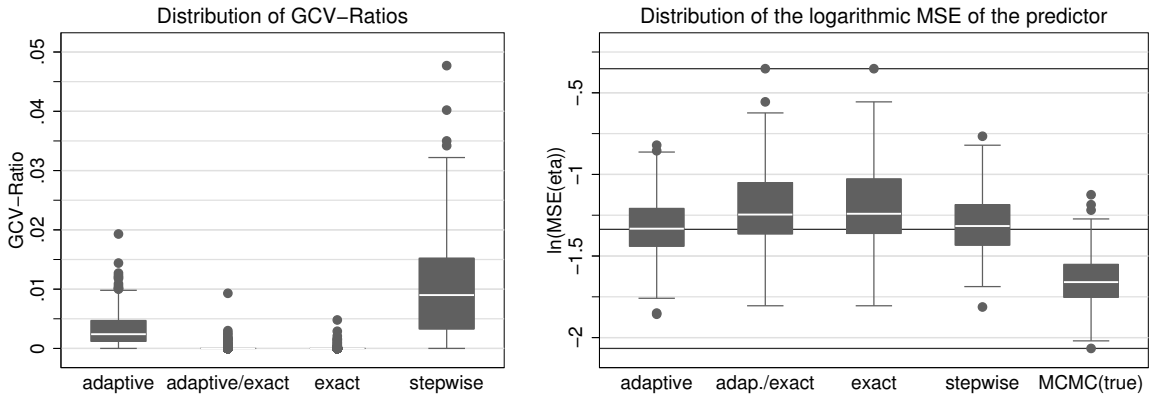

Figure 7.13: Binomial distribution: The left plot shows the distributions of ratio (7.1). The right plot shows the distributions of $\log(MSE(\eta))$ *for all different approaches. The constant lines indicate the common minimum, median and maximum calculated over all approaches contained in the plot.*

- The results regarding ratio (7.1) for the GCV values (compare figure 7.13) are comparable to those of the other distributions: the stepwise algorithm produced the worst results whereas exact and adaptive/exact search nearly always selected the best model. The median for the adaptive search is 0.0024. Hence, the differences between adaptive search and adaptive/exact and exact search are only small.

- In terms of logarithmic MSE of the overall predictor (compare figure 7.13), the MCMC techniques conditional on the true model performed clearly better than any of the selection algorithms. Exact and adaptive/exact search yielded slightly worse results than adaptive search and stepwise algorithm although they obtained better GCV values. This indicates, that the minimal GCV value does not correspond with the best model.

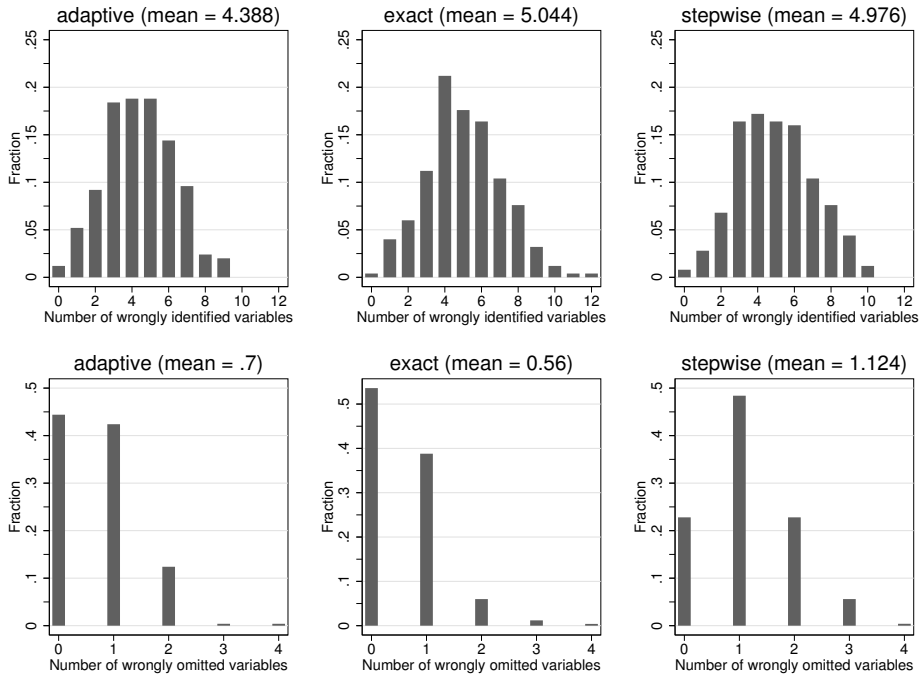

Figure 7.14: Binomial distribution: The upper row shows histograms for the distribution of the number of wrongly identified covariates. Wrongly identified means that either important variables were removed from the model or that unimportant variables were included into the model. The bottom row shows histograms for the distribution of the number of wrongly omitted covariates, i.e. only important variables that were removed from the model are considered here.

- Regarding the number of wrongly identified variables, the adaptive search yielded better results than stepwise algorithm and exact search. For each selection algorithm the total number of mistakes was here slightly larger than for the Gaussian simulation. The differences to the Gaussian results are mainly due to the larger number of wrongly omitted covariates.
- Once again, the adaptive search was the most efficient estimation approach regarding the time needed for selecting and estimating all 250 replications (see table 7.5). Exact search and stepwise algorithm needed considerably more time than MCMC techniques conditional on the true model.

algorithm	adaptive	adaptive/exact	exact	stepwise	mgcv	mcmc (true)
runtime	0:58	11:26	22:15	39:00	—	13:55

Table 7.5: Binomial distribution: Computing times in hours for all 250 replications each.

7.1.3.4 Poisson distribution

Like with the Binomial simulation there occurred convergence problems with mgcv for some replications. Furthermore, mgcv needs even more then two hours for the estimation of one replication. So again, we cannot show the results of mgcv here.
The results of the Poisson simulation study can be summarised as follows:

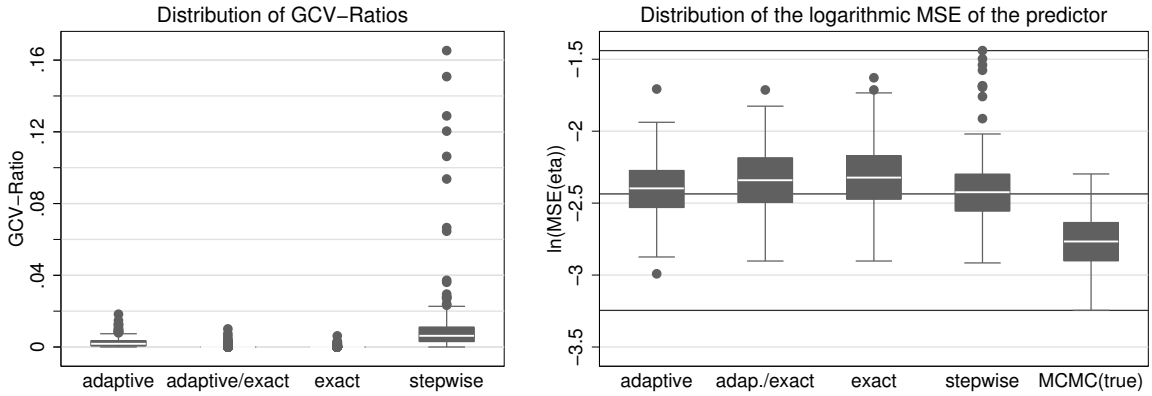

Figure 7.15: Poisson distribution: The left plot shows the distributions of ratio (7.1). The right plot shows the distributions of $\log(MSE(\eta))$ *for all different approaches. The constant lines indicate the common minimum, median and maximum calculated over all approaches.*

- Regarding ratio (7.1) for the GCV values shown in figure 7.15 we obtained the same results as for all other distributions: the stepwise algorithm performed worst whereas the exact and the adaptive/exact search nearly always found the best model. Again, the median of about 0.002 for the adaptive search indicates that the differences to the best model are only small.

- In terms of logarithmic empirical MSE for the predictor (compare figure 7.15), the results obtained conditional on the true model are better than those of the selection algorithms. Like for the Binomial simulation, exact and adaptive/exact search yielded slightly worse results than adaptive search and stepwise algorithm, although they obtained better GCV values. Adaptive search and stepwise algorithm yielded comparable results with the exception of a few outliers with higher values for the stepwise algorithm.

- Regarding the estimates of the individual functions, particularly of the weak functions, the results were here slightly worse than for all other distributions. This applies likewise to the results of the selection algorithms and those conditional on the true model. Partly, this can be attributed to the fact that the influence of each function

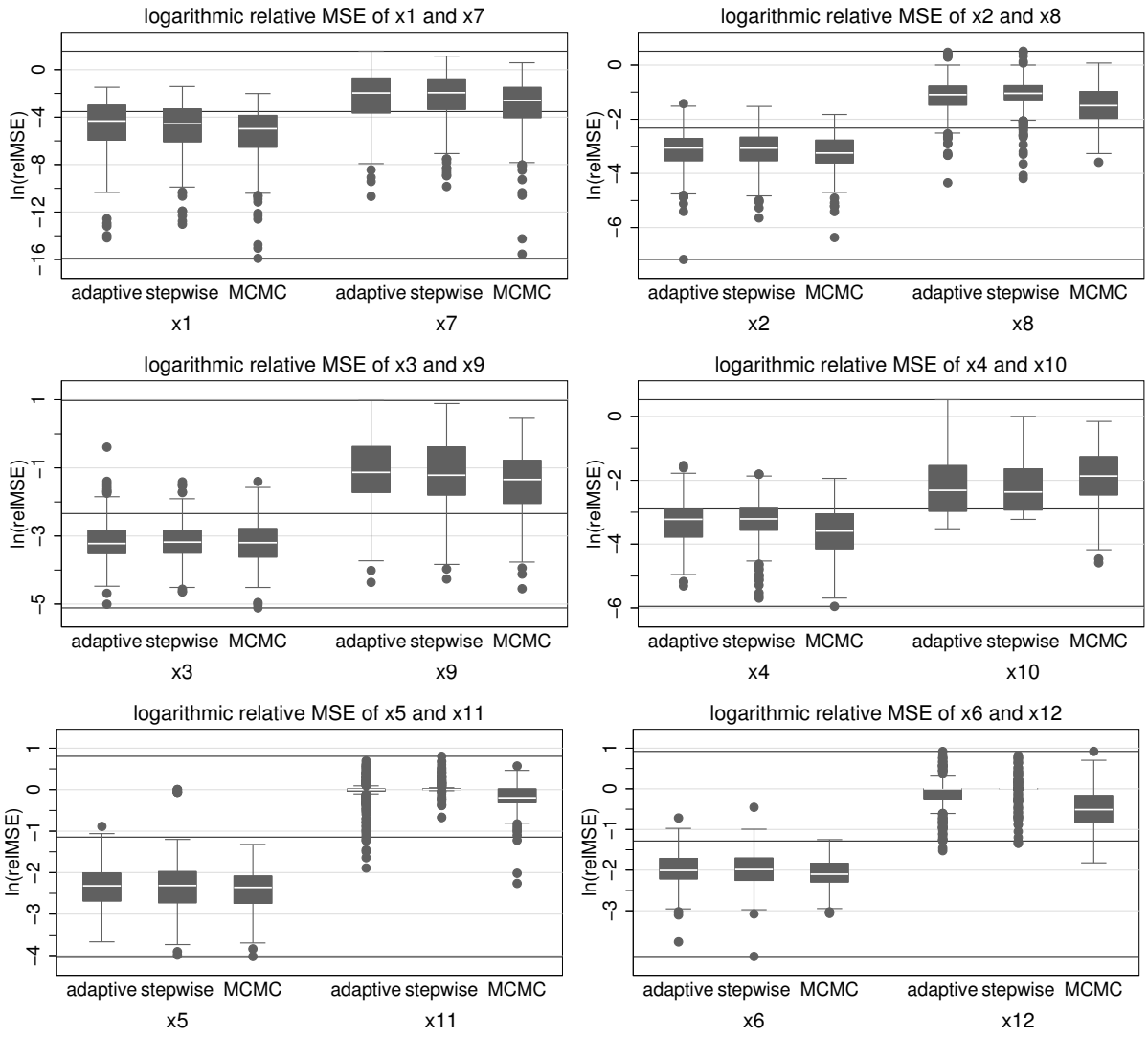

Figure 7.16: Poisson distribution: Distributions of the relative logarithmic MSE for the important functions.

is here only half as strong as with all other distributions. Figure 7.16 shows the logarithmic relative MSE values which are larger than those of the Gaussian simulation. The bad results for functions f_{11} and f_{12} are due to the fact that these functions were often removed from the model, particularly by the stepwise algorithm.

- Figure 7.17 shows the average estimated functions for the adaptive search. The functions are more biased than for the Gaussian simulation, especially the wiggly functions. The results of MCMC techniques conditional on the true model are only slightly less biased than those of the adaptive search and are not shown.
- The results regarding the zero functions are comparable to the other distributions

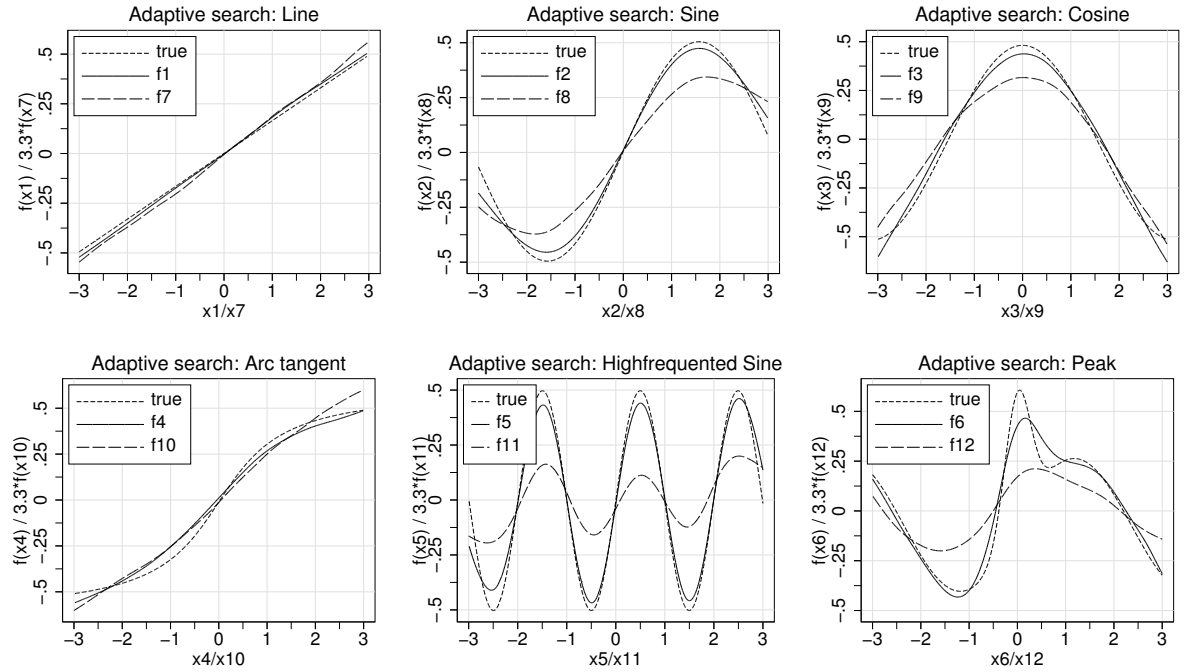

Figure 7.17: Poisson distribution: Average estimates of the adaptive search. By multiplying weak functions with factor 3.3, both functions of the same type are plotted on the same scale.

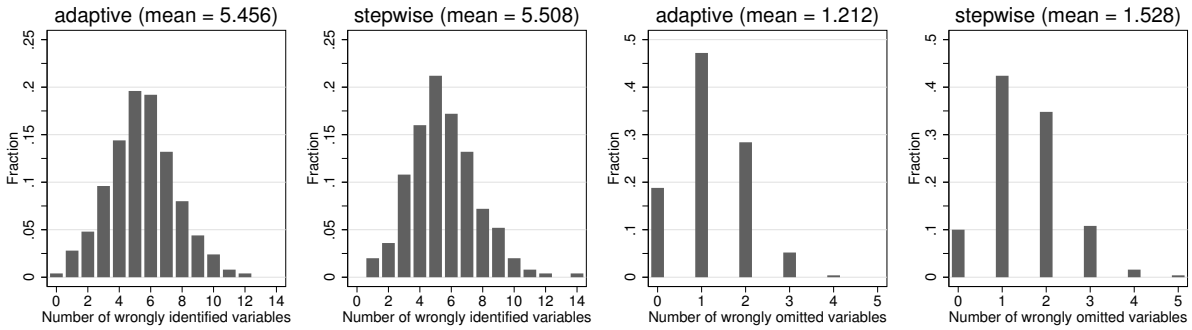

Figure 7.18: Poisson distribution: Histograms for the distribution of the number of wrongly identified covariates (left) and the number of wrongly omitted covariates (right). Wrongly identified means that both cases of mistakes are considered (i.e. relevant variables which were removed or irrelevant variables which were included into the model).

and are not shown here.

- Figure 7.18 shows the number of wrongly identified terms for stepwise algorithm and adaptive search. The results of exact and adaptive/exact search (not shown) are similar to those of the adaptive search. Here, the total number of mistakes is considerably higher than for the Gaussian simulation and even higher than for the Binomial simulation. This is due to the increased number of important terms that

were removed from the model. Most often, the weak wiggly functions f_{11} and f_{12} were not recognised.

- In terms of computing time (compare table 7.6) the adaptive search was the most efficient selection method again.

algorithm	adaptive	adaptive/exact	exact	stepwise	mgcv	mcmc (true)
runtime	0:53	10:35	16:56	48:27	—	11:59

Table 7.6: Poisson distribution: Computing times in hours for all 250 replications each.

7.2 Simulation of a multinomial logit model

For the simulation of a multinomial logit model with three possible outcomes, i.e. $k = 2$, we used the 12 functions of the additive simulation study (shown in figure 7.1) and constructed two predictors as

$$\begin{aligned}\eta^{(1)} &= f_1(x_1) + f_3(x_3) + f_5(x_5) + f_8(x_8) + f_{10}(x_{10}) + f_{12}(x_{12}),\\ \eta^{(2)} &= f_2(x_2) + f_4(x_4) + f_6(x_6) + f_7(x_7) + f_9(x_9) + f_{11}(x_{11}).\end{aligned}$$

Hence, each predictor contains the same number of functions and includes both weak and strong functions. We created $R = 200$ replications with $n = 700$ observations each. Each observation y_i consists of $m = 3$ repetitions, i.e. $y_i \sim M(3, (\pi_i^{(1)}, \pi_i^{(2)})')$, with probabilities

$$\begin{aligned}\pi_i^{(1)} &= \frac{\exp(\eta_i^{(1)})}{1 + \exp(\eta_i^{(1)}) + \exp(\eta_i^{(2)})},\\ \pi_i^{(2)} &= \frac{\exp(\eta_i^{(2)})}{1 + \exp(\eta_i^{(1)}) + \exp(\eta_i^{(2)})}.\end{aligned}$$

For both predictors, the selection algorithms had to select the relevant covariates out of variables x_1–x_{15} where covariates x_{13}–x_{15} have no influence on either predictor. The modelling possibilities for the covariates were the same as described in section 7.1.
For this distribution, we compare the results of the adaptive, exact and adaptive/exact search and the stepwise algorithm. As reference, we estimated the true model using the adaptive search. True model means that for each category we used only functions with an influence on the respective predictor. Thereby, we estimated linear functions using a linear fit and specified nonlinear functions as nonlinear so that merely appropriate degrees of freedom had to be selected. The selection was always performed using AIC.

algorithm	adaptive	adaptive/exact	exact	stepwise	true model
runtime	1:04	19:20	44:25	85:54	0:32

Table 7.7: Multinomial logit model: Computing times in hours for the 200 replications.

The results can be summarised as follows:

- In terms of ratio (7.1) for AIC values, figure 7.19 shows similar results as the plots for the univariate simulations in section 7.1: the stepwise algorithm yielded the worst results followed by the adaptive search. But as described for the results of the normal distribution, the differences between the approaches based on the coordinate descent method are only small.

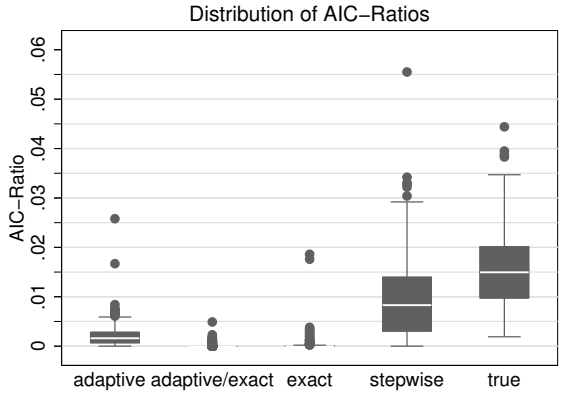

Figure 7.19: Distributions of ratio (7.1) for all different approaches.

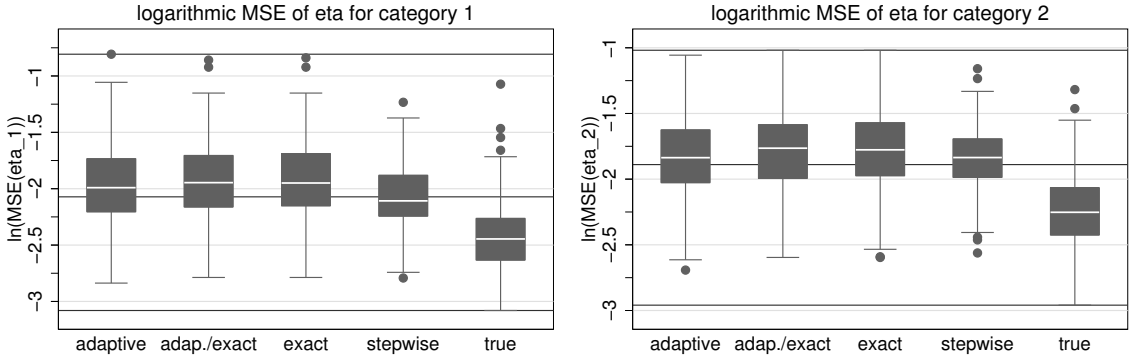

Figure 7.20: Distributions of $\log(MSE(\eta^{(1)}))$ *and* $\log(MSE(\eta^{(2)}))$ *for all different approaches. The constant lines indicate the common minimum, median and maximum calculated over all approaches.*

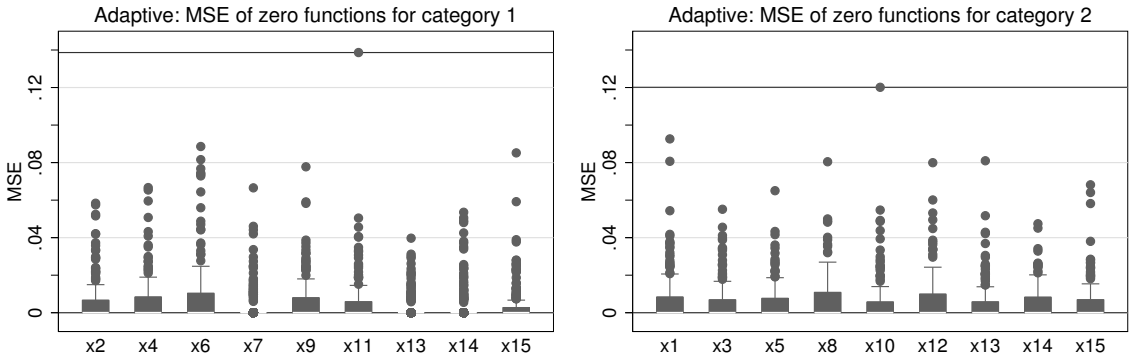

Figure 7.21: Distributions of the empirical MSE for the null functions.

- Regarding the empirical logarithmic MSE of the two predictors (figure 7.20), there was practically no difference between the selection algorithms. As expected, the results conditional on the true model were better.

- The average estimates of the important functions (not shown) are slightly more biased than those of the Gaussian simulation (see figures 7.8 and 7.9). The function estimates based on the true model were slightly less biased than those obtained by the selection algorithms. Again, the estimates are better for the strong than for the weak functions. The same picture arises when considering the empirical MSE of the important functions (not shown). There is practically no difference between the different selection algorithms, whereas the estimates conditional on the true model were slightly better. The only real difference resulted for function f_{11} of the second category (weak highly frequented sine). In 101 replications the stepwise algorithm excluded this function from the model whereas the other algorithms included it in more than 160 cases.
- The span of average estimates of the null functions always lies below 0.06. For the adaptive search figure 7.21 shows the distributions of the empirical MSE for all null functions. The MSE values are all sufficiently small and there is no difference between functions that are important for the other category and completely unimportant functions.
- Table 7.8 compares the average numbers of wrongly identified variables separately for both categories. There are only small differences between the algorithms. The stepwise algorithm seems to select slightly sparser models because the number of wrongly removed variables is larger and the number of wrongly added variables smaller.
- The adaptive search was by far the fastest approach (compare table 7.7) and performed the selection for all replications in one hour. In contrast, the stepwise algorithm needed more than three days. Even the adaptive/exact search took nearly 20 hours for the selection so that, altogether, the adaptive search is the algorithm which is most preferable.

algorithm	adaptive	adaptive/exact	exact	stepwise
category 1: wrongly added	2.45	2.24	2.21	1.92
category 1: wrongly omitted	0.24	0.24	0.22	0.41
category 1: total	2.69	2.48	2.43	2.33
category 2: wrongly added	2.61	2.52	2.55	2.28
category 2: wrongly omitted	0.30	0.27	0.28	0.58
category 2: total	2.91	2.79	2.83	2.85
total	5.60	5.27	5.26	5.18

Table 7.8: Multinomial logit model: Average numbers of wrongly identified variables.

7.3 Simulation of a geoadditive mixed model

For this simulation study we used a geoadditive mixed model which contains a smooth spatial function and a random intercept in addition to six nonlinear functions of continuous covariates. For the nonlinear functions of continuous variables we used functions f_1 to f_6 from the simulation study in section 7.1 which are shown in figure 7.1. The functions are indicated by the same numbers in the geoadditive simulation study. The smooth spatial function and the random effect are both shown in figure 7.22. For the spatial effect we used the 309 regions of West-Germany and created a two–dimensional function using the centroids (r_1, r_2) of the regions as variables. The spatial function is then given by

$$f_{spat} = \sin(r_1 \cdot r_2) + 0.1483,$$

where r_1 is the value of a centroid in east–west direction and r_2 its value in north–south direction. Both variables r_1 and r_2 had been centered and standardised before. The function is centered around zero by the value 0.1483. For each region we generated three observations so that we have 729 observations for the geoadditive simulation. Then, we generated a group variable *ind* with twenty individuals for a random effect. The individuals were randomly assigned to the observations in such a way that there are either 46 or 47 observations per individual. The random effect was created according to a normal distribution with mean zero and a standard deviation of 0.4.
The span between minimum and maximum of these two functions again amounts to 2 (like for the continuous variables) so that all functions have an equally strong influence on the predictor. The predictor takes the form

$$\eta = \sum_{j=1}^{6} f_j(x_j) + f_{spat}(region) + f_{rand}(ind).$$

Additionally, we used six continuous covariates without effect. The number of replications is $R = 250$ and we assumed a Gaussian model with a standard deviation of $\sigma = 1.1$.
For the modelling of the spatial function a Markov random field was used with possible degrees of freedom $\{0, 10, 20, \ldots, 300\}$ and $df = 10$ for the basis model. The effect of the continuous variables were represented by cubic P–splines with 22 basis functions and possible degrees of freedom $\{0, 1, 2, \ldots, 21\}$ where the linear fit $df = 1$ was used for the basis model. The random effect was represented by an i.i.d. Gaussian random effect with possible degrees of freedom $\{0, 1, 2, \ldots, 19\}$. For the basis model we used a random effect with $df = 1$. For all functions, $df = 0$ corresponds to the removal of the respective function from the model.
To analyse the results we computed average function estimates, empirical MSE, empirical bias and the ratio of AIC_{imp} values. We draw the following conclusions:

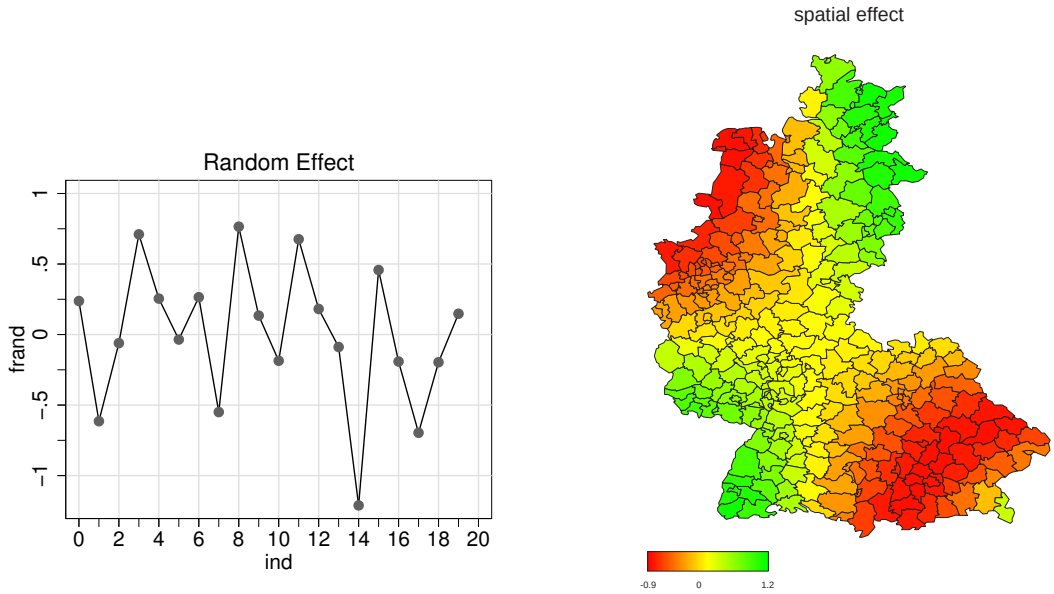

Figure 7.22: True smooth spatial function f_{spat} and random effect f_{rand} used in the geoadditive simulation study.

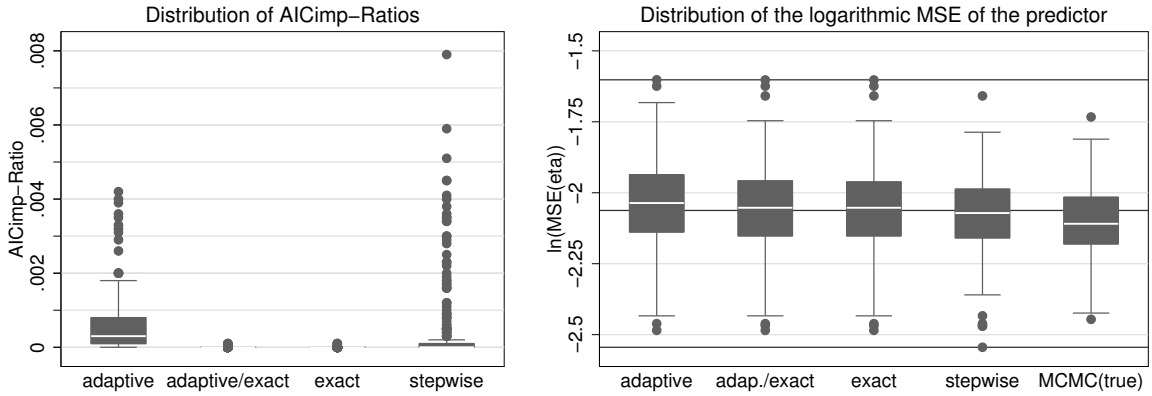

Figure 7.23: The left plot shows the distributions of ratio (7.1). The right plot shows the distributions of $\log(MSE(\eta))$ *for all different approaches. Here, the constant lines indicate the common minimum, median and maximum calculated over all approaches.*

- In terms of ratio 7.1 of AIC_{imp} values shown in figure 7.23 the adaptive search performed slightly worse than the exact and adaptive/exact search and even than the stepwise algorithm. For the adaptive search, the median of the distribution, however, is just about 0.00025 indicating that the difference to the best model is only 0.025% of the difference between the best and the empty model. Hence, in this respect, there is practically no difference between the algorithms.
- Regarding the empirical MSE of the predictor (compare figure 7.23), there is no

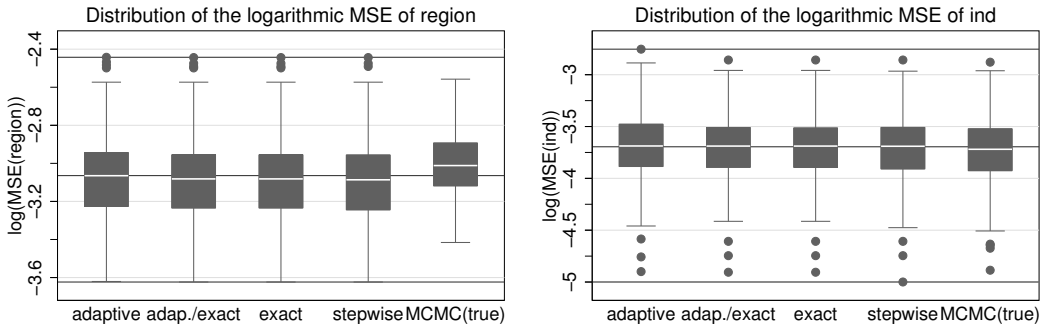

Figure 7.24: Distributions of the logarithmic MSE for the random effect and the spatial function. The constant lines indicate in each case the common minimum, median and maximum calculated over all approaches.

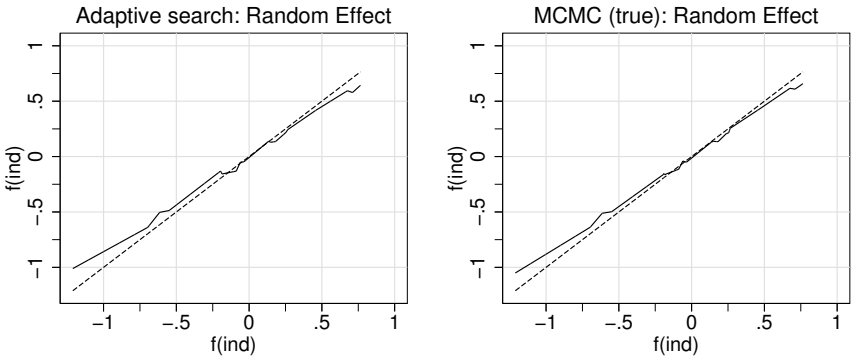

Figure 7.25: Estimated random effects (solid line) together with the true underlying random effect (dashed line). All functions are plotted against the true random effect.

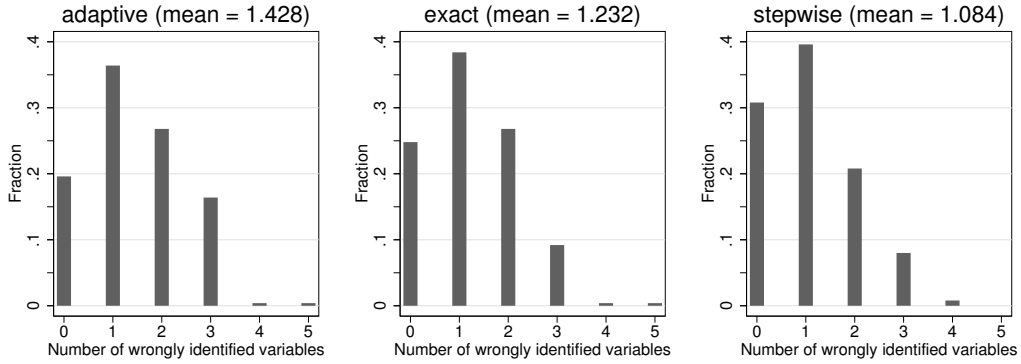

Figure 7.26: Histograms for the distribution of the number of wrongly identified covariates. Wrongly identified means in this case unimportant covariates that were included into the model as there were never any important variables removed.

difference between the selection algorithms. Only the estimation conditional on the

true model by MCMC techniques yielded slightly better results.

- For the individual nonlinear functions f_1 to f_6, the logarithmic MSE values show no difference between the different approaches (not even for MCMC techniques conditional on the true model) and, therefore, are not shown. The same applies to the logarithmic MSE for random effect and spatial function (compare figure 7.24). The only exception are the values of *MCMC(true)* for the spatial function which are in average slightly larger than for the other approaches.
- The average estimated functions $\hat{f}_1, \ldots, \hat{f}_6$ are very similar to the respective estimated functions of the additive simulation study shown in figures 7.8 and 7.9. Therefore, they are not shown. For some functions there is a small bias which is slightly larger for the adaptive search than for the true model. The largest bias was obtained for function f_6 (*peak*). The average estimated random effects together with the true random effect are shown in figure 7.25. Here, the bias from the adaptive search is not distinguishable from the bias obtained from the true model. For the spatial effect, average estimated functions and empirical bias are shown in figure 7.27. The bias of the spatial function is slightly larger for the adaptive search than for *MCMC(true)*.
- Figure 7.26 shows the number of unimportant variables which were wrongly added to the model wheres neither approach removed important variables from the model. Again, the results are very similar where the adaptive search yielded slightly worse results and the stepwise algorithm slightly better results than exact and adaptive/exact search. The results of exact and adaptive/exact search are identical.
- The computing times displayed in table 7.9 yielded greater differences between the selection algorithms than all other results. The adaptive search was by far the fastest approach whereas the stepwise algorithm again took the most time.

algorithm	adaptive	adaptive/exact	exact	stepwise	mcmc (true)
runtime	0:59	2:13	2:49	5:04	4:53

Table 7.9: Computing times in hours for all 250 replications each.

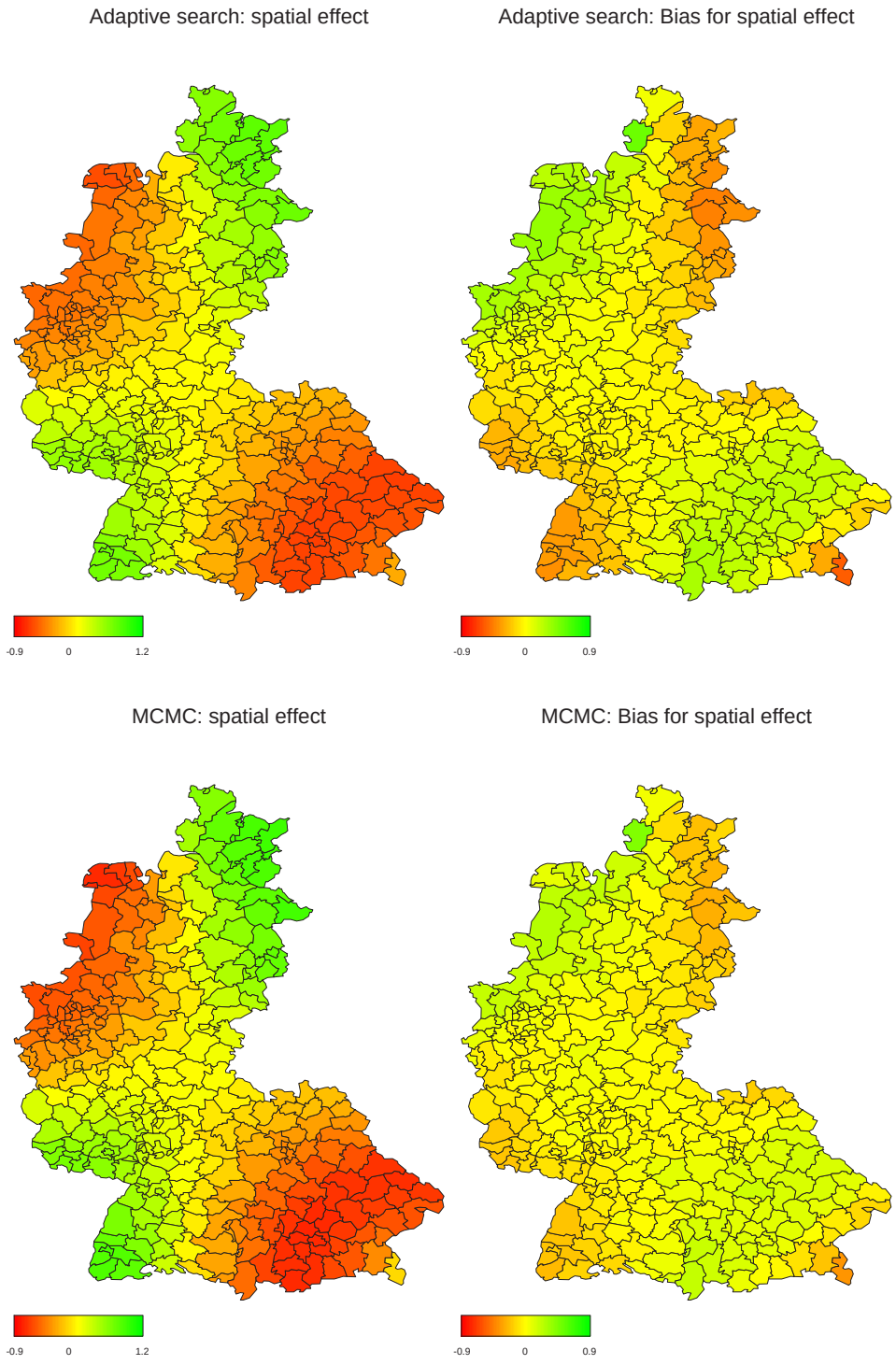

Figure 7.27: Average estimated spatial functions (left column) and their empirical bias (right column) for the adaptive search (top row) and the true model estimated by MCMC techniques (bottom row). Yellow indicates regions without bias.

7.4 Simulation of a varying coefficient model

For this simulation study we used a varying coefficient model which imitates the kind of models analysed in chapter 8 where nonlinear effects can be different across two groups (boys and girls in the example). That means, we consider varying coefficients of the form $g(v)s$, where s is a two–categorical variable. Here, the values for s where chosen uniformly from $\{-1; 1\}$.
The predictor contains two smooth spatial functions: the average effect f_{spat} and the varying effect g_{spat}. The underlying map again consisted of the 309 regions of West–Germany and two–dimensional functions were calculated by using the centered and standardised centroids (r_1, r_2) of the regions as variables. The spatial functions are given by

$$\begin{aligned} f_{spat} &= (\sin(r_1 \cdot r_2) + 0.1483)/0.555, \\ g_{spat} &= (r_1 + r_2)/2.409, \end{aligned}$$

where r_1 is the value of a centroid in east–west direction and r_2 its value in north–south direction. Both functions are centered around zero and shown in figure 7.29. For each region we generated three observations so that, altogether, we have 729 observations for the VC simulation.
In addition to the spatially varying coefficient we used two nonlinear varying coefficients shown in figure 7.28. Moreover, the model contains two nonlinear functions that do not vary across the two groups (also shown in figure 7.28) and two continuous covariates without any influence on the response. The values for the six continuous covariates were chosen independently of each other and uniformly from the range $[-3; 3]$ but rounded to two decimal places afterwards. All functions f_j, $j = 1, \ldots, 4, spat$, were chosen such that $\sigma_{f_j} = 1$ whereas the effect of functions g_j, $j = 1, 3, spat$, is weaker with $\sigma_{g_j} = 0.5$. The true predictor takes the form

$$\eta = f_1(x_1) + g_1(x_1)s + f_2(x_2) + f_3(x_3) + g_3(x_3)s + f_4(x_4) + f_{spat}(region) + g_{spat}(region)s.$$

Since the categorical variable s is effect–coded, effects for $s = 1$ are obtained by $f_j + g_j$ whereas those for $s = -1$ are given by $f_j - g_j$. Hence, the main effects f_j represent the average estimate of both categories and functions g_j the deviation of this average effect and the individual effects. The number of replications is $R = 250$ and we assumed a Gaussian model with a standard deviation of $\sigma_\varepsilon = 0.82$ leading to a signal–to–noise ratio of $\sigma_\eta/\sigma_\varepsilon = 3$.
Together with the unimportant terms the most general possible predictor is

$$\eta = \gamma_0 + f_1(x_1) + g_1(x_1)s + \ldots + f_6(x_6) + g_6(x_6)s + f_{spat}(region) + g_{spat}(region)s + \gamma_s\, s.$$

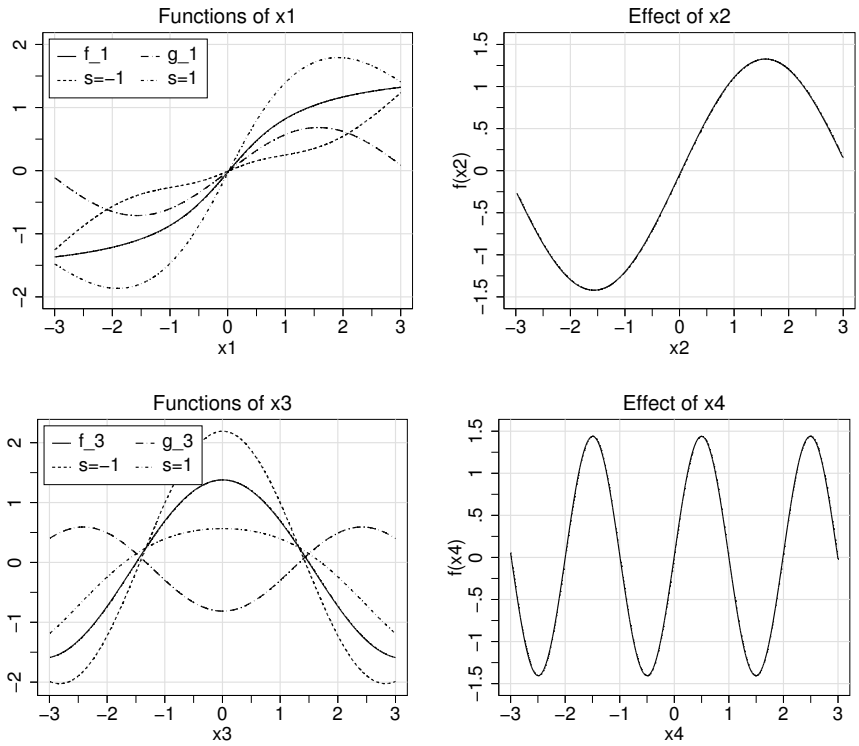

Figure 7.28: True nonlinear functions used in the VC simulation study.

For this simulation study we compared the adaptive, adaptive/exact and exact search with the stepwise algorithm, the mgcv package and the fully Bayesian approach via MCMC techniques conditional on the true model. For each of the two spatial functions f_{spat} and g_{spat} we used a two–dimensional P–spline with $12^2 = 144$ basis functions and a second order random walk penalty. The possible degrees of freedom were given by $\{0, 1, 5, 10, \ldots, 120\}$. (As an alternative, we also tried Markov random fields for the spatial functions but the results were worse and not directly comparable to those of mgcv.) For the one–dimensional functions we used P–splines with 22 basis functions, a second order random walk penalty and possible degrees of freedom $\{0, 1, 2, \ldots, 21\}$. For mgcv we used cubic smoothing splines instead of P–splines with 22 basis functions for univariate functions and 70 basis functions for the spatial functions. The selection was based on AIC_{imp} or on GCV with $\alpha = 1.4$ for mgcv, respectively. For the MCMC techniques we used every 20th sample for the calculation of estimates where the first 4000 samples presented the burn–in phase. Altogether, we used 1000 samples for the calculation of estimates.
The results lead to the following conclusions:

- In terms of ratio (7.1) (compare figure 7.30) there are only small differences between the selection algorithms: adaptive search and stepwise algorithm performed a bit

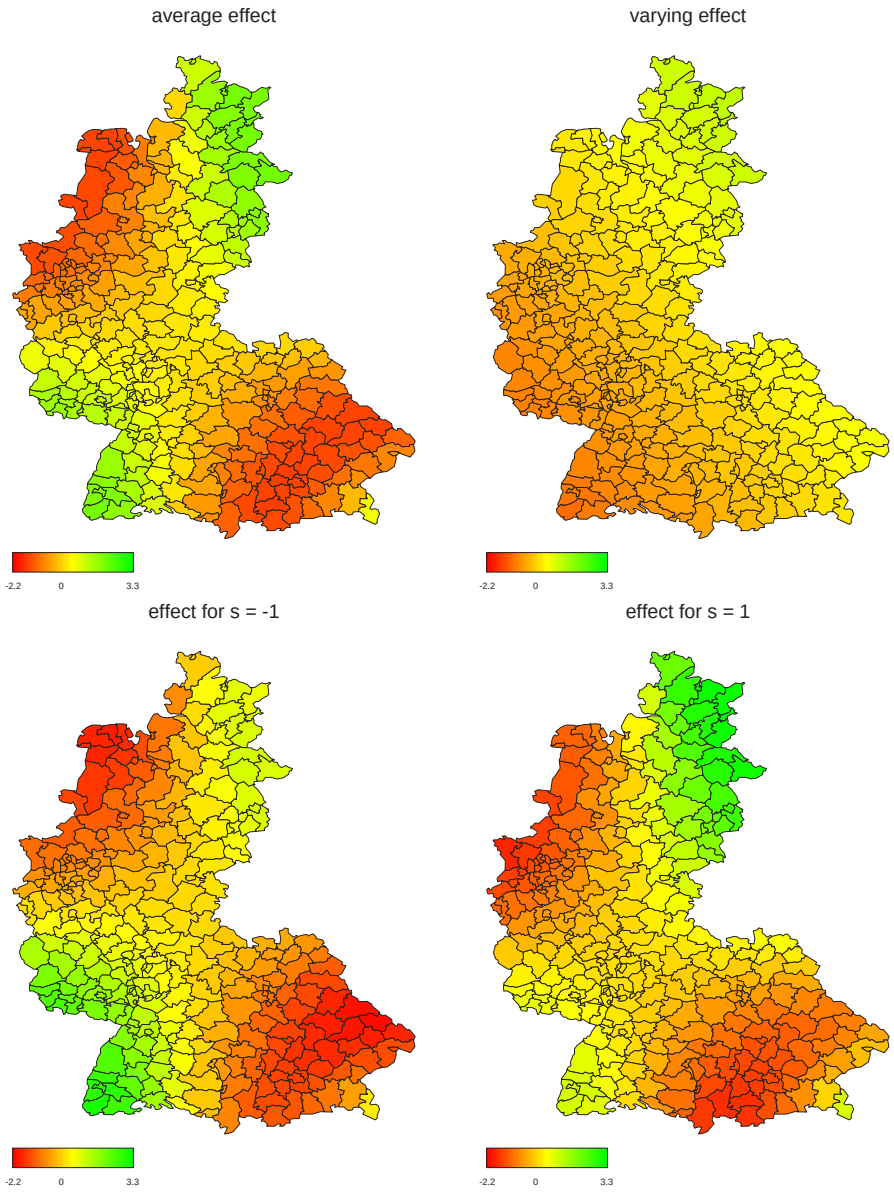

Figure 7.29: True smooth spatial functions used in the VC simulation study.

worse whereas adaptive/exact and exact search nearly always found the best model.

- Regarding the empirical MSE of the predictor shown in figure 7.30, there is no notable difference between the approaches with the exception of MCMC conditional on the true model: this approach performed slightly better than the rest.
- Regarding the estimates of the individual functions there are no differences between the approaches either. The only exceptions are the spatial functions where mgcv performed for f_{spat} worse than all other approaches but better for g_{spat}. Altogether,

algorithm	adaptive	adaptive/exact	exact	stepwise	mgcv	MCMC (true)
runtime	0:07	0:18	0:26	0:42	3:05	1:31

Table 7.10: Computing times in hours for the first 25 replications each.

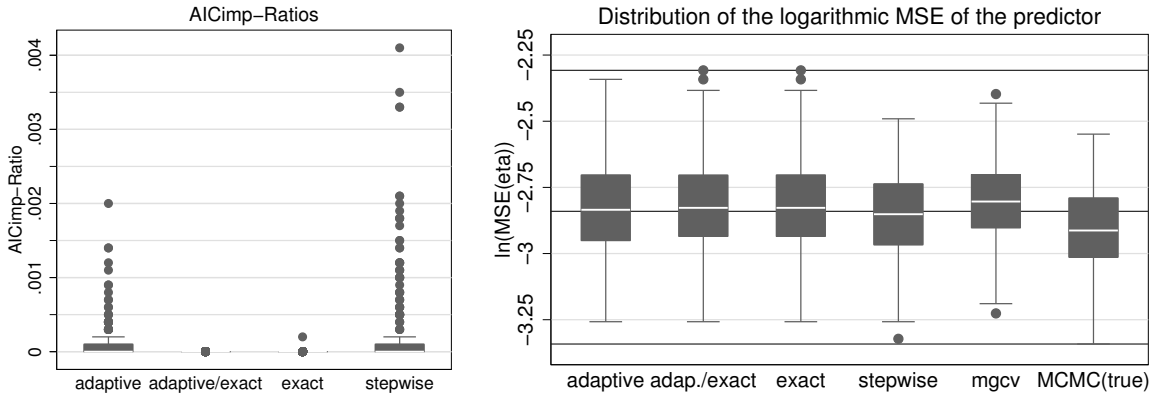

Figure 7.30: Distributions of ratio (7.1) (left plot) and distributions of $\log(MSE(\eta))$ *(right plot) for all different approaches.*

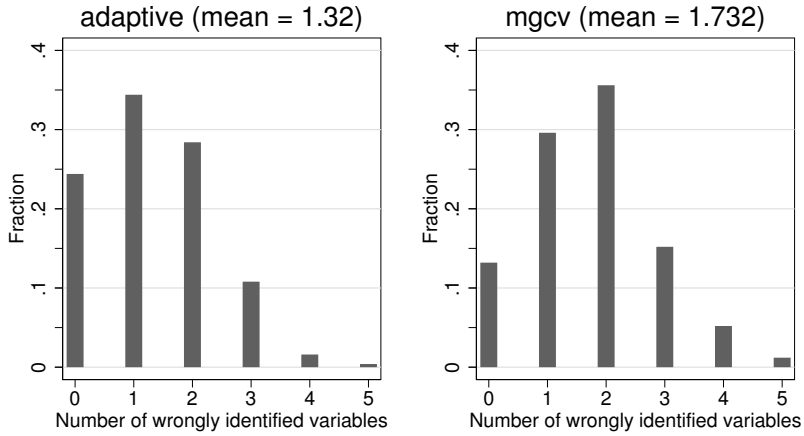

Figure 7.31: Histograms for the distribution of the number of wrongly identified covariates; here only irrelevant variables which were incorrectly included into the model.

the average estimated important functions are only slightly biased (compare figures 7.33–7.35) and the average estimates of the unimportant functions are nearly zero (not shown). The empirical MSE of the unimportant functions (not shown) is never above 0.02 indicating that individual estimated functions are close to zero. Each unimportant function was removed from the model in at least 72% and at most 80.4% of replications by the adaptive search with similar values for the other selection methods.

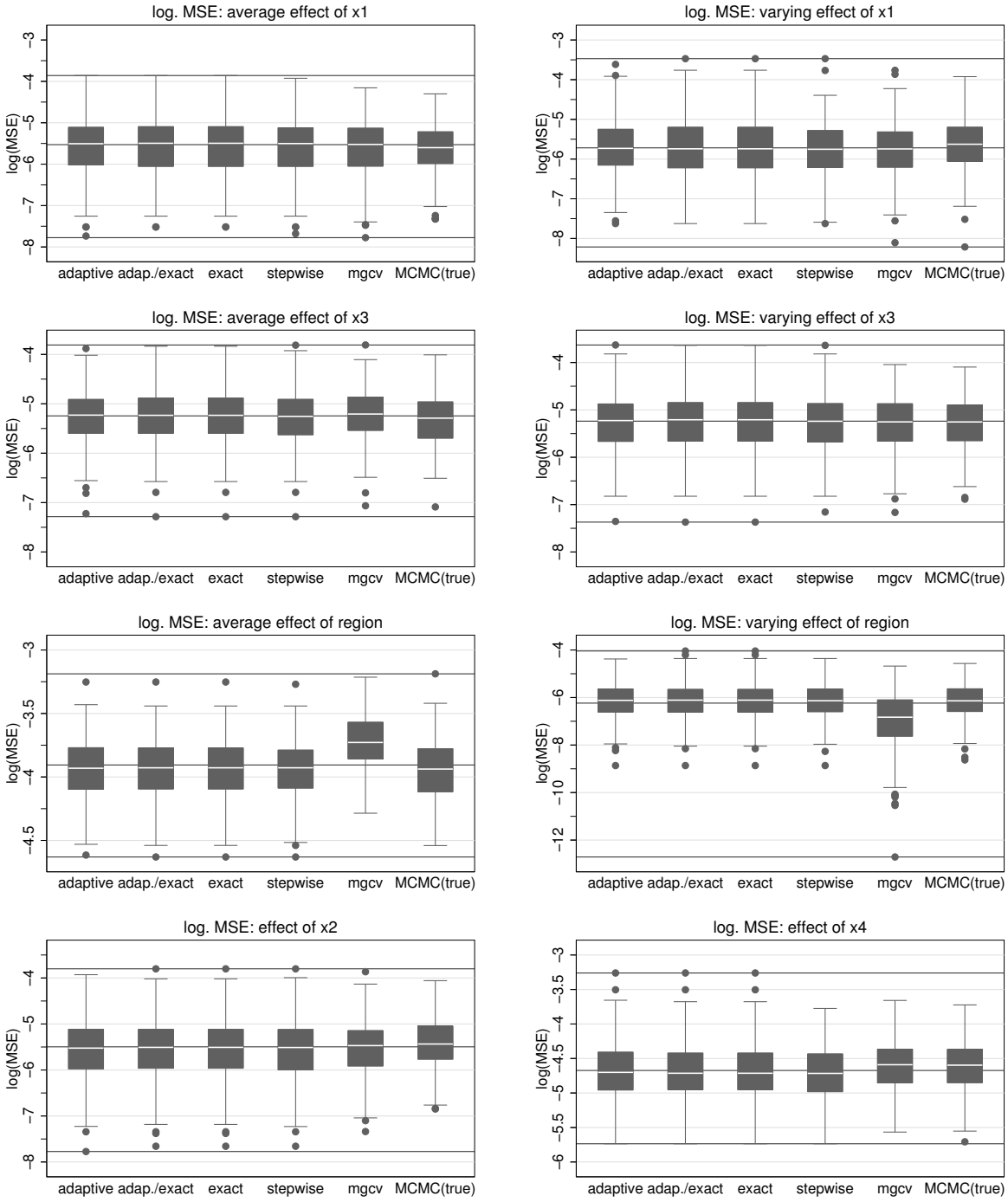

Figure 7.32: Distributions of the logarithmic MSE for the individual functions. The constant lines indicate in each case the common minimum, median and maximum calculated over all approaches.

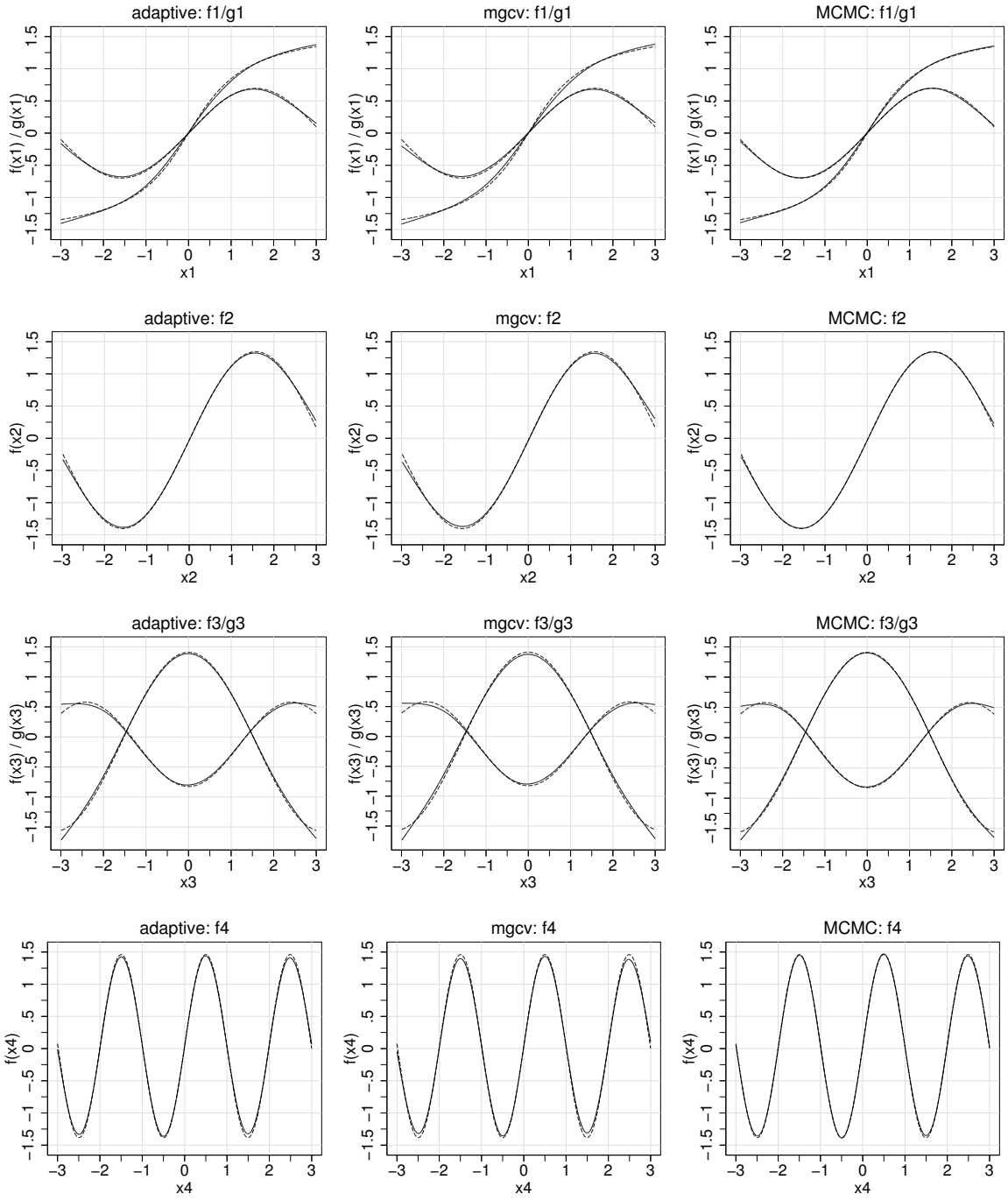

Figure 7.33: Average estimated functions (solid line) together with the true function (dashed line) for adaptive search (left column), mgcv package (middle) and MCMC techniques (right column).

- Figure 7.31 shows the number of wrongly identified terms of the adaptive search and mgcv where mgcv made slightly more wrong decisions. The other selection

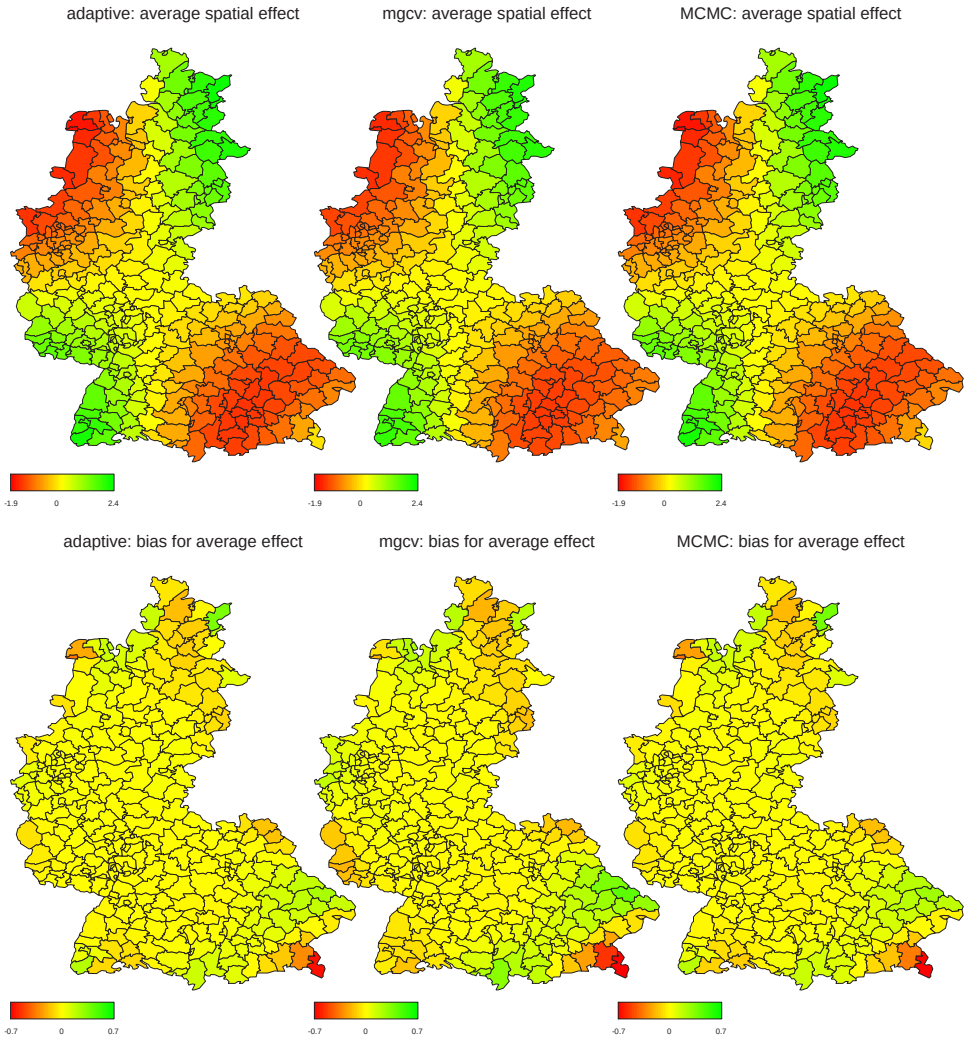

Figure 7.34: Average estimates and empirical bias of the spatial main effect f_{spat} for adaptive search (left column), mgcv package (middle) and MCMC techniques (right column). In the bias plots, yellow indicates regions without bias. For some approaches there is one region with a bias lower than -0.7 (mgcv: -1.19 and MCMC: -0.73).

algorithms yielded comparable results to the adaptive search. All mistakes are due to unimportant variables that were additionally included into the model.

- The computing times displayed in table 7.10 show that the adaptive search was once more the fastest algorithm. Mgcv was considerably slower than any of the other approaches.

In addition to the selection of a single best model we performed further evaluations to investigate the performance of conditional and unconditional credible intervals (compare chapter 5). For this purpose, we used the original data set with $n = 927$ observations and

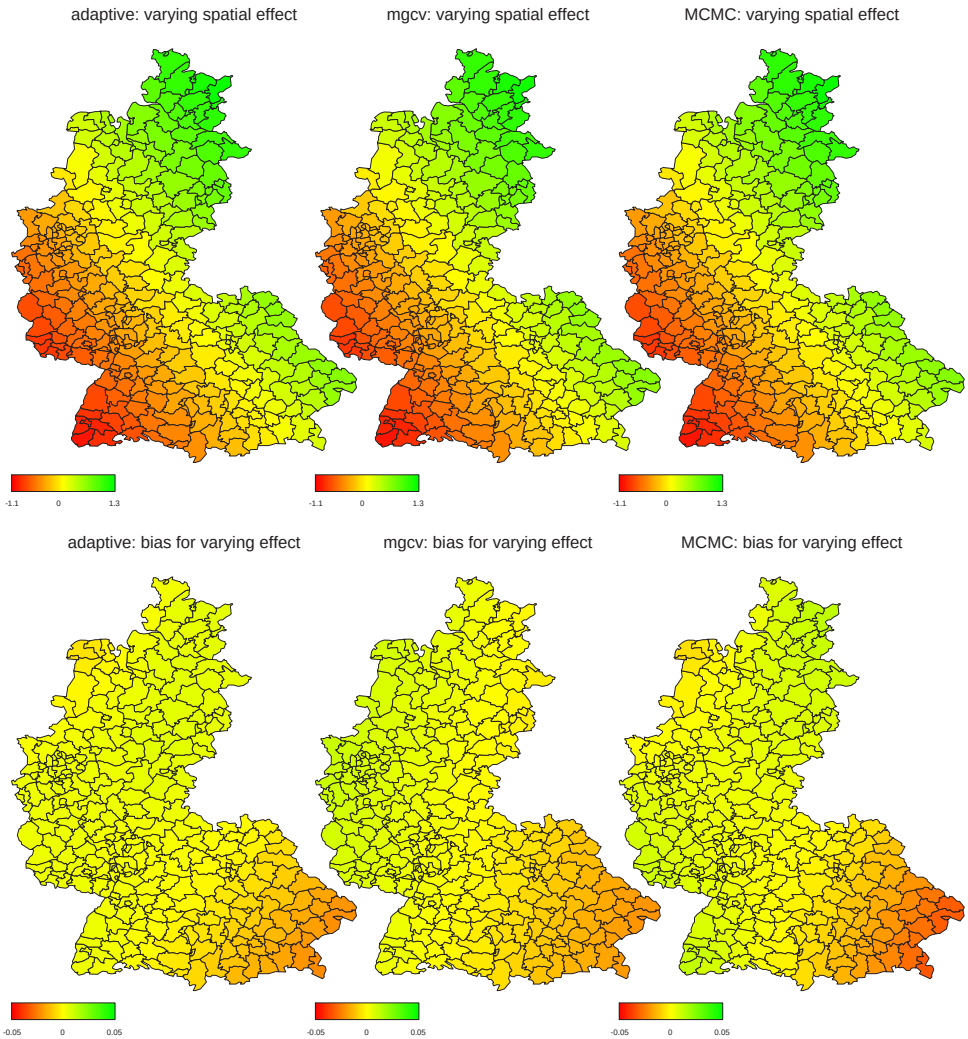

Figure 7.35: Average estimates and empirical bias of the spatial varying effect g_{spat} for adaptive search (left column), mgcv package (middle) and MCMC techniques (right column). Yellow indicates regions without bias.

a larger data set with $2n = 1854$ observations but based on the same predictor. For each replication, unconditional confidence bands were obtained conditional on the respective selected model. We drew 20000 MCMC samples with a thinning parameter of 20, so that the confidence bands are based on 1000 samples. For the unconditional confidence bands we used the same number of MCMC samples that were equally divided between the original data set and 99 bootstrap data sets. For comparison, we show confidence bands of a fully Bayesian approach conditional on the true model (i.e. the unimportant functions are not included in the model but the confidence bands are unconditional with regard to the important functions since their variance parameters can change during the

estimation process) and we show confidence bands obtained by a combination of mgcv and bootstrap (compare Wood (2006c)) with 9 bootstrap data sets (since this approach is very time consuming we could not use more than 9). For all approaches, we present average pointwise coverage probabilities for the individual functions in table 7.11. Here, the results can be summarised as follows:

- For the important nonlinear functions of continuous covariates, the conditional credible bands frequently show undercoverage. The same applies to the mgcv bands. This suggests that 9 bootstrap samples are not enough to consider the full model selection uncertainty. In contrast, the MCMC bands often are considerably above the nominal level. Here, the best results were achieved by the unconditional bands which mostly yielded coverage rates near the nominal level.
- For the unimportant functions, the coverage rates of the unconditional bands are considerably above the nominal level. This could be due to the fact that here only the mistake of overfitting can be made whereas underfitting is impossible. This phenomenon can also be observed with the mgcv bands.
- The credible bands for the spatial functions mostly show considerable overcoverage. Here, only mgcv yielded coverage rates that were close at the nominal level.
- For all approaches, average coverage rates are closer to the nominal level if the sample size is increased.
- Figure 7.36 compares conditional bands, unconditional bands and MCMC bands for some individual functions. In order to highlight the differences between the approaches, we plotted the differences between the bands and the respective true underlying function. The MCMC bands are clearly wider than the other bands. Between unconditional and conditional bands there is a small difference where the unconditional bands are slightly wider than the conditional ones. An example for distinctly different conditional and unconditional bands is given in figure 8.18 for a real data set.

data		conditional	uncond.	MCMC	mgcv	conditional	uncond.	MCMC	mgcv
		f_1				g_1			
n	95%	0.898	0.933	0.969	0.906	0.952	0.963	0.973	0.889
2n	95%	0.932	0.959	0.970	0.933	0.939	0.958	0.972	0.941
n	80%	0.738	0.769	0.834	0.735	0.799	0.828	0.856	0.757
2n	80%	0.781	0.816	0.843	0.769	0.781	0.814	0.844	0.789
		f_3				g_3			
n	95%	0.921	0.939	0.966	0.861	0.923	0.939	0.961	0.903
2n	95%	0.940	0.953	0.970	0.935	0.948	0.962	0.970	0.950
n	80%	0.748	0.767	0.837	0.670	0.763	0.781	0.819	0.741
2n	80%	0.780	0.798	0.838	0.763	0.794	0.819	0.844	0.794
		f_{spat}				g_{spat}			
n	95%	0.988	0.984	0.990	0.917	0.984	0.985	0.987	0.960
2n	95%	0.994	0.991	0.995	0.926	0.980	0.983	0.987	0.941
n	80%	0.945	0.926	0.951	0.766	0.912	0.917	0.927	0.829
2n	80%	0.966	0.952	0.969	0.779	0.904	0.912	0.925	0.804
		f_2				g_2			
n	95%	0.951	0.962	0.971	0.952	0.947	0.983	—	0.992
2n	95%	0.943	0.958	0.969	0.955	0.956	0.991	—	0.995
n	80%	0.807	0.830	0.857	0.826	0.812	0.924	—	0.924
2n	80%	0.794	0.827	0.848	0.818	0.852	0.930	—	0.946
		f_4				g_4			
n	95%	0.940	0.949	0.964	0.950	0.938	0.979	—	0.992
2n	95%	0.944	0.951	0.958	0.950	0.945	0.983	—	0.989
n	80%	0.782	0.793	0.822	0.796	0.851	0.917	—	0.928
2n	80%	0.789	0.802	0.817	0.799	0.846	0.920	—	0.936
		f_5				g_5			
n	95%	0.930	0.967	—	0.982	0.938	0.978	—	0.992
2n	95%	0.950	0.981	—	0.992	0.947	0.982	—	0.980
n	80%	0.803	0.885	—	0.906	0.857	0.920	—	0.922
2n	80%	0.864	0.935	—	0.948	0.867	0.930	—	0.911
		f_6				g_6			
n	95%	0.951	0.982	—	0.993	0.948	0.983	—	0.986
2n	95%	0.942	0.983	—	0.992	0.955	0.986	—	0.982
n	80%	0.877	0.938	—	0.936	0.867	0.935	—	0.926
2n	80%	0.844	0.918	—	0.945	0.868	0.933	—	0.899

Table 7.11: Average coverage probabilities for the individual functions based on nominal levels of 95% and 80%. Values that are more than 2.5% below (above) the nominal level are indicated in red (green).

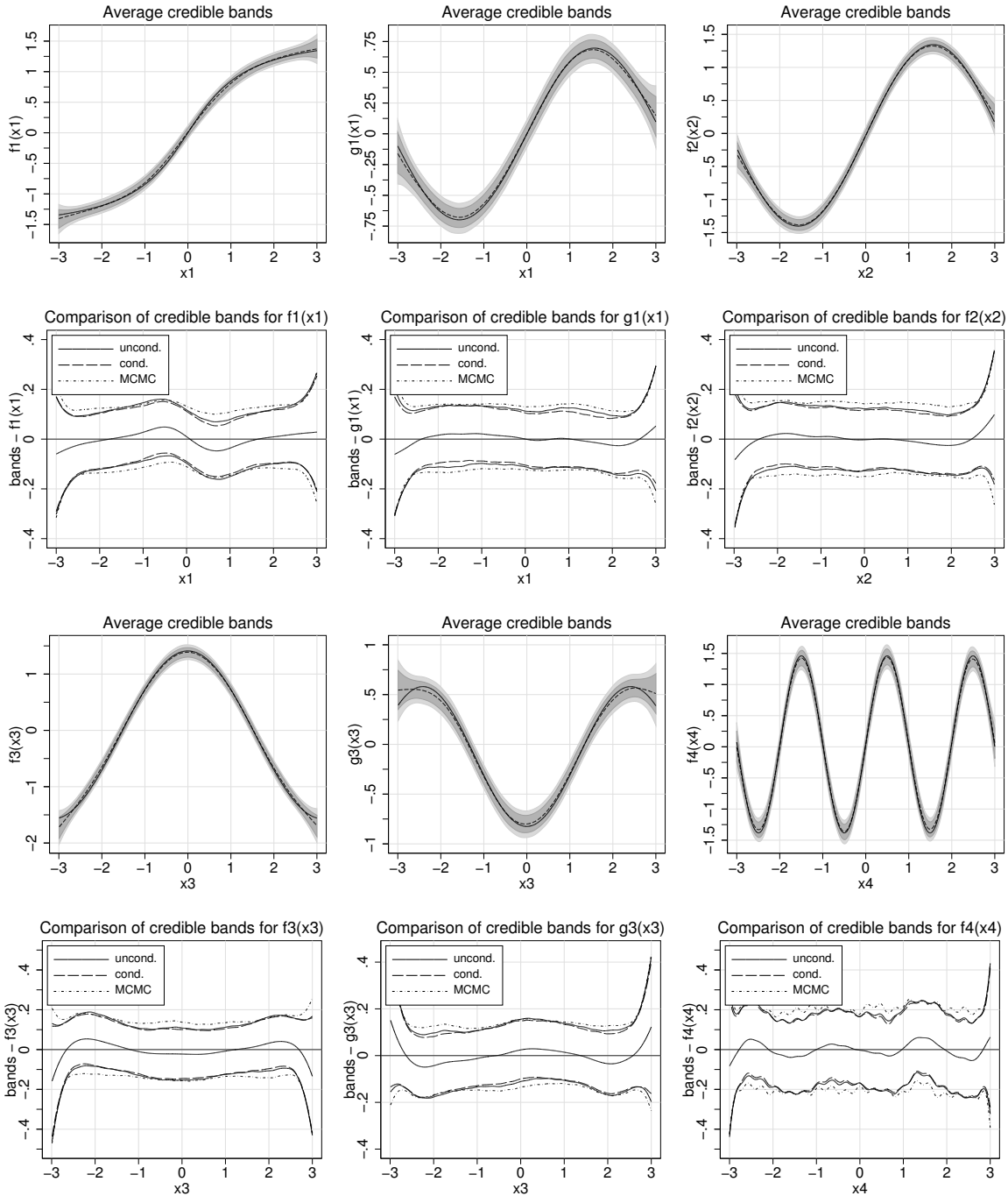

Figure 7.36: Lines 1 and 3 show average unconditional credible bands for the adaptive search together with the true underlying function (solid line) and average estimated function (dashed line). Lines 2 and 4 show differences between 95% credible bands and true function and thus compare conditional bands, unconditional bands and bands obtained by MCMC techniques.

7.5 Simulation of ANOVA type interaction models

In this section we examine the performance of the ANOVA type decomposition of a two–dimensional surface into two main effects and interaction component as described in section 2.2.8.2. For this purpose we show the results of two simulation studies. The predictor in the first simulation study includes an interaction whereas the predictor of the second study consists of two main effects only.

7.5.1 Model including an interaction

The aim of this first simulation study is to examine the performance of the ANOVA type decomposition regarding the following aspects:

- the overall performance of the estimated model, i.e. the estimated predictor is compared to the true predictor,
- the quality of the individual functions (both main effects and interaction), i.e. the individual estimated functions are compared to the respective true function.

For this purpose we use a predictor containing two nonlinear main effects of continuous covariates and a complex interaction, i.e.

$$\eta = \gamma_0 + f_1(x_1) + f_2(x_2) + f_{1|2}(x_1, x_2)$$

with functions

$$\begin{aligned} f_1(x_1) &= 12 \cdot (x_1 - 0.5)^2 - 1.13, \\ f_2(x_2) &= 1.5 \cdot \sin(3 \cdot \pi \cdot x_2) - 0.28, \\ f_{1|2}(x_1, x_2) &= 3 \cdot \sin(2 \cdot \pi \cdot x_1) \cdot (2x_2 - 1). \end{aligned}$$

The functions are chosen such that the sum of main effects has about the same range of values as the interaction component (the range of values is about $[-3; 3]$ in both cases). The interaction component is carefully chosen such that it is not possible to extract a main effect, i.e. neither a function of x_1 nor of x_2, from it. That will later enable us to compare the estimated functions to the true underlying functions. The true functions and the predictor are shown in figure 7.37.
The covariate values of x_1 and x_2 for the $n = 300$ observations lie in the interval $[0; 1]$. 121 observations lie on a 11×11 grid of equidistant points between 0 and 1, so that each point of this grid appears at least once in the data set. All other values for x_1 and x_2 were chosen independently of each other and uniformly on the range $[0; 1]$ but rounded to two

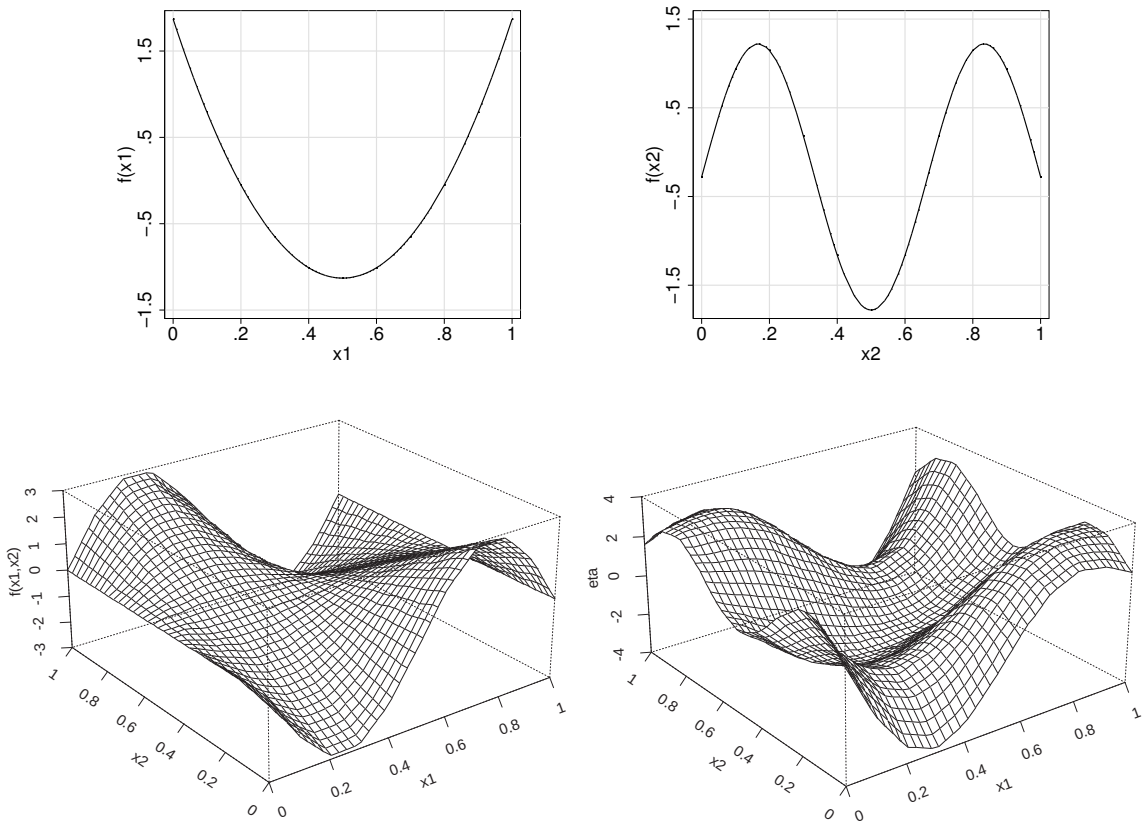

Figure 7.37: Model including interaction: True components and predictor η for the simulation of the ANOVA type interaction model.

decimal places afterwards.
From the predictor we created $R = 250$ replications with Gaussian distributed response variables. The variance of the error terms was chosen as $\sigma^2 = 1.16$ leading to a ratio of $\sigma_\eta^2/\sigma_\varepsilon^2 = 3$.
In order to be able to assess the quality of the estimates we compare the results of the following approaches:

- ANOVA type decomposition as described in section 2.2.8.2 (*anova*):
  For the overall surface we used a two–dimensional cubic P–spline with $12^2 = 144$ basis functions. Hence, the extracted main effects are P–splines with 12 basis functions. For the penalisation in the direction of the main effects we used second order random walk penalties. The estimation was carried out by the adaptive search algorithm. For each component it was possible to be removed from the model, to be approximated by a linear effect or to be modelled by a nonlinear function with the restriction that the interaction component cannot be more complex than any of the main effects

(regarding the used function type, i.e. zero function, linear fit or nonlinear function). For the nonlinear functions of the two main effects (i.e. for λ_1 and λ_2 by setting $\lambda = \infty$) 10 different degrees of freedom were given by $\{2, \ldots, 11\}$. For parameter λ of the nonlinear interaction component (i.e. setting $\lambda_1 = \lambda_2 = 0$) the values were determined according to 27 degrees of freedom specified by $\{25, \ldots, 90\}$.

- Model containing a surface estimator only (*surface*):
  Here we used a two–dimensional P–spline with 12^2 B–spline basis functions of third degree and a second order random walk penalty (compare section 2.2.8.1). The estimation was also carried out using the adaptive search algorithm. For the smoothing parameter we specified 35 possibilities with resulting degrees of freedom equidistant between 5 and 90. Besides, there were the possibilities of a linear effect and the removal from the model.

- Model containing two main effects and interaction component (*mcmc*):
  In contrast to the first approach, the two main effects are not extracted from an overall surface but specified and estimated as separate components. As penalties we used one- or two–dimensional second order random walk penalties. The estimation was carried out using a fully Bayesian approach based on MCMC simulation techniques. In contrast to the *anova* approach, no selection is performed. That means, the model specification using the three spline functions is fixed but smoothing parameters are estimated.

With this simple predictor the exact search yielded the same results as the adaptive search. Therefore, the results of the exact search are not shown.
For the comparison of results we computed average estimates, empirical bias and empirical mean squared errors and draw the following conclusions:

- Regarding the MSE values of predictor and individual components shown in figure 7.38, the median of the distributions for *mcmc* and *anova* is nearly identical. Mostly, the distribution for *anova* has a slightly larger variance than the one for *mcmc*. Regarding the predictor, both approaches *mcmc* and *anova* perform considerably better than the approach with a surface estimator only.

- Apart from f_1 where the estimators of both approaches *mcmc* and *anova* are practically identical with the true function, the bias of the individual components is slightly larger for *anova* than for MCMC (compare figures 7.39 and 7.40). This is also true for the predcitor shown in figure 7.41. The bias of the predictor for *surface* is considerably larger than those of the other approaches.

- Although *anova* had the possibility of model selection (i.e. to remove the interaction

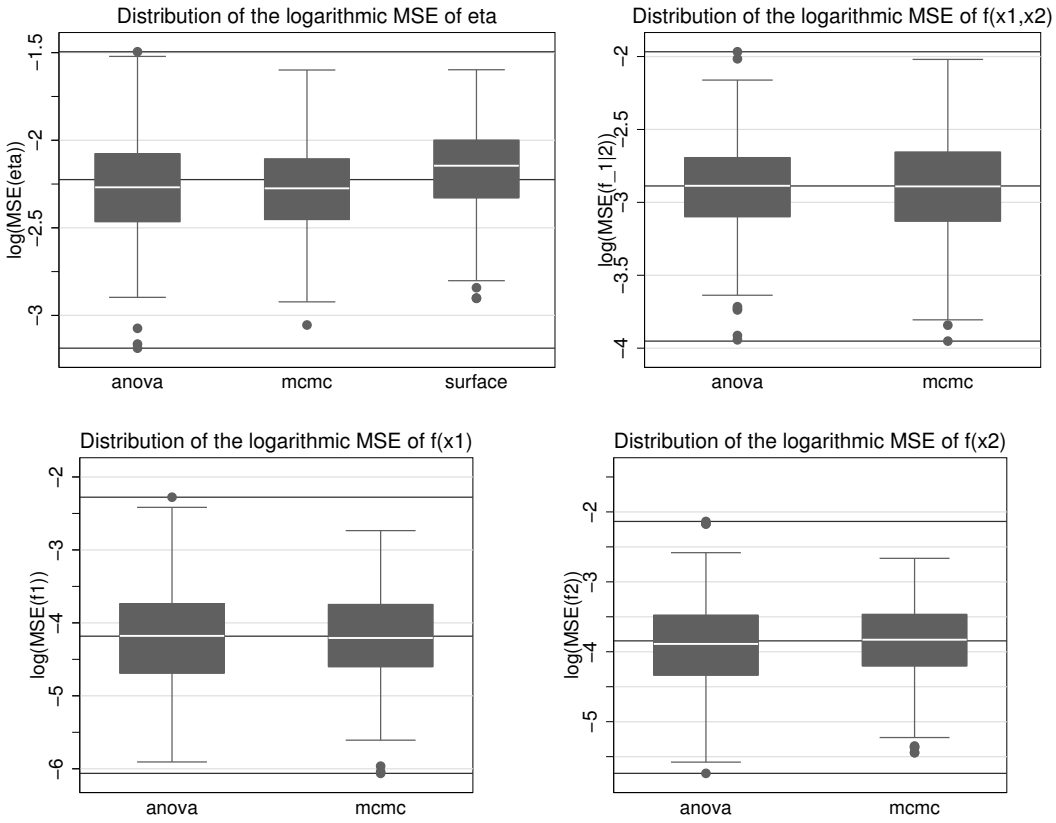

Figure 7.38: Model including interaction: Distributions of the empirical logarithmic MSE for predictor and individual functions.

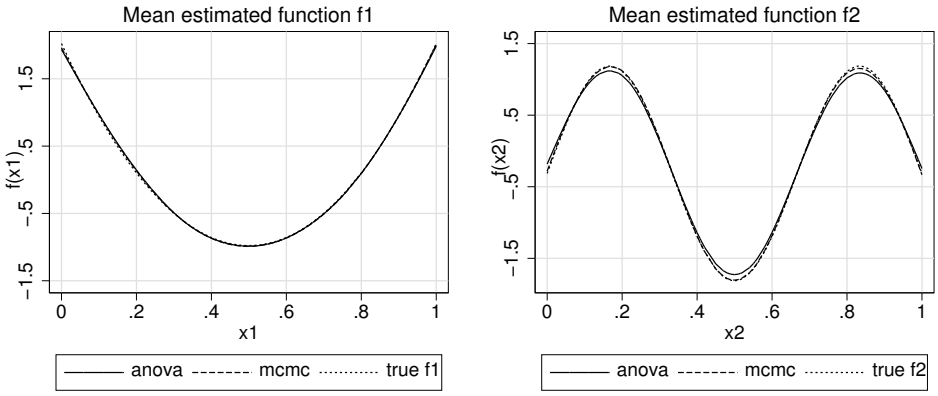

Figure 7.39: Model including interaction: Average estimated main effects together with the true underlying functions.

term from the model and estimate a main effects model), the full interaction was always selected (not shown).

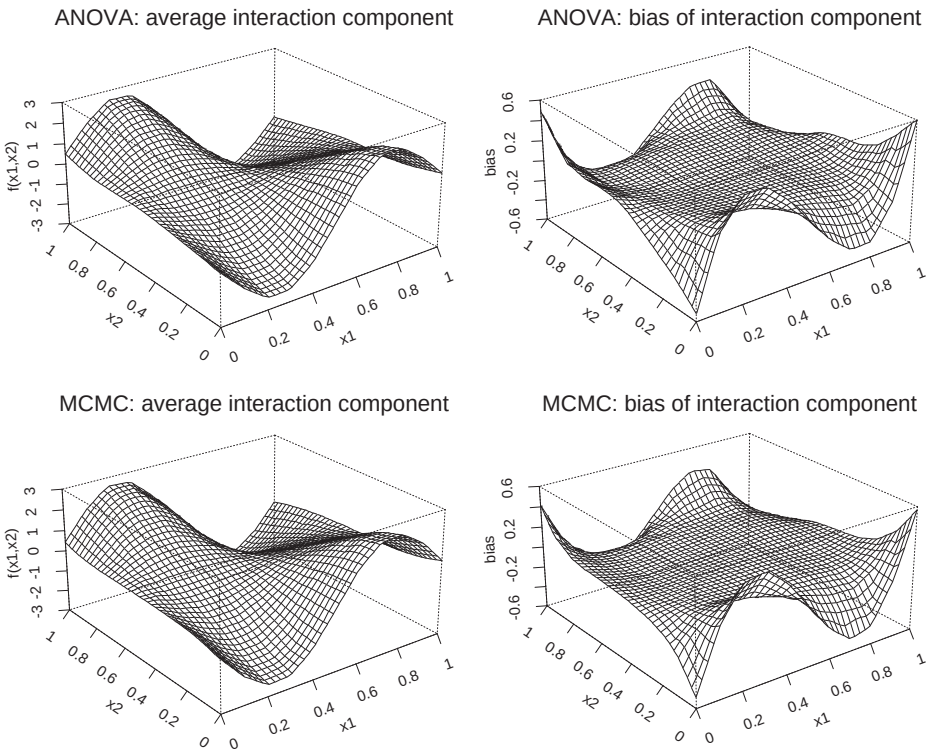

Figure 7.40: Model including interaction: Average estimated interaction components (left column) and their bias (right column). The upper plots show the results of the ANOVA type decomposition and the lower plots those of the MCMC approach

- Summarising these results, the estimates of the ANOVA type decomposition are nearly as good as those obtained by the real interaction model *mcmc*.

7.5.2 Model without interaction

Based on the same covariates x_1 and x_2 and the same functions f_1 and f_2 as above, we created a predictor containing no interaction component, i.e.

$$\eta = \gamma_0 + f(x_1) + f(x_2).$$

With this simulation study we examine if the search algorithms are able to detect that the interaction term has no influence on the response. Additionally, we analyse if the selected model depends on the chosen starting model and if so, which starting model produces the best results.
From the predictor we again created $R = 250$ replications with Gaussian distributed response variables. The variance of the error terms was chosen by $\sigma^2 = 0.63$ again leading

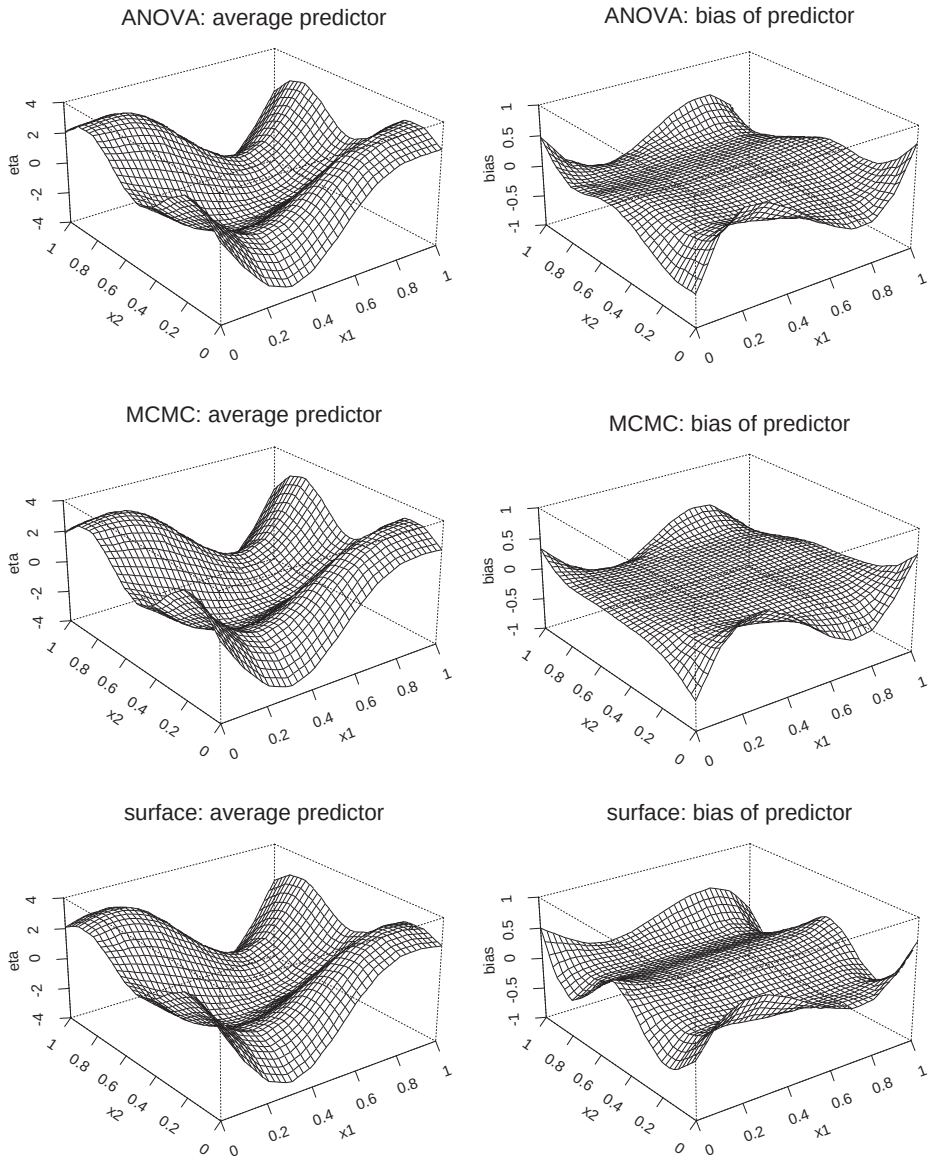

Figure 7.41: Model including interaction: Average estimated predictors (left column) and their bias (right column). The upper plots show the results of the ANOVA type decomposition, the plots in the middle the results of the MCMC approach and the lower plots those of the surface estimator.

to a ratio of $\sigma_\eta^2/\sigma_\varepsilon^2 = 3$.
In order to be able to assess the performance of the search algorithms we compare the results of the following approaches:

- ANOVA type interaction model starting with the linear basis model (*linear*):

For the overall surface we used a two–dimensional cubic P–spline with 12^2 basis functions. Hence the extracted main effects are cubic P–splines with 12 basis functions. For the penalisation in the direction of the main effects we used second order random walk penalties. The search started from the linear predictor $\eta = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_{1|2} x_1 \cdot x_2$.

- ANOVA type interaction model starting with a linear main effects model (*removed*):
  In contrast to the *linear* approach, the starting predictor contains only the two linear main effects, i.e. $\eta = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2$.
- ANOVA type interaction model starting with a nonlinear model (*nonlinear*):
  Here the starting predictor contains a relatively smooth nonlinear overall surface, i.e. $\eta = \gamma_0 + f_1(x_1) + f_2(x_2) + f_{inter}(x_1, x_2)$.
- Main effects model (*main*):
  This approach serves as a reference because the interaction component is not considered at all. That means, the predictor cannot contain an interaction term and the search algorithm only has to estimate the two main effects. Again, the starting model is the linear model.

As search algorithms we used the adaptive and the exact search and compared the results. We draw the following conclusions:

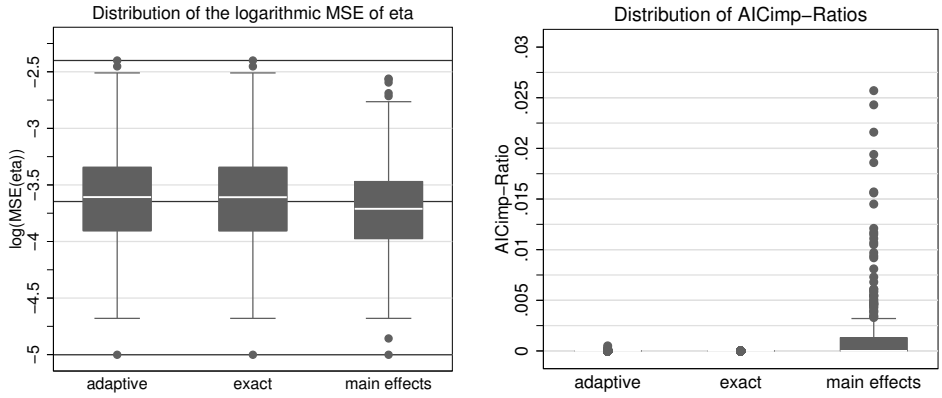

Figure 7.42: Model without interaction: Distributions for the empirical logarithmic MSE of the predictor and for the ratio of AIC_{imp} values.

- For exact and adaptive search the results are independent of the basis model as the same models were selected with each of the basis models (results are not shown).
- The results regarding empirical MSE and ratio of AIC_{imp} values show that there is

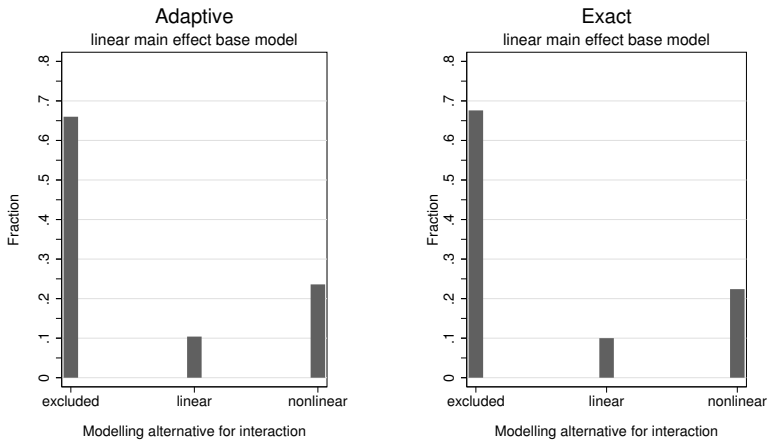

Figure 7.43: Model without interaction: Distributions of modelling alternatives for the interaction component.

practically no difference between exact and adaptive search (compare figure 7.42). The same is indicated by the results regarding the selected representation of the interaction component shown in figure 7.43.

- In terms of empirical MSE the results of the main effects model are slightly better than the results of the selected models. The ratios of AIC_{imp} values, however, show that the selected model always has equal or lower AIC_{imp} values than the main effects model. Hence, the selection algorithms chose wrong models because of their better AIC_{imp} values.
- Figure 7.43 shows that in about 65% of replications the interaction component was correctly removed from the model. But in more than 20% of replications, the interaction component was modelled nonlinearly, meaning that an ANOVA type decomposition was used for the overall surface with small degrees of freedom for the interaction component.
- Summarising the results it turns out that the selection algorithms are able to detect that a complex modelling of two continuous variables including an interaction component is not necessary. But this strongly depends on the evidence given by the selection criterion. If the selection criterion decides in favour of an interaction component the selection algorithms do not remove it from the model. However, the values of ratio (7.1) are only small for the main effects model with zero median. This indicates, that there are only small differences between the selected model and the main effects model.

7.6 Conclusion

During the last sections we examined the performance of our selection algorithms by means of several simulation studies which imitated different data situations. In conclusion, the results of all simulation studies show the following pattern:

- There was no notable difference between the results of the approaches based on the coordinate descent method, i.e. adaptive search, exact search and adaptive/exact search. The values of the selection criteria obtained by the adaptive search were only slightly worse than those obtained by each of the other approaches. This indicates that very similar models were selected.
- The stepwise algorithm often yielded worse results than the algorithms based on the coordinate descent method. Above all, its selected model strongly depends on the chosen basis model whereas the adaptive search's selected model proved to be independent of the basis model.
- The results (in terms of quality of estimates and correctly selected terms) obtained by the selection algorithms based on the coordinate descent method are fully comparable to those obtained by mgcv. However, with the discrete response distributions mgcv failed due to convergence problems whereas our algorithms worked well.
- For the coordinate descent methods, estimated functions and predictors are only slightly worse than the estimates obtained by MCMC techniques conditional on the true model.
- The adaptive search was by far the fastest approach for model selection. Even for complex models the selection was performed in a very short time. For some simulations, the adaptive search needed one hour to select all replications whereas mgcv needed more than a week.

Summarising these results, our adaptive search algorithm is a strongly efficient and easy to apply approach in the context of model selection in STAR models.

Chapter 8

Applications

8.1 Belgian car insurance data

In order to calculate appropriate premiums for a car insurance, there are two different factors to be considered: on the one hand the frequency of claims per policyholder and on the other hand the costs that have arisen by these claims. The data in this application is from two Belgian insurance companies from 1997. Altogether, the data contains information of about 160000 policyholders of whom about 18000 had at least one claim during this year. In the next sections we analyse both claim frequency and claim size using different kinds of models. Available covariates with a possible influence both on the costs and on the frequencies are:

ageph	Age of the policyholder
agec	Age of the car
bm	Bonus–malus score
hp	Horse power of the car (in kilowatts)
dist	District in Belgium in which the car is licensed
fuel	Fuel oils (1 = gasoline, -1 = diesel)
fleet	The vehicle belongs to a fleet (= 1) or not (= -1)
s	Gender of the policyholder (1 = male, -1 = female)
use	Use of the vehicle (1 = professional, -1 = private)
cov	Coverage: additional subscriptions to ordinary TPL (1 = none, 2 = limited material damage or theft, 3 = comprehensive coverage)

The three–categorical variable *cov* is represented in the model (if used) by two effect variables with the first category as reference. For the analysis we excluded cases where the

car's age lay above 20.
The data was already analysed by Denuit & Lang (2004) using geoadditive models and MCMC inference techniques. They had to perform model choice and variable selection in a time–consuming procedure by comparing a small number of competing models via the Deviance information criterion (compare Spiegelhalter, Best, Carlin & Van der Linde (2002)). Hence, they could not compare such a large number of models as our automatic selection algorithms. Nevertheless, we can use their results to judge the plausibility of our results.

8.1.1 Claim size

In this section, we want to analyse the costs of claims (for insured events) and find the important regressors which influence them. For this purpose, we use the data from the $n = 18139$ policyholders who had at least one claim. Here, the response variable *logs* is the logarithmic average cost per claim per policyholder leading to a log–normal model. The logarithmic costs are used because the costs of a claim can take only positive values and are right–skewed. The number of claims per policyholder (*nclaims*) are used as weight variable. A descriptive analysis shown in figure 8.1 suggests different effects of the policyholder's

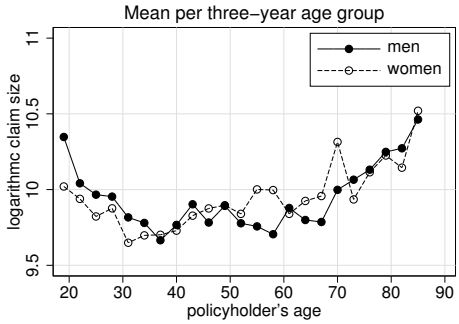

Figure 8.1: Average logarithmic claim sizes each calculated over three successive years of age separately for men and women.

age for men and women. In order to investigate if there actually is a difference, we use in addition to the main effect $f_1(ageph)$ a varying coefficient with s as interacting variable. The effects of other covariates could also show differences between men and women and, hence, the largest possible predictor takes the form

$$\begin{aligned}\eta \;=\; & \gamma_0 + \gamma_1\, fleet + \gamma_2\, use + \gamma_3\, fuel + \gamma_4\, fuel \cdot s + f_c(cov) + g_c(cov, s) + \\ & f_1(ageph) + g_1(ageph) \cdot s + f_2(agec) + g_2(agec) \cdot s + f_3(hp) + g_3(hp) \cdot s + \\ & f_4(bm) + g_4(bm) \cdot s + f_{spat}(dist) + g_{spat}(dist) \cdot s + \gamma_s\, s. \qquad (8.1)\end{aligned}$$

This predictor provides the possibility of estimating separate effects for men and women for all covariates apart from *fleet* and *use*. By removing the respective interaction term from the predictor it is also possible to estimate a non–varying effect. For the categorical variables *fleet* and *use* the reference category is observed for most observations so that there is not enough information for the estimation of two separate effects. Variable s is effect–coded so that female marginal effects are obtained as $f_j^{(fem)} = f_j - g_j - \gamma_s$ whereas male marginal effects are $f_j^{(male)} = f_j + g_j + \gamma_s$. The categorical variables are all effect–coded with the exception of the interaction between *cov* and s. Here, we use dummy–coding leading to the function

$$g_c(cov, s) = \begin{cases} 0 & \text{, if } s = -1 \text{ or } cov = 1 \\ \gamma_{cs1} & \text{, if } s = 1 \text{ and } cov = 2 \\ \gamma_{cs2} & \text{, if } s = 1 \text{ and } cov = 3. \end{cases}$$

Multicategorical variables are either completely removed from the predictor or represented by the complete set of dummy or effect variables. The effects of the continuous covariates (f_j and g_j, $j = 1, \ldots, 4$) can each be represented either by P–splines with different degrees of freedom, by a straight line or they can be removed from the model. For the two spatial functions (f_{spat} and g_{spat}) there are only the possibilities of using a Markov random field with different degrees of freedom or removing the function from the model. All different possibilities for the individual model terms are listed in table 8.1. Model selection is performed using the adaptive search in combination with the improved AIC.
The selected predictor is

$$\begin{aligned} \eta^{(cost)} &= \gamma_0 + \gamma_1\, fleet + f_c(cov) + f_1(ageph) + g_1(ageph) \cdot s + f_2(agec) + \\ &\quad g_3(hp) \cdot s + f_4(bm) + f_{spat}(dist) + \gamma_s\, s \end{aligned} \tag{8.2}$$

where only the effects of the policyholder's age and of horsepower show a difference between men and women. The details of the final model, i.e. the chosen degrees of freedom are listed in table 8.1. The interpretation of this selected model is given below.
The progression of the selection on the basis of AIC_{imp} values and modelling alternatives of each term is shown in table 8.2. The greatest improvement was yielded during the first iteration. From the third iteration onward, there is only one minor change in the model. The last row shows the AIC_{imp} value for the final model after convergence of the backfitting algorithm. The trend of AIC_{imp} is additionally shown in figure 8.2. The selection process took only about two minutes to get the final model.
In addition to the selection of a single best model, we perform a further analysis in order to obtain unconditional confidence intervals and frequency distributions of the modelling alternatives for each term. This analysis is performed using the hybrid algorithm of MCMC techniques and bootstrap sampling described in chapter 5. Here, we use 99 bootstrap

term	no	possible term types	range for df	chosen possibility
$fuel$	1	linear effect	$\{0,1\}$	df = 0
$fuel \cdot s$	2	linear effect	$\{0,1\}$	df = 0
use	3	linear effect	$\{0,1\}$	df = 0
$fleet$	4	linear effect	$\{0,1\}$	df = 1
s	5	linear effect	$\{0,1\}$	df = 1
$f_c(cov)$	6	linear effects	$\{0,2\}$	df = 2
$g_c(cov,s)$	7	linear effects	$\{0,2\}$	df = 0
$f_1(ageph)$	8	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 6
$g_1(ageph)$	12	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 2
$f_2(agec)$	9	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 7
$g_2(agec)$	13	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 0
$f_3(hp)$	10	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 0
$g_3(hp)$	14	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 1
$f_4(bm)$	11	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 5
$g_4(bm)$	15	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 0
$f_{spat}(dist)$	16	Markov random field	$\{0,5,\ldots,200\}$	df = 35
$g_{spat}(dist)$	17	Markov random field	$\{0,5,\ldots,200\}$	df = 0

Table 8.1: Summary of possible term types and degrees of freedom. The last column shows the degrees of freedom chosen for the final model. Column no *yields numbers for figure 8.2.*

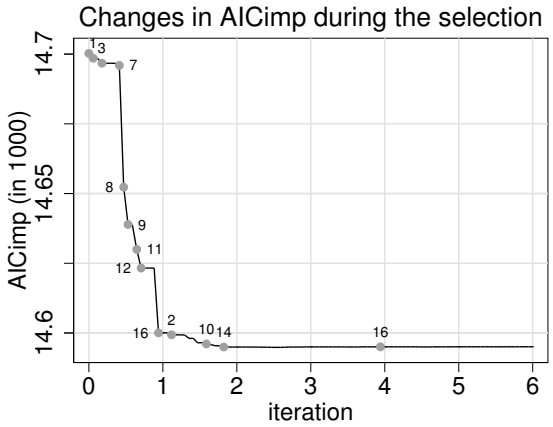

Figure 8.2: Changes in AIC_{imp} during the selection. The grey dots and numbers mark variables whose modelling is changed. The variables / terms belonging to the numbers are given in table 8.1.

samples so that, together with the original data, we have 100 (possibly) different selected models. For each of these selected models we draw 1000 random samples using the Gibbs sampler. We use every tenth MCMC sample for the calculation of confidence bands so that, altogether, each confidence band is based on 10000 samples.
For the final model, the effects of continuous covariates (black lines) together with 95%

and 80% confidence bands are shown in figure 8.3 and the spatial effect together with 95% and 80% significance maps in figure 8.4. The sampling distributions of degrees of freedom obtained from bootstrapping can be found in figures 8.6 and 8.5. They can be used to perform a sensitivity analysis regarding the selected model.
The selected predictor (8.2) shows that most of the interactions with sex are not selected. For the horsepower of the car a linear interaction effect is in the AIC_{imp} best model but not the main effect. Hence we observe a sex specific linear effect of horsepower with opposite sign as shown in figure 8.3. However, the effect is uncertain as we will see below. Among the other potential interactions only the effect of *ageph* varies with s. The selected model (8.2) is similar to the model used in Denuit & Lang (2004). However, the interactions with s are not included in their model because a systematic investigation of interaction effects was not possible at that time.
The old drivers report more expensive claims than younger ones. Moreover, there is a clear interaction with the gender of the policyholder. The claim sizes of female policyholders are mostly higher than for males at the same age. The sampling distribution of the degrees of freedom of the main effect shows a mode around 5–6, whereas for the interaction effect a mode at $df = 1$ (linear effect) is obtained. The effect of the bonus malus score has an inverse U–form, i.e. the average claim sizes increase until a score of about 16 and decrease thereafter. The decrease for policyholders with very high bonus malus score is probably caused by more cautious driving due to the negative experience in the past. Note however that only a few observations with $bm > 16$ are available and as a consequence large confidence intervals are obtained. Moreover, the sampling distribution of the degrees of freedom is bimodal with a local maxima at $df = 1$ suggesting that a linear effect might be reasonable as well. Overall we conclude that the effect for $bm > 16$ is relatively uncertain. Even more uncertain is the effect of horsepower showing increasing average claim sizes for female drivers and decreasing claim sizes for male drivers. The effect is small compared to other covariates and the confidence intervals are comparably large including the zero everywhere. The sampling distribution of the degrees of freedom shows almost equal probabilities of about 40% for zero or one degrees of freedom suggesting the exclusion of the effect as a reasonable alternative. Altogether, the selected effect of *hp* is likely to be an artefact. The spatial effect shows that highly urban areas (Brussels and Antwerp) are less dangerous as far as severities are concerned, whereas highly rural zones, like the extreme South of Belgium are much more dangerous in that respect. The spatial effect shows clearly no differences between the sexes and the significance maps of the varying effect (not shown) are zero everywhere. For the categorical covariates, the decision if the variables are important or not is very stable. For the effects of *cov* there was even always the same alternative selected: The average effect of coverage is absolutely important whereas there is clearly no interaction regarding gender.

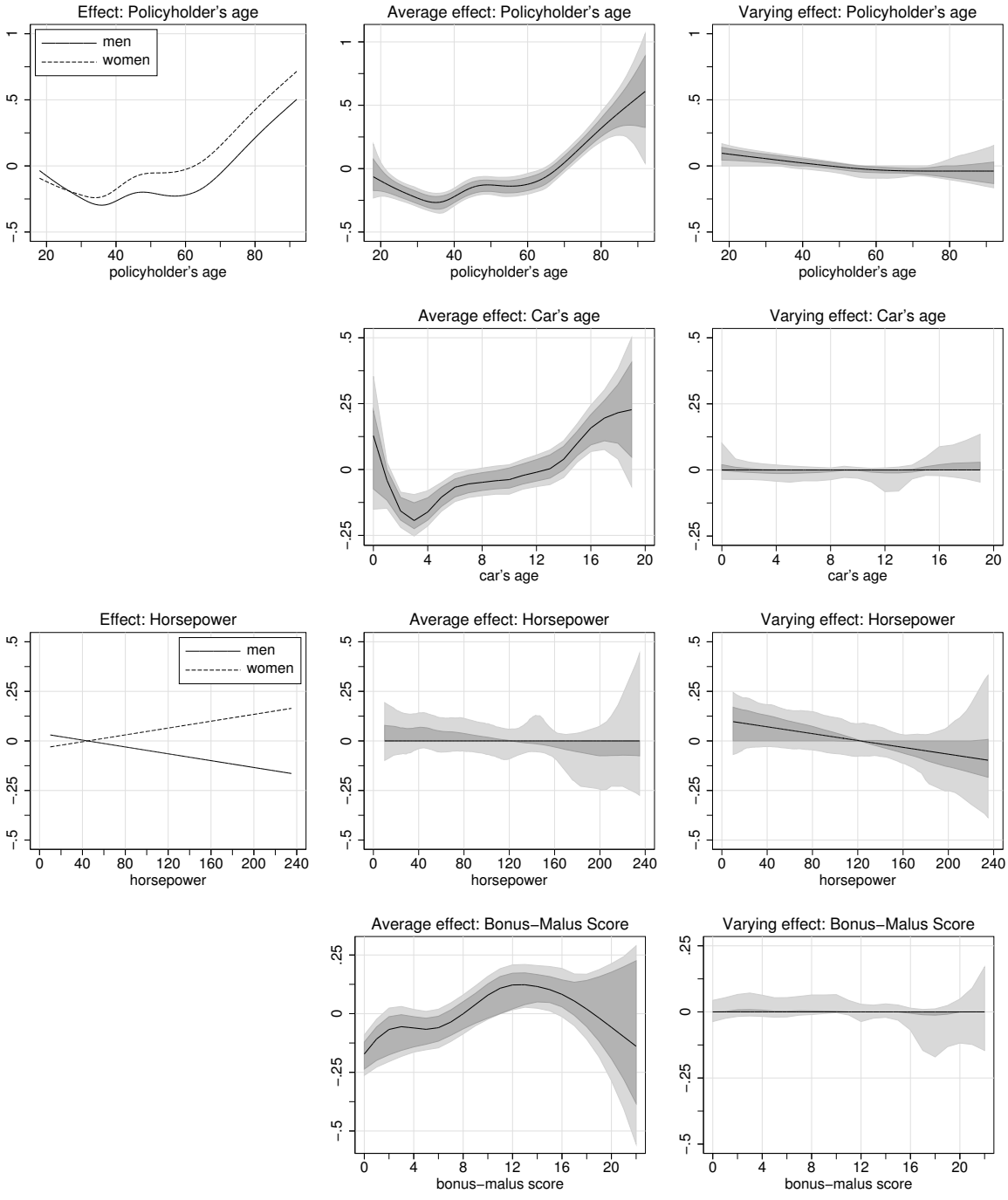

Figure 8.3: Effects including confidence bands of the continuous covariates.

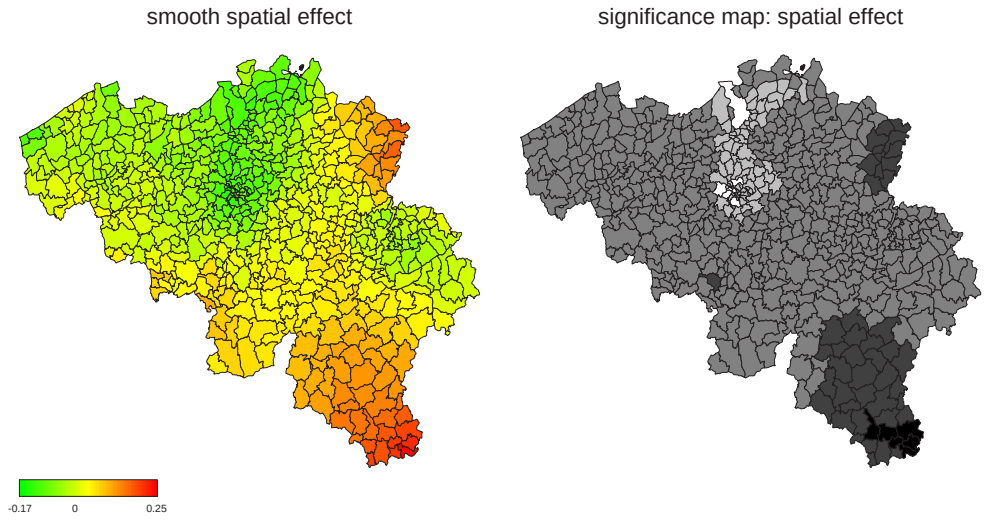

Figure 8.4: Average spatial effect and corresponding significance map. The significance map indicates significant positive (white or light grey) and significant negative regions (black or dark grey) at both 80% and 95% levels (white/black) or at 80% level (otherwise). The significance map for the varying spatial effect shows no variation and is therefore omitted.

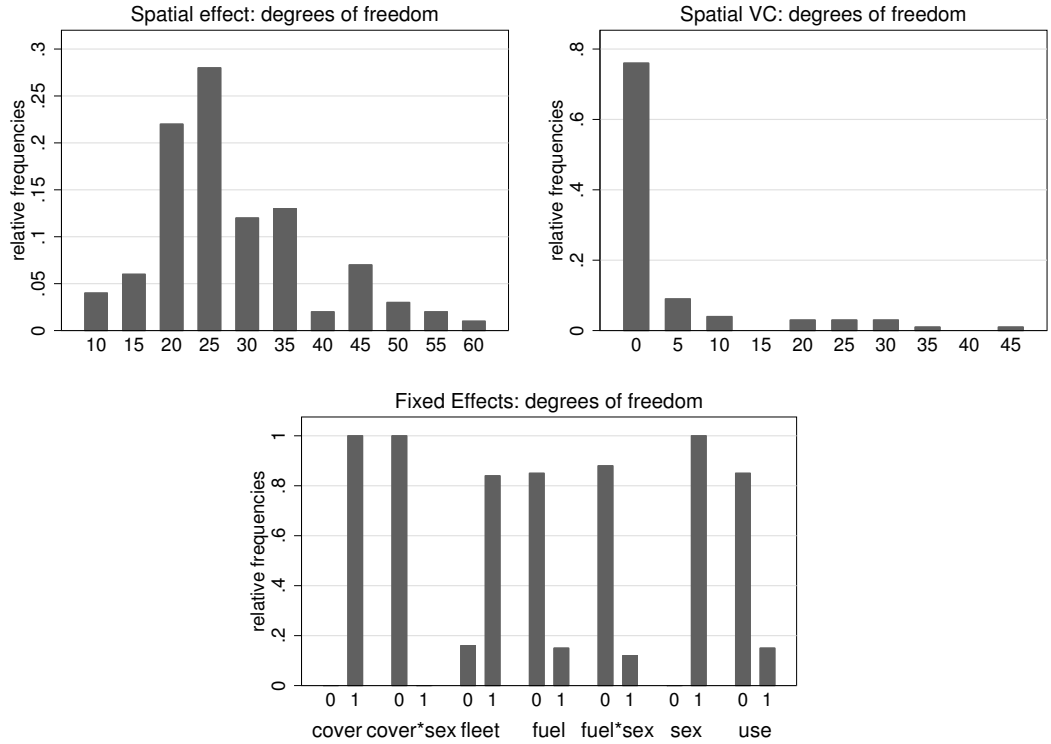

Figure 8.5: Sampling distributions of the different modelling alternatives obtained by bootstrap replications.

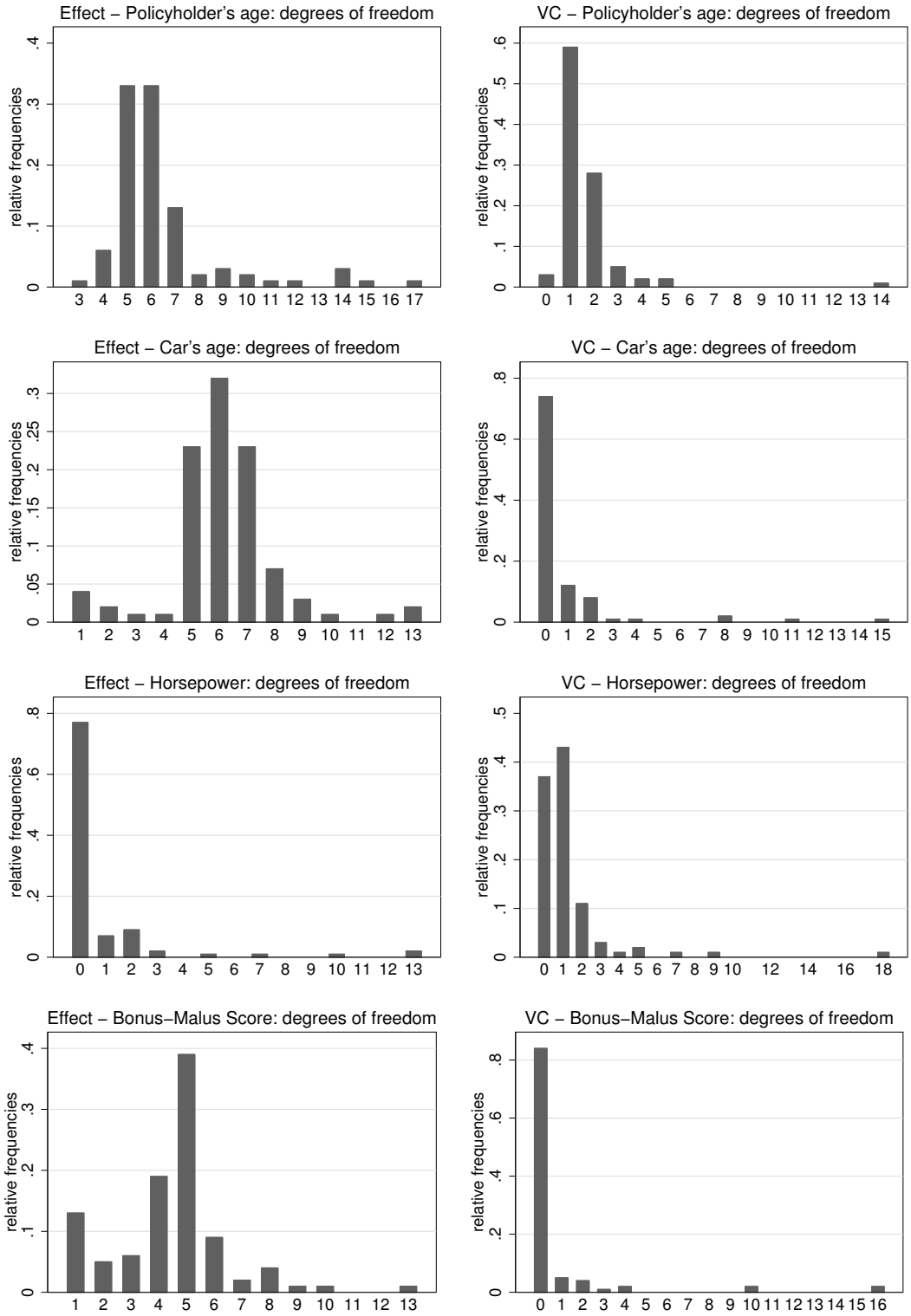

Figure 8.6: Sampling distributions of the different modelling alternatives obtained by bootstrap replications.

It.	AIC_{imp}	s	*fleet*	*use*	*fuel*	$(fuel,s)$	*cov*	(cov,s)	*ageph*	$(ageph,s)$	*agec*	$(agec,s)$	*hp*	(hp,s)	*bm*	(bm,s)	*dist*	$(dist,s)$
0	14700.239	1	1	1	1	1	2	2	1	0	1	0	1	0	1	0	5	0
1	14600.057	1	1	0	0	1	2	0	6	2	7	0	1	0	5	0	40	0
2	14594.954	1	1	0	0	0	2	0	6	2	7	0	0	1	5	0	40	0
3	14595.013	1	1	0	0	0	2	0	6	2	7	0	0	1	5	0	40	0
4	14595.042	1	1	0	0	0	2	0	6	2	7	0	0	1	5	0	35	0
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
B	14595.071	1	1	0	0	0	2	0	6	2	7	0	0	1	5	0	35	0

Table 8.2: Claim size: Degrees of freedom of model terms during the selection process. Changes in the degrees of freedom from one iteration to another are underlined. Iteration 0 corresponds to the start model. B shows the value of AIC_{imp} after the last backfitting step.

It.	AIC	s	*fleet*	*use*	*fuel*	$(fuel,s)$	*cov*	(cov,s)	*ageph*	$(ageph,s)$	*agec*	$(agec,s)$	*hp*	(hp,s)	*bm*	(bm,s)	*dist*	$(dist,s)$
0	120748.4	1	1	1	1	1	2	2	1	0	1	0	1	0	1	0	5	0
1	120164.65	1	1	1	1	0	2	2	6	4	10	1	7	0	14	3	110	5
2	120123.55	1	1	1	1	0	2	2	6	5	10	1	6	0	14	2	120	10
3	120112.41	1	1	1	1	0	2	2	6	5	10	1	7	0	14	1	125	10
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
B	120104.89	1	1	1	1	0	2	2	6	5	10	1	7	0	14	1	125	10

Table 8.3: Claim frequency: Degrees of freedom of model terms during the selection process. Changes in the degrees of freedom from one iteration to another are underlined. Iteration 0 corresponds to the start model. B shows the value of AIC after the last backfitting step.

8.1.2 Claim frequency

Here, the claim frequency *nclaims*, i.e. the number of claims per policyholder, is analysed. For that purpose, we use a loglinear Poisson model with a structured additive predictor. Again, the largest possible predictor is predictor (8.1) where the covariates' effects can differ between men and women. For possible term types and possible degrees of freedom compare table 8.1. Some policyholders were insured for only a part of the year so that the number of days during which the policy was valid (*duration*) has also to be considered. This leads to the definition of a risk variable by

$$risk = 0.5 \cdot \ln(duration/365).$$

Variable *risk* is added to the predictor as an offset parameter, i.e. no regression parameter is specified for *risk* and it is not included in the selection process.
The selection procedure uses AIC which can be readily used as there are considerably more observations than maximum possible degrees of freedom. The selected predictor is

$$\begin{aligned}\eta^{(freq)} &= risk + \gamma_0 + \gamma_1\, fleet + \gamma_2\, use + \gamma_3\, fuel + f_c(cov) + g_c(cov) + f_1(ageph) + \\ &\quad g_1(ageph) \cdot s + f_2(agec) + g_2(agec) \cdot s + f_3(hp) + f_4(bm) + g_4(bm) \cdot s + \\ &\quad f_{spat}(dist) + g_{spat}(dist) \cdot s + \gamma_s\, s\end{aligned}$$

The interpretation of the covariates' effects is given below.
The details of the final model, i.e. the chosen degrees of freedom are listed in table 8.4. The progression of the selection on the basis of AIC values and modelling alternatives of each term is shown in table 8.3. The greatest improvement was yielded during the first iteration. From the second iteration onward, there are only minor changes in the model, i.e. the degrees of freedom of some nonlinear functions change slightly. The last row shows the AIC value for the final model after convergence of the local scoring procedure. The trend of AIC is additionally shown in figure 8.7. The selection process took only about 15 minutes to get the final model.
In addition to the selection of a single best model, we again use the hybrid algorithm of MCMC techniques and bootstrap sampling described in chapter 5 in order to obtain unconditional confidence intervals and frequency distributions of the modelling alternatives for each term. Again, we use 99 bootstrap samples so that, together with the original data, we have 100 (possibly) different selected models. For each of these selected models we draw 300 random samples using a Metropolis–Hastings algorithm with IWLS proposal. We use every 30–th MCMC sample for the calculation of confidence bands so that, altogether, each confidence band is based on 1000 samples.
For the final model, the effects of continuous covariates (black lines) together with 95%

term	chosen possibility
s	df = 1
$fleet$	df = 1
use	df = 1
$fuel$	df = 1
$fuel \cdot s$	df = 0
$f_c(cov)$	df = 2
$g_c(cov, s)$	df = 2
$f_1(ageph)$	df = 6
$g_1(ageph)$	df = 5
$f_2(agec)$	df = 10
$g_2(agec)$	df = 1
$f_3(hp)$	df = 7
$g_3(hp)$	df = 0
$f_4(bm)$	df = 14
$g_4(bm)$	df = 1
$f_{spat}(dist)$	df = 125
$g_{spat}(dist)$	df = 10

Table 8.4: Degrees of freedom chosen for the model of claim frequencies. For possible term types and possible degrees of freedom compare table 8.1.

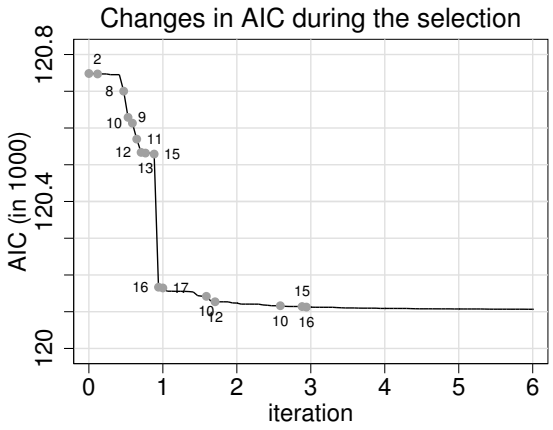

Figure 8.7: Changes in AIC during the selection. The grey dots and numbers mark variables whose modelling is changed. The variables / terms belonging to the numbers are given in table 8.1.

and 80% confidence bands are shown in figure 8.8 and the average spatial effect together with 95% and 80% significance maps in figure 8.9. The sampling distributions of degrees of freedom obtained from bootstrapping can be found in figures 8.10 and 8.12. They can be used to perform a sensitivity analysis regarding the selected model.
Again, the selected model is similar to the model used by Denuit & Lang (2004). In

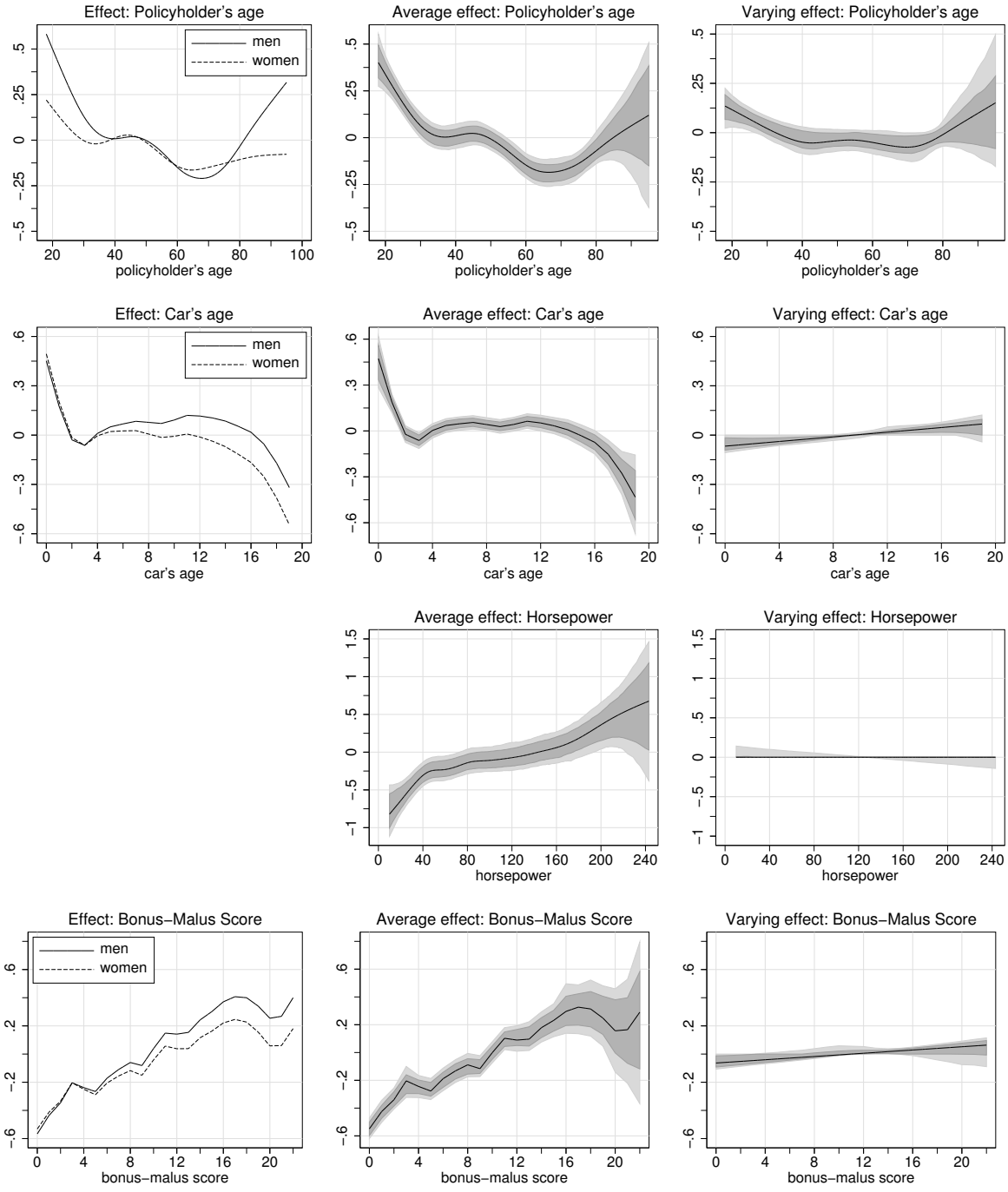

Figure 8.8: Effects including confidence bands of the continuous covariates.

contrast to the model for claim sizes, there are more effects with an interaction regarding the gender of the policyholder. The policyholder's age shows clearly different effects for men and women that were also discovered by Denuit & Lang (2004). Generally, young and

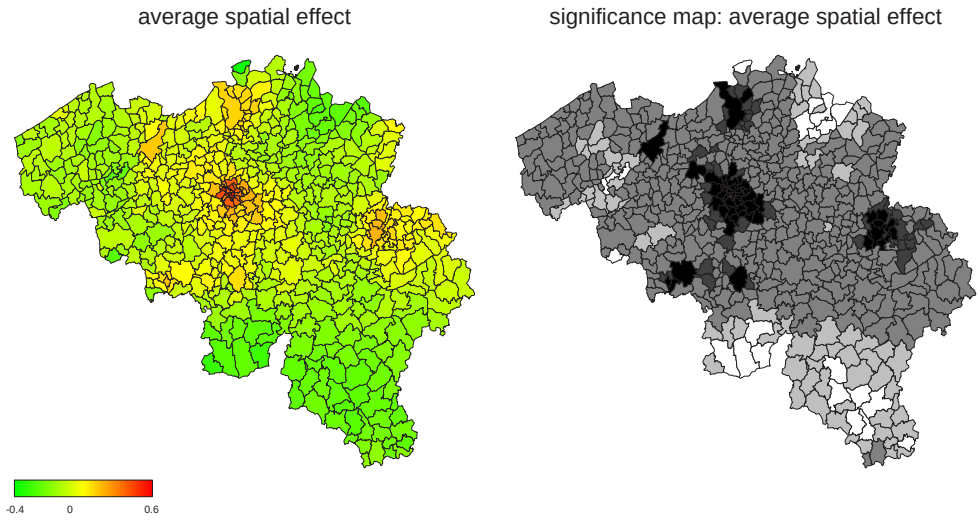

Figure 8.9: Average spatial effect and corresponding significance map. The significance map indicates significant positive (white or light grey) and significant negative regions (black or dark grey) at both 80% and 95% levels (white/black) or at 80% level (otherwise). The significance map for the varying spatial effect shows no variation and is therefore omitted.

old drivers produce more claims what is more clearly pronounced with men. Young and old women report less accidents than men of the same age whereas there is no difference between women and men for the age of 40 to 70. Note however, that both average and varying effect have broad confidence intervals for an age above 80 due to few observations in that range. The peak at an age of about 45 in the effects of both sexes could be caused by children driving their parent's car. This peak is especially pronounced in the female effect what can be attributed to the fact that young car owners often ask their mother to purchase the policy (compare Denuit & Lang (2004)). The varying effect for the policyholder's age is quite strong with the mode of the sampling distribution at $df = 3$.

New cars produce more accidents than old cars. The effect reaches a local minimum at the age of three. This can be attributed to the Belgian characteristic that up to three year old cars don't have to undergo the annual mechanical check–in. The male and female effects are nearly identical up to the age of three but differ afterwards: Women report less accidents than men. The number of accidents decreases for very old cars. Here, the varying effect is also identified as important with a mode at $df = 1$ corresponding to a linear varying effect.

The number of reported accidents increases with horsepower. Here, there is clearly no difference between the sexes. The effect of the bonus–malus score has also a positive trend but with differences between men and women: the effect is identical for values up to six, whereas for higher values women report less claims than men. The varying effect has its

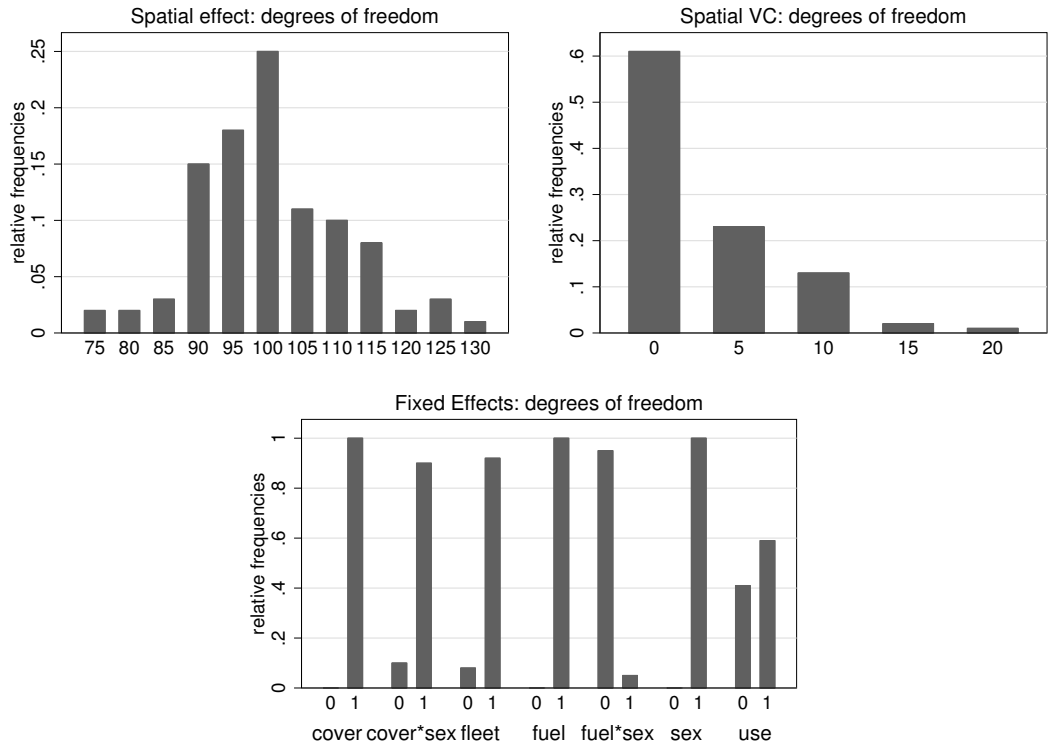

Figure 8.10: Sampling distributions of the different modelling alternatives obtained by bootstrap replications.

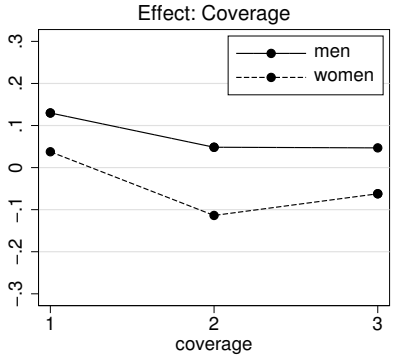

Figure 8.11: Effect of coverage.

mode at $df = 1$ corresponding to a linear varying effect and is identified as important. The average effect is very rough with a mode at $df = 11$ and a selected value of $df = 14$. However, when estimating a model without the offset parameter *risk*, the selected value for the effect of *bm* is $df = 6$ leading to a smooth, increasing function. (The modelling of all other terms is not influenced by removing the offset parameter.)
The spatial effect is also selected as varying over s, but the varying effect with a selected

value of $df = 10$ is only small. Moreover, the mode of the sampling distribution is at $df = 0$ with a frequency of 60%. This indicates that the varying spatial effect is very uncertain and should rather be excluded from the model. The same is indicated by the significance maps (80% and 95%) that are zero everywhere (not shown). The average spatial effect shows that in urban areas more claims are reported and less claims in highly rural areas, especially the extreme south of Belgium. Hence, for claim frequencies the opposite effect can be observed compared to the claim size.
The effects of the categorical covariates are quite stable since the frequency distribution clearly support the selected alternatives. The only exception is *use* that is selected with a frequency of only 60% indicating that the alternative of removing this variable from the model should be considered as well. The effect of coverage is here varying with s. As $f_c(cov)$ uses effect coding and $g_c(cov)$ dummy coding the marginal effects are obtained as

$$f_c^{(fem)}(cov) = \begin{cases} -\gamma_{c1} - \gamma_{c2} - \gamma_s & \text{, if } cov = 1 \\ \gamma_{c1} - \gamma_s & \text{, if } cov = 2 \\ \gamma_{c2} - \gamma_s & \text{, if } cov = 3 \end{cases}$$

$$f_c^{(male)}(cov) = \begin{cases} -\gamma_{c1} - \gamma_{c2} + \gamma_s & \text{, if } cov = 1 \\ \gamma_{c1} + \gamma_{cs1} + \gamma_s & \text{, if } cov = 2 \\ \gamma_{c2} + \gamma_{cs2} + \gamma_s & \text{, if } cov = 3 \end{cases}$$

For both sexes, the number of claims is largest for the simple alternative $cov = 1$. Women with comprehensive coverage ($cov = 3$) report more claims than with $cov = 2$ whereas the male effect shows no difference between these alternatives (compare figure 8.11).

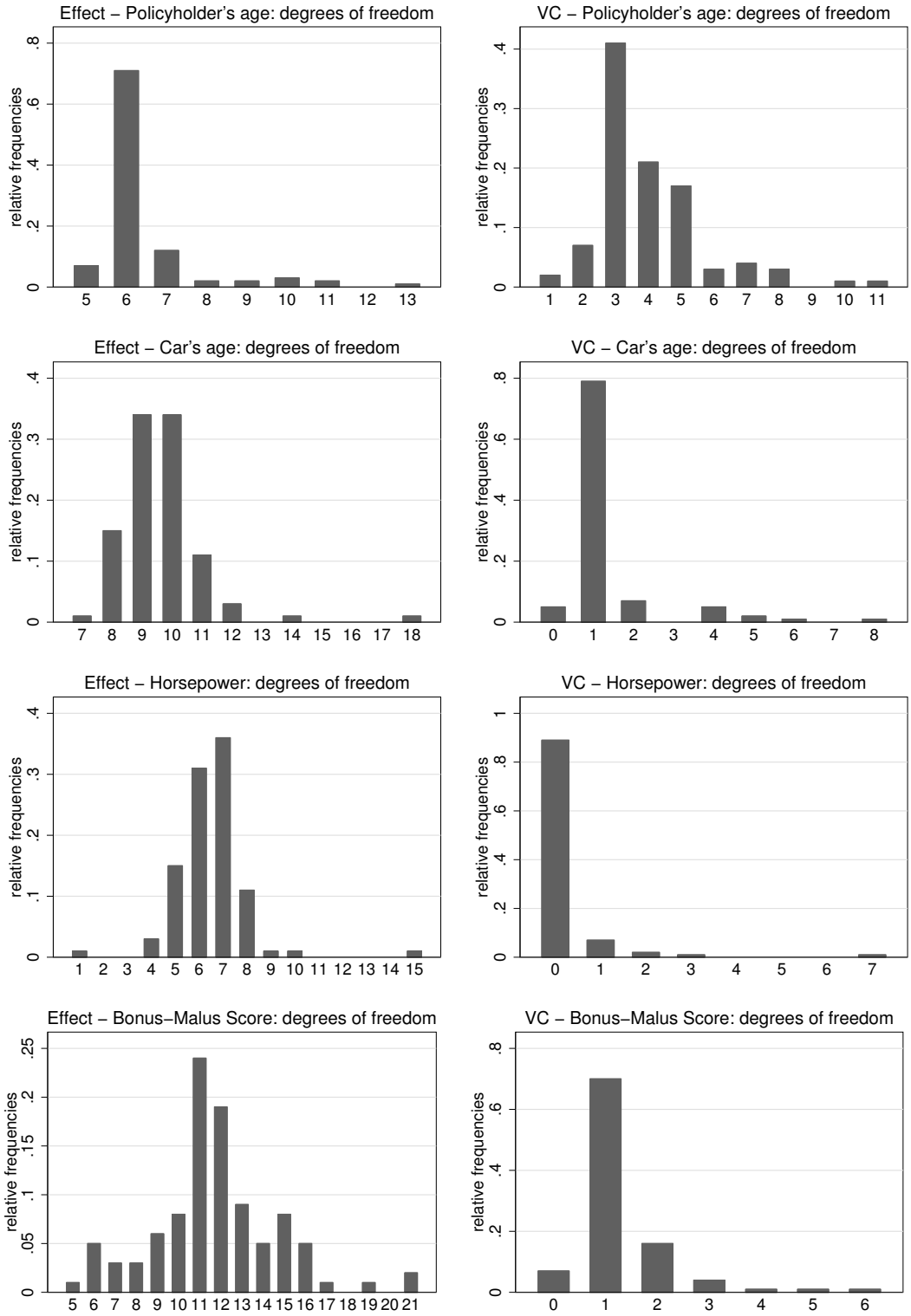

Figure 8.12: Sampling distributions of the different modelling alternatives obtained by bootstrap replications.

8.2 Malnutrition of children in India

Very high prevalence of childhood undernutrition as well as very large gender bias are two of the most severe development problems in India. In this section, we will consider these two problems. Our analysis is based on micro data from the second National Family Health Survey (NFHS-2) from India which was conducted in the years 1998 and 1999. Among others, the survey collected detailed health, nutrition and anthropometric information on children born in the three years preceding the survey. The data includes approximately 13000 observations of male and 12000 observations of female children.

Undernutrition among children is usually measured by determining the anthropometric status of the child relative to a "reference population" of children known to have grown well. Researchers distinguish between three types of undernutrition: wasting or insufficient weight for height indicating acute undernutrition; stunting or insufficient height for age indicating chronic undernutrition; and underweight or insufficient weight for age which could be a result of either. In this section we focus on stunting. For a child i stunting is typically determined using a Z-score which is defined as

$$Z_i = \frac{AI_i - MAI}{\sigma}, \tag{8.3}$$

where AI refers to the height of the child, MAI and σ refer to the median height and the standard deviation of children in the reference population at the same age. The analysis in this section is strongly oriented at the analysis performed by Belitz, Hübner, Klasen & Lang (2007).

Undernutrition in India shows a clear regional pattern which is different for boys and girls. This is visible in the maps (a) and (b) of figure 8.19 which show smooth spatial functions without controlling for other covariates. In North–Central India (particularly Uttar Pradesh, Madhya Pradesh, Rajasthan, and Orissa), both sexes suffer from significant undernutrition, while in the very North, the East, and the South West, they are doing significantly better. This spatial pattern seems to be more pronounced for girls than boys. As a result, the significance map of the sex differences in undernutrition (figure 8.20 (b)) shows that girls are significantly worse off than boys in Uttar Pradesh, Madhya Pradesh and West Bengal, while they are significantly better off in the relatively small Northeastern states (e.g. Assam, Nagaland, Tripura).

In the analysis we want to examine if these regional differences can be at least partially explained by other factors. Therefore, we want

- to select and analyse the most important socio–demographic, environmental and health specific determinants of undernutrition,
- to determine the functional form of the effects and

- to investigate possible sex–specific differences of undernutrition.

Moreover, we use the unconditional approach (with 99 bootstrap data sets, 20000 overall MCMC samples and a thinning parameter of 20) for the construction of credible bands and to perform a sensitivity analysis for the selected model. Afterwards, the residuals of the selected model are used for an examination of the remaining spatial differences.
The covariates used in this study are listed in table 8.5. The variables *ecstatH* and *womstatM* need some further explanation: they are linear indices and specified as linear combinations of certain centered and standardised covariates where the weights were calculated by a principal components analysis. The household's economic status *ecstatH* captures the household's economic resource base and includes factors which indicate the household's wealth like e.g. owning a refrigerator, owning a bicycle, having access to piped drinking water, having electricity, owning land, etc. The mother's women's status *womstatM* indicates the mother's power relative to the power of men. Among other disadvantages, women with a low status have weaker control over resources in their household and a more restricted access to health services what is supposed to negatively influence the quality of care they can provide to their children. The index *womstatM* includes variables like e.g. the difference in the years of education between the mother and her partner, their age difference, if the partner's permission is needed for decisions regarding medical care, the frequency of being beaten during the last year, etc. For the exact definition of these two indices compare Belitz, Hübner, Klasen & Lang (2007) or Hübner (2003).
The two variables *ageC* and *bfmC* are strongly interrelated since a child's age automatically constitutes the highest possible value for its duration of breastfeeding. Hence, we need to specify an interaction term for the joint effect of these variables. Here we compare the results of two models that merely differ in the representation of the interaction effect. In the first model (M1) we use a two–dimensional surface, i.e. a two–dimensional P–spline with second order random walk penalty and 17^2 basis functions, both for the interaction effect and the respective varying coefficient term. In contrast, the ANOVA type decomposition (also with 17^2 basis functions) is used for both interaction and varying interaction term in the second model (M2). The ANOVA type decomposition provides the possibility to reduce the interaction term to two main effects.
All available covariates and terms and their modelling alternatives are listed in table 8.6 together with the selected alternatives for both models. Thereby, functions f_j refer to average effects whereas functions g_j refer to varying coefficients with gender as interacting variable. Table 8.6 displays that the selected models are nearly identical with regard to selected variables and terms. However, the AIC_{imp} values of the final models differ with $\text{AIC}_{imp} = 16835.54$ for model (M1) and $\text{AIC}_{imp} = 16812.017$ for model (M2). This difference in the final AIC_{imp} values can only be due to the different interaction terms (for

variable	description
ageC	Child's age in months
bfmC	Months child was breastfed
agebirM	Mother's age at child's birth in years
bmiM	Mother's body mass index
educM	Mother's educational attainment (in years)
heightM	Mother's height in cm
womstatM	Index for mother's women status
ecstatH	Index for household's economic status
sexC	Gender of the child (male = -1; female = 1)
areaH	Place of residence? (urban = -1; rural = 1)
birthinC	Preceding birth interval > 24 months? (no = -1; yes = 1)
born1stC	First born child? (no = -1; yes = 1)
bplaceC	Child was born in hospital? (no = -1; yes = 1)
firstmC	Child got first milk? (no = -1; yes = 1)
hhsizeH	Size of household (small $\hat{=} \leq 5$; medium $\hat{=}$6–10; large $\hat{=} > 10$ members)
ironfolM	Mother got iron folic tablets during pregnancy? (no = -1; yes = 1)
plannedC	Was the child planned? (no = -1; yes = 1)
precareM	Mother received medical care during pregnancy? (no = -1; yes = 1)
religM	Mother's religion (Hinduism, Islam, Christianity, Sikh, others)
tetanusM	Mother got tetanus injection during pregnancy? (no = -1; yes = 1)
toiletH	Household has toilet facility of any kind? (no = -1; yes = 1)
twinC	Child was born under multiple birth? (no = -1; yes = 1)
vacC	Child is vaccinated according to its age? (no = -1; yes = 1)
district	District in India the mother and her child live in

Table 8.5: List of available covariates.

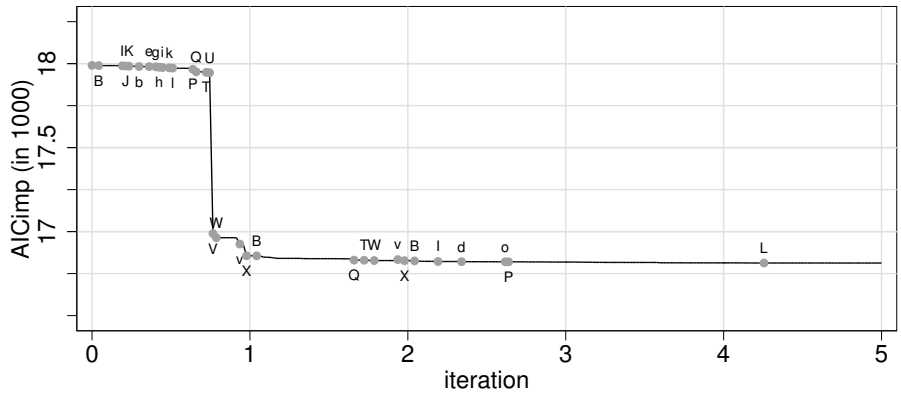

Figure 8.13: Changes in AIC_{imp} during the selection of model (M2). The grey dots and letters mark variables whose modelling is changed. The variables / terms belonging to the letters are given in table 8.6.

the interaction between *ageC* and *bfmC*): model (M1) using a simple surface estimator performed worse than model (M2) using the ANOVA type decomposition.
Figure 8.13 shows the trend of AIC_{imp} during the selection of (M2). The greatest improvement was yielded during the first iteration, particularly for changing the modelling of the age effect. From the second iteration onward, there occurred only minor adjustments.
The comparison of the two different interaction terms for *ageC* and *bfmC* yields interesting results: The overall degrees of freedom of the average interaction effect are for (M2) with $df = 15 + 8 + 24.5 = 47.5$ only slightly smaller than for (M1) with $df = 52.5$ (compare table 8.6). The same applies to the varying interaction term with $df = 7.5$ for (M1) and $df = 5$ for (M2). However, in model (M2), the VC term is only a main effect of child's age whereas variable *bfmC* does not contribute to the sex–varying effect. The respective bootstrap sampling distributions (not shown) confirm this result since the sex–varying effects of *bfmC* or of the interaction component were practically never selected.
Effects of the joint effect of *ageC* and *bfmC* are shown in figure 8.14 for both models (M1) and (M2). Apart from the fact that the effects of model (M1) are slightly smoother than those of (M2), both kinds of effects show the same trend. The nutritional status of all children rapidly deteriorates between birth and an age of 20 months. This indicates that children are not born malnourished but only develop this as a result of disease and inadequate nourishment. The improvement around 24 months is an artefact of the reference standard. At the age of 24 months the reference population changes and children older than 24 months are compared to a worse nourished population than younger children. This artefact is more strongly pronounced in the ANOVA type decomposition effect.
Children who are breastfed for six or twelve months have a better nutritional status, whereas long breastfeeding durations (18 or 24 months) carry no benefits and could indicate a poor availability of alternative nourishments.
Since there are hardly any differences between the two models regarding all other effects, we only show the effects of model (M2). The effects of the categorical covariates are shown in figure 8.15. Many of the effects display the same tendency for boys and girls. In particular, being a twin, having a short preceding birth interval, living in a large household, not being breastfed immediately after birth, and having poor access to prenatal care is all associated with poorer nutrition. According to the sampling distributions, the decision regarding inclusion or exclusion was very certain for most covariates. Figure 8.17 (a) shows only covariates whose selected alternative was chosen in less than 90% of bootstrap samples. Nevertheless, for most of the covariates shown in figure 8.17 (a) the sampling distribution is clearly in favour of the selected alternative. Exceptions are the interactions of *birthinC* and *toiletH* with gender. Here, the relative frequencies are about 50% for inclusion and exclusion. This explains why models (M1) and (M2) differ in these two terms.
Although only a few interaction terms were selected, there turned out to be some notable

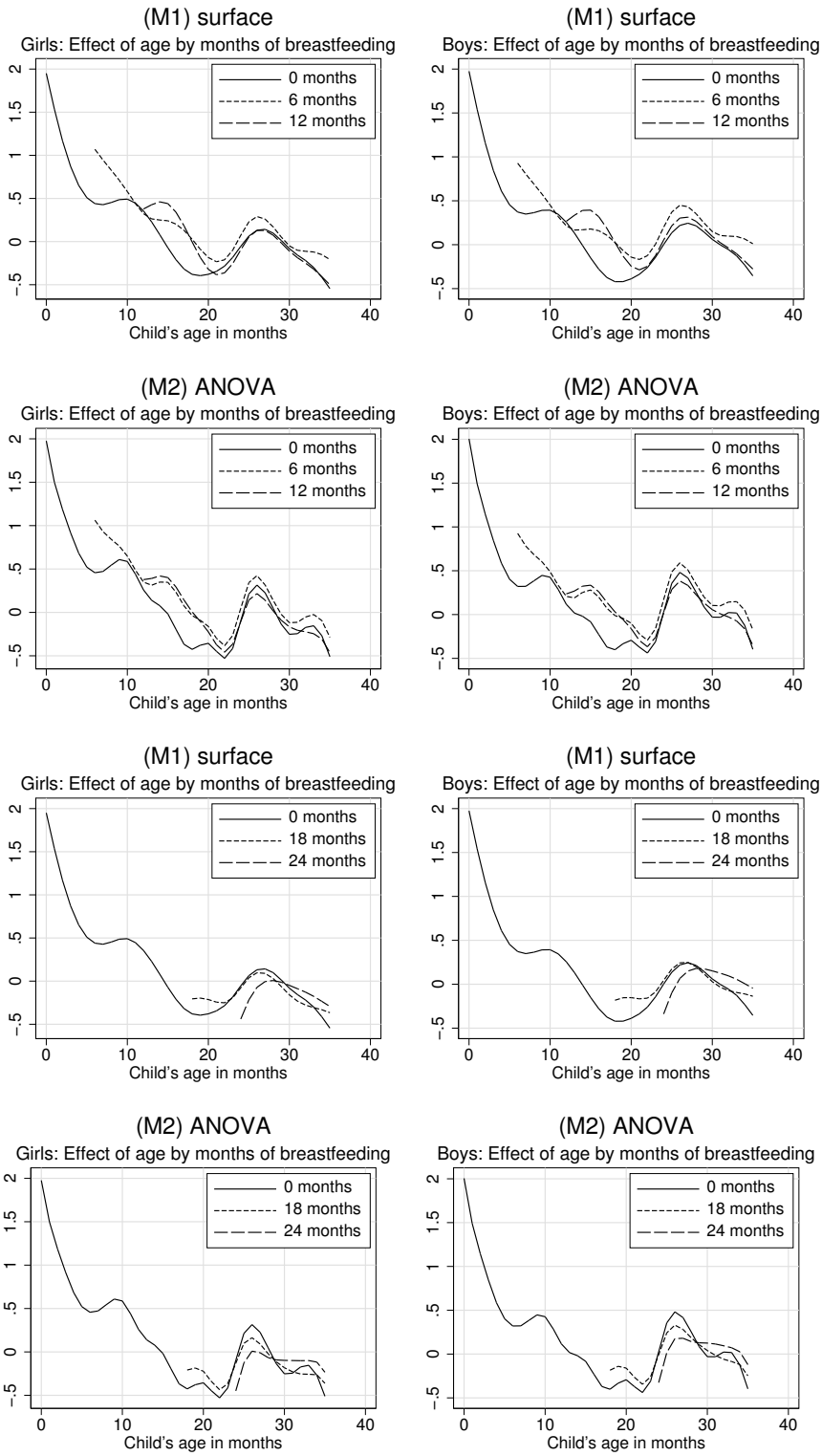

Figure 8.14: Nonlinear effects of the child's age for different durations of breastfeeding.

and systematic differences between boys and girls: It appears that the nutritional status of girls reacts more sensitively to competition for resources within the household. The effect of being a twin or having a short preceding birth interval are more negative for girls than for boys. Especially, the cultural environment matters more for girls with stronger positive effects for Christian and other religions and stronger negative effects for Islam and Sikh.
Among the effects for continuous covariates (apart from *ageC* and *bfmC*) only the effect of mother's women's status differs with sex (compare figure 8.16). The effects of this variable are surprising and should be treated with caution since *womstatM* is highly correlated with other covariates used in the regression. Moreover, the sampling distributions in figure 8.17 (c) and (d) indicate that the sex–varying term is not at all relevant (mode at $df = 0$) and that the relevance of the average effect is at least questionable with two modes at $df = 0$ and $df = 5$. In fact, if one just considers the univariate impact of women's status on the Z-score, the effect is strongly positive for both girls and boys (with a stronger effect for girls) (compare figure 8.18). Thus, women's relative status has a positive impact, but this is mediated via the other effects. The effect shown in figure 8.16 is only positive for high relative women's status for girls, and negative for boys which seems plausible if one can assume that high status mother's exhibit, under the same other conditions, a preference for favouring their daughters.
Additionally, figure 8.18 compares the conditional and the unconditional confidence bands for the varying effect of *womstatM*. The confidence bands show considerable differences where only the unconditional bands indicate clearly the areas of greater uncertainty.
All other effects show no relevant interaction with gender and the respective bootstrap sampling distributions confirm this fact. But there are strong (common) increasing effects of mother's age at birth, her BMI, as well as her educational attainment on the nutrition of her child. A high household's economic status has also a positive effect. The sampling distributions in 8.17 (b) and (e) suggest to use a linear effect for the currently nonlinear functions of *agebirM* and *ecstatH*.
Finally, based on model (M2) we examine the spatial structure of the residuals after controlling for covariates to see whether we have been able to explain the spatial pattern of undernutrition. The kernel density estimates in figure 8.21 show that we have been able to significantly reduce the spatial information which is left in the residuals. Compared to the distribution of the spatial effects before using covariates (dotted line), the solid line shows a much tighter distribution of the residual spatial effects. Nevertheless, a distinctly spatial pattern of undernutrition remains. When comparing the maps in figure 8.19, one can recognise some notable shifts in the residual spatial patterns compared to the other maps. In particular, the areas of unexplained poor nutritional status have now shifted from the Central-North to the North-West. Conversely, new areas of 'better than expected' female nutrition appear in the East (e.g. in West Bengal and parts of Orissa), while undernutri-

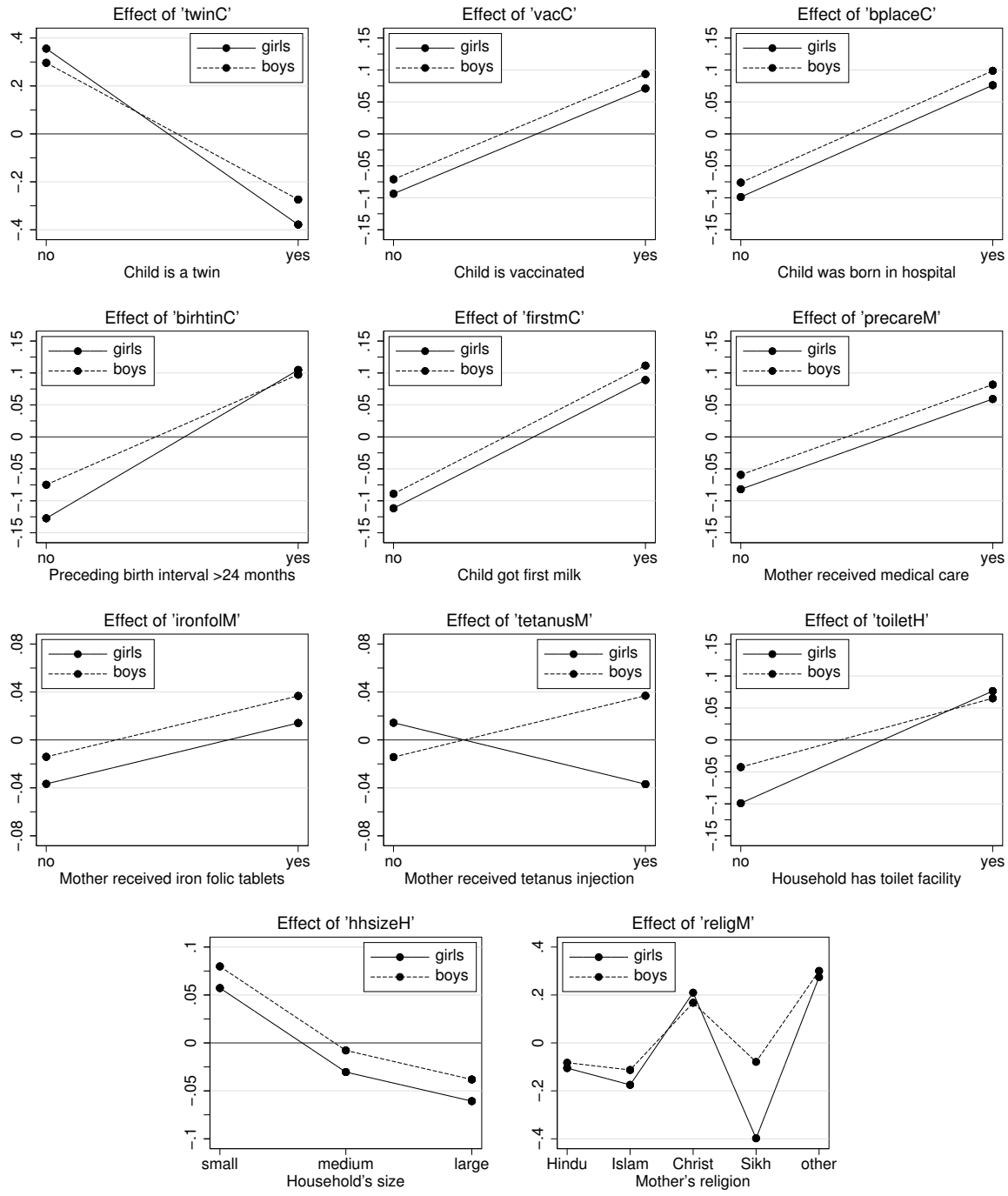

Figure 8.15: Effects of categorical covariates for model (M2).

tion in some areas of the extreme east (e.g. Assam, Manipur, Mizoram and Triupura) is no longer better than expected. Regarding the spatial pattern of the sex differences in undernutrition, our model seems to perform very well. As shown in figure 8.20 (d), there

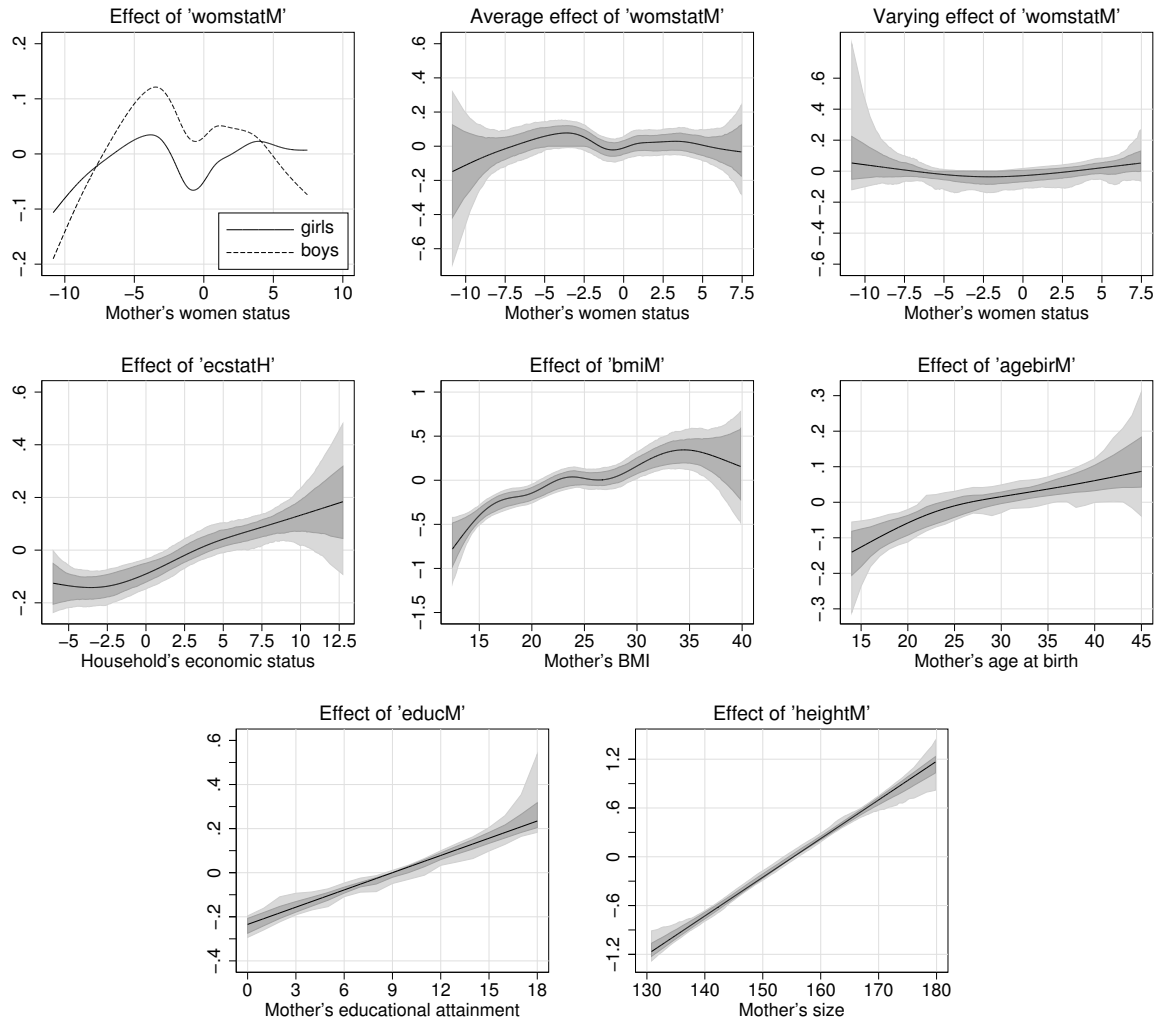

Figure 8.16: Effects of continuous covariates (apart from ageC and bfmC).

are hardly any significant sex differences remaining.

There are several possible explanations for the remaining spatial pattern. One possibility is that our covariates are not sufficiently capturing regional differences due to factors like e.g. different female roles, different public action in the fields of health and nutrition or different religions, although they were designed for that purpose. Or there could exist cultural customs affecting the treatment of children which are not closely correlated with religious affiliation or our measures of female autonomy and might therefore account for the remaining regional pattern. Another possible explanation is that certain aspects of public commitment and public activism are not sufficiently captured by our variables. For instance, the areas of significantly poorer than expected performance are concentrated in areas which recently witnessed the rise of Hindu nationalism, the ascendancy of the Hindu

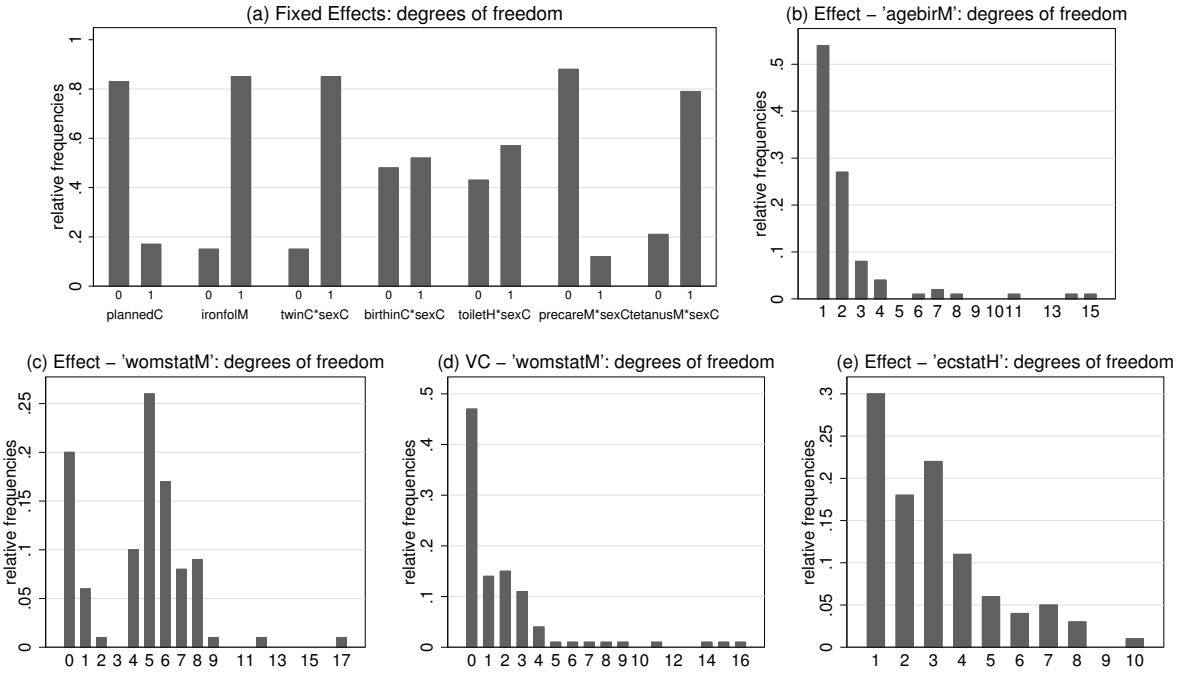

Figure 8.17: Bootstrap sampling distributions for the different modelling alternatives of selected terms.

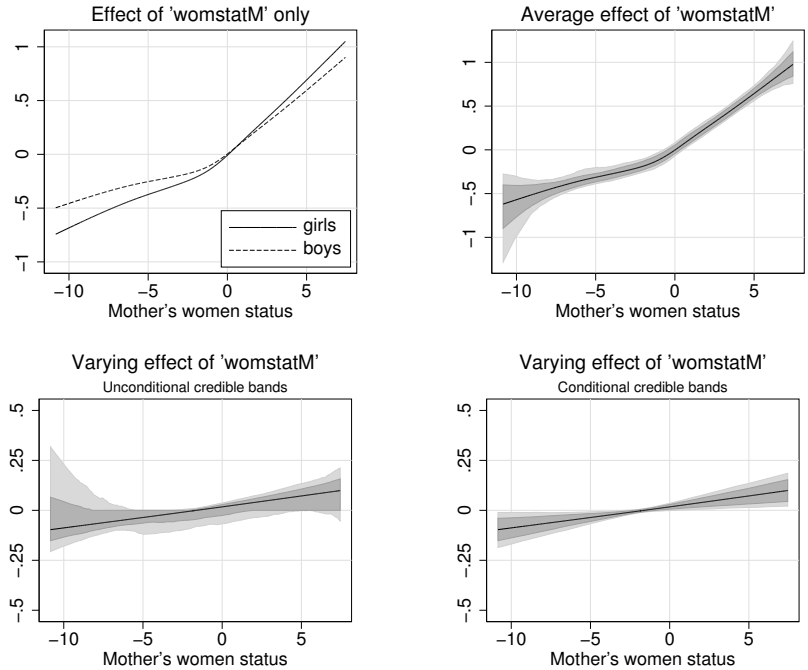

Figure 8.18: Effects of mother's women's status without controlling for other covariates.

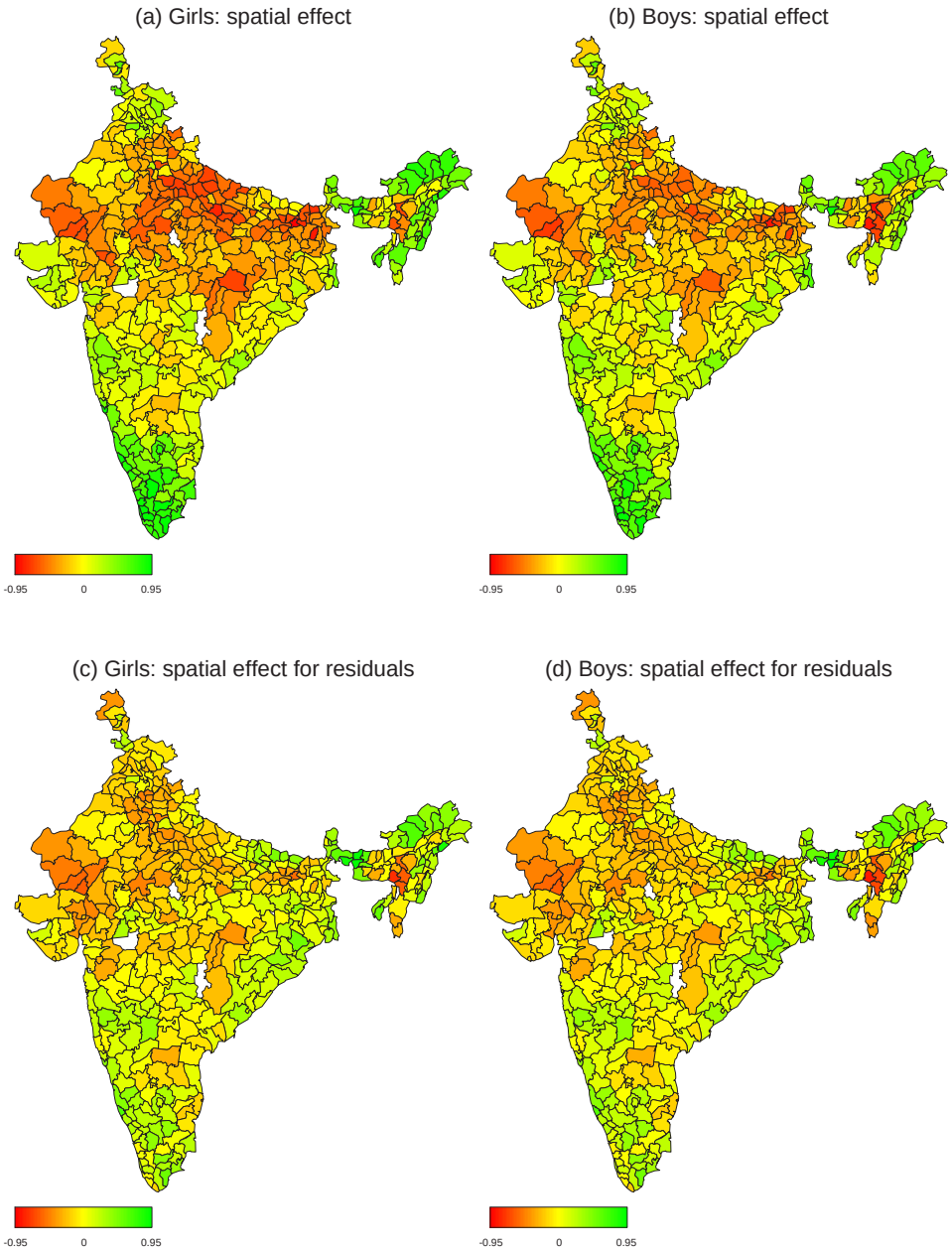

Figure 8.19: Spatial effects for boys and girls without and with controlling for other effects.

nationalist BJP to political prominence, and related incidences of communal violence between Muslims and Hindus. Finally, there could be climatic factors that help to explain these different patterns of undernutrition. We do not have the data at our disposal to investigate these hypotheses but we hope they will stimulate a further analysis of the remaining spatial patterns of undernutrition.

no	term	possible term types	range for df	selected (M1)	selected (M2)
M	$sexC$	linear effect	$\{0,1\}$	df = 1	df = 1
A	$twinC$	linear effect	$\{0,1\}$	df = 1	df = 1
a	$twinC \cdot sexC$	linear effect	$\{0,1\}$	df = 1	df = 1
B	$born1stC$	linear effect	$\{0,1\}$	df = 0	df = 0
b	$born1stC \cdot sexC$	linear effect	$\{0,1\}$	df = 0	df = 0
C	$birthinC$	linear effect	$\{0,1\}$	df = 1	df = 1
c	$birthinC \cdot sexC$	linear effect	$\{0,1\}$	df = $\underline{0}$	df = $\underline{1}$
D	$vacC$	linear effect	$\{0,1\}$	df = 1	df = 1
d	$vacC \cdot sexC$	linear effect	$\{0,1\}$	df = 0	df = 0
E	$firstmC$	linear effect	$\{0,1\}$	df = 1	df = 1
e	$firstmC \cdot sexC$	linear effect	$\{0,1\}$	df = 0	df = 0
F	$toiletH$	linear effect	$\{0,1\}$	df = 1	df = 1
f	$toiletH \cdot sexC$	linear effect	$\{0,1\}$	df = $\underline{0}$	df = $\underline{1}$
G	$bplaceC$	linear effect	$\{0,1\}$	df = 1	df = 1
g	$bplaceC \cdot sexC$	linear effect	$\{0,1\}$	df = 0	df = 0
H	$precareM$	linear effect	$\{0,1\}$	df = 1	df = 1
h	$precareM \cdot sexC$	linear effect	$\{0,1\}$	df = 0	df = 0
I	$ironfolM$	linear effect	$\{0,1\}$	df = 1	df = 1
i	$ironfolM \cdot sexC$	linear effect	$\{0,1\}$	df = 0	df = 0
J	$tetanusM$	linear effect	$\{0,1\}$	df = 0	df = 0
j	$tetanusM \cdot sexC$	linear effect	$\{0,1\}$	df = 1	df = 1
K	$plannedC$	linear effect	$\{0,1\}$	df = 0	df = 0
k	$plannedC \cdot sexC$	linear effect	$\{0,1\}$	df = 0	df = 0
L	$areaH$	linear effect	$\{0,1\}$	df = 0	df = 0
l	$areaH \cdot sexC$	linear effect	$\{0,1\}$	df = 0	df = 0
N	$religM$	linear effects	$\{0,4\}$	df = 4	df = 4
n	$religM \cdot sexC$	linear effects	$\{0,4\}$	df = 4	df = 4
O	$hhsizeH$	linear effects	$\{0,2\}$	df = 2	df = 2
o	$hhsizeH \cdot sexC$	linear effects	$\{0,2\}$	df = 0	df = 0
P	$f_1(agebirM)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 2	df = 2
p	$g_1(agebirM)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 0	df = 0
Q	$f_2(bmiM)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 7	df = 7
q	$g_2(bmiM)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 0	df = 0
R	$f_3(educM)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 1	df = 1
r	$g_3(educM)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 0	df = 0
S	$f_4(heightM)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 1	df = 1
s	$g_4(heightM)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 0	df = 0
T	$f_5(womstatM)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 6	df = 6
t	$g_5(womstatM)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 2	df = 2
U	$f_6(ecstatH)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 3	df = 3
u	$g_6(ecstatH)$	P–spline, straight line	$\{0,1,\ldots,21\}$	df = 0	df = 0
V	$f_7(ageC)$	P–spline, straight line	$\{0,1,\ldots,16\}$	—	df = 15
v	$g_7(ageC)$	P–spline, straight line	$\{0,1,\ldots,16\}$	—	df = 5
W	$f_8(bfmC)$	P–spline, straight line	$\{0,1,\ldots,16\}$	—	df = 8
w	$g_8(bfmC)$	P–spline, straight line	$\{0,1,\ldots,16\}$	—	df = 0
X	$f_9(ageC, bfmC)$	2D P–spline, linear eff.	$\{0,1,3,5.5\ldots,58\}$	—	df = 24.5 ($\hat{=}$30.5)
x	$g_9(ageC, bfmC)$	2D P–spline, linear eff.	$\{0,1,3,5.5\ldots,58\}$	—	df = 0
-	$f_7(ageC, bfmC)$	2D P–spline, linear eff.	$\{0,1,5,7.5,\ldots,90\}$	df = 52.5	—
-	$g_7(ageC, bfmC)$	2D P–spline, linear eff.	$\{0,1,5,7.5,\ldots,90\}$	df = 7.5	—

Table 8.6: Summary of possible term types and degrees of freedom. The last two columns show the degrees of freedom chosen for the final models (M1) and (M2). Differences between (M1) and (M2) are underlined. Column no *yields the letters for figure 8.13. All functions f_j refer to average effects whereas functions g_j indicate varying coefficients regarding gender. For a better readability, the interaction effects for* ageC *and* bfmC *are optically separated.*

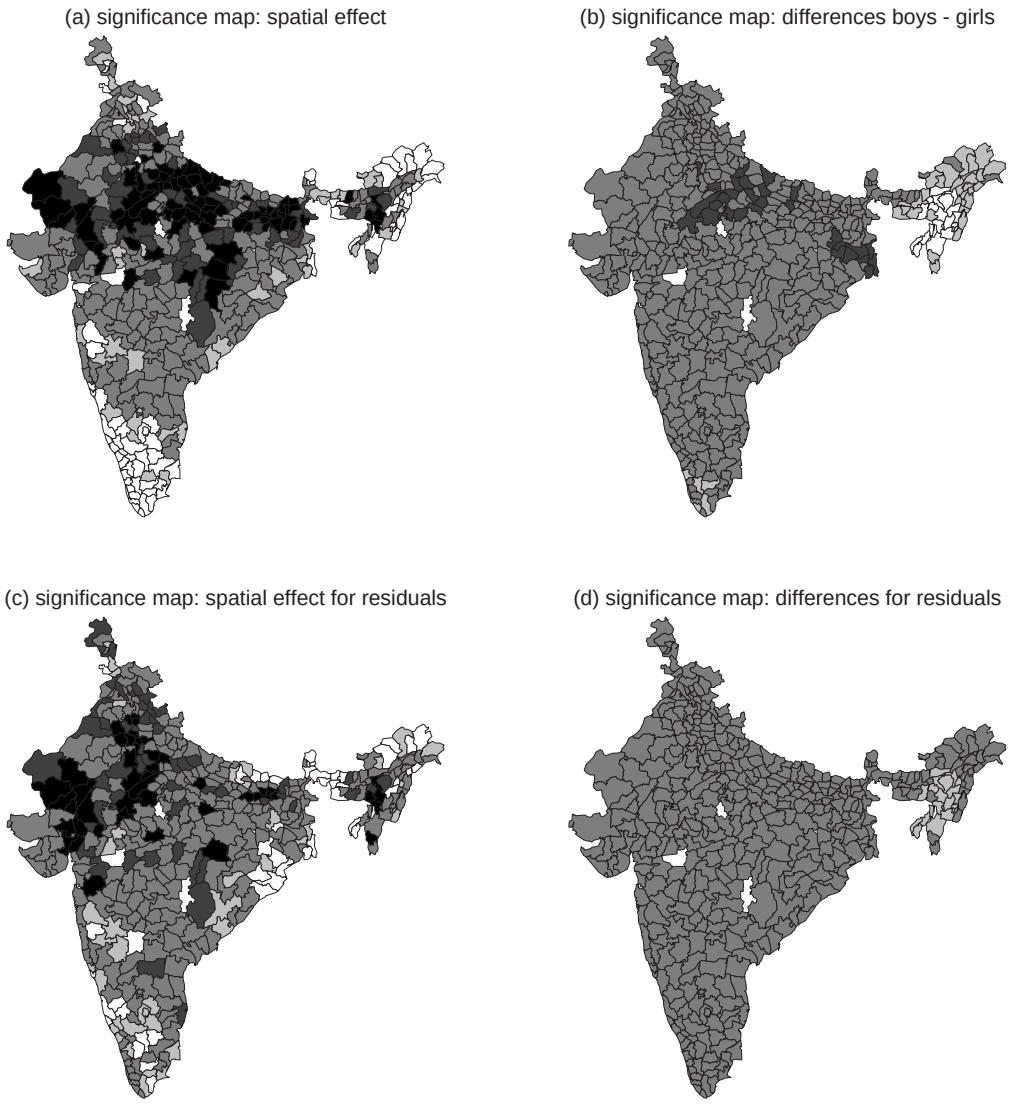

Figure 8.20: Significance maps indicating significant positive (white or light grey) and significant negative regions (black or dark grey) at both 80% and 95% levels (white/black) or at 80% level (otherwise).

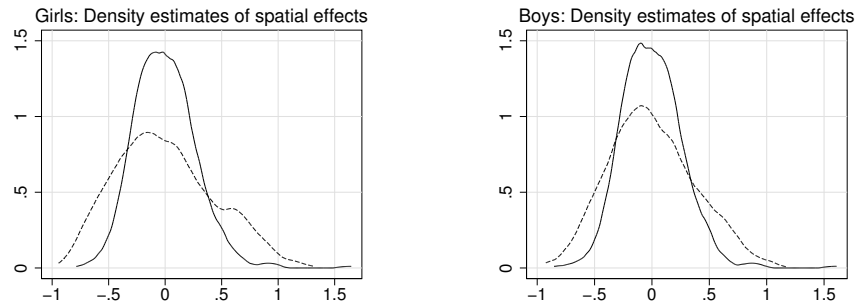

Figure 8.21: Kernel density estimates for the spatial effects.

Chapter 9

Conclusion

In this thesis, we dealt with structured additive regression models which are based on a very flexible predictor. This predictor allows an appropriate modelling for different types of information, e.g. by using smooth functions for spatial information or nonlinear functions for the effects of continuous covariates. We addressed certain aspects of STAR models in particular detail:

- Among the different types of effects, we especially examined complex interaction terms between two (or in some cases even three) covariates. One type of interaction effects are varying coefficients. We used them in the applications from chapter 8 to determine the covariates which require a modelling by sex–varying effects. Regarding varying coefficients, we discovered that the performance of backfitting algorithm and selection algorithms improves if the interacting variable 'gender' is effect coded rather than dummy coded. For the same reason, a continuous interacting variable should get centered.

- Moreover, we want to point out the ANOVA type decomposition for the modelling of a complex interaction effect of two continuous covariates. Here, we introduced the possibility of first estimating a two–dimensional surface and of obtaining main effects and interaction component only afterwards. This approach prevents identifiability problems that would occur between main effects and interaction term if all components were estimated separately. Nevertheless, our approach can treat each component differently by using a combination of three penalty terms. Moreover, the ANOVA type decomposition covers some interesting special cases as shown in section A.4 of the appendix. One simulation study of chapter 7 was especially constructed for the examination of the ANOVA type decomposition. Here, the ANOVA type decomposition yielded comparable results to an approach with separately specified

main effects and interaction term. And it was clearly superior to a surface estimator based on the simple penalty term which is usually used in this context.

- A central aspect of the thesis is the question of model selection in STAR models. Hence, in chapters 3 and 4 we introduced selection algorithms that can automatically select a good model among a large set of possible models. These algorithms can not only perform a variable selection and decide which covariates and terms are relevant but can also determine which degree of smoothness is appropriate for nonlinear functions and can choose between a linear effect and a nonlinear function for the effect of continuous covariates. An important aspect was the computational efficiency of the selection algorithms because the selection should still be feasible for data sets with many potential covariates and many observations. This aspect was particularly realised with the adaptive search algorithm which is by far the fastest approach. The fact that very complex models (even with many possible interaction terms) can be automatically selected within a few minutes makes the adaptive search a tool of high practical relevance.

- Our approaches base model selection on goodness of fit criteria and provide several of the most widely used, like e.g. AIC, BIC or GCV. Most of these criteria include the degrees of freedom of the model as a measure of model complexity. We approximate the degrees of freedom of a model by the degrees of freedom of the individual functions as described in Hastie & Tibshirani (1990). This approach proved to have limitations when functions are highly correlated. In a structured additive predictor, this problem always occurs between an i.i.d. Gaussian random effect or a seasonal component and the intercept term and between a structured and an unstructured spatial effect. For these special cases, we efficiently compute the degrees of freedom of both correlated functions together.

- All selection algorithms of chapter 3 are based on the backfitting algorithm, i.e. they use the backfitting algorithm to obtain estimates for the selected model. That means that credible intervals for regression parameters and nonlinear functions of the selected model are not easily available. Since credible bands for nonlinear functions are an important tool for a further analysis, chapter 5 described a hybrid MCMC approach for the computation of conditional bands (conditional on the selected model) and an approach based on a combination of bootstrap methods and MCMC techniques for unconditional bands (originally introduced by Wood (2006c)). In the simulation study in chapter 7, the unconditional bands frequently showed undercoverage whereas the unconditional bands solved this problem. Often there was, however, only a slight difference in the credible bands of both approaches which is practically not

visible in plots. Hence, if only credible bands of nonlinear functions are required the faster conditional approach would be preferable. The advantage of the unconditional approach is that it yields a sampling distribution for the selected model and, thus, allows a sensitivity analysis regarding model selection. For each term, the marginal frequency distribution reveals the certainty for the selected modelling alternative. Hence, if the stability of the selected model (or merely the certainty of the representation of one covariate) is of interest, the unconditional approach is an appropriate method.

- In several extensive simulation studies we compared our selection algorithms to competing approaches. Here, we discovered that, regarding the quality of estimated functions and estimated predictor, our results are equally good compared to the competing approaches. Moreover, our adaptive search algorithm proved to be by far the most efficient approach: In complex situations where the R software package *mgcv* needed more than a week for the estimation of all replications the adaptive search needed merely one hour. Additionally, our selection algorithms could estimate complex models where *mgcv* failed due to convergence problems.

- We also analysed real applications with our methodology and presented the results in chapter 8. In both applications, our selection algorithm had to cope with a large number of observations and available terms. Moreover, in each case we placed a focus on sex–varying effects and thus further increased the number of terms. The first application was based on data from a Belgian insurance company regarding damage events in car insurance and consisted of two separate models for the response variables *claim frequency* and *claim size* (given a claim occurred). The data was already analysed by Denuit & Lang (2004) with a fully Bayesian approach who had to tediously select appropriate models by hand. We could use their findings to judge the plausibility of our results. In fact, our automatical selection approach selected similar models but needed only a few minutes (in spite of the many possible terms and the large number of observations).
  The second application examined childhood undernutrition in India. Here, two of the covariates (child's age and duration of breastfeeding) are interrelated so that a representation by two separate main effects was probably inadequate. Hence, we had to use an appropriate interaction term. The ANOVA type decomposition was used and yielded very interesting results: For the average effect of boys and girls an interaction term was selected, whereas for the varying effect (i.e. for the difference between boys and girls) only the main effect of the child's age was relevant.

- In summary, the methodology presented in this thesis is of a high practical relevance

and can be applied to problems in many different fields. Since we implemented the methodology in the programming language C++ within the software package *BayesX* our methods are available for everyone.

There are several possible future extensions for our selection algorithms:

- The methodology could easily be adapted to the context of survival models with a structured additive predictor as described in Kneib & Fahrmeir (2007) based on REML estimation or in Hennerfeind, Brezger & Fahrmeir (2006) based on a fully Bayesian approach estimated by MCMC techniques.
- Furthermore, with Gaussian response variables, the variance could also be modelled by a structured additive predictor allowing for heteroscedastic regression models.
- The adaptive selection algorithm works similar as boosting (for boosting compare Bühlmann & Yu (2003)). It would be interesting to examine common characteristics and differences between these two approaches more closely.
- Most selection criteria described in chapter 3 are based on the degrees of freedom of a model. Our algorithms use only an approximation of this number. As was shown in chapter 3, this approximation is rather accurate for a large number of observations as was the case in the applications in chapter 8. However, a more thorough investigation of its accuracy in cases with a small number of observations would be desirable. Moreover, the approximation seems to overestimate the true number in all cases. Hence, a further issue would be to investigate if the approximation is always conservative and, if so, to prove the fact.
- The methodology in chapter 5 for unconditional credible bands additionally offers the possibility to compute model averaged effects and model averaged expectations of the response variable. The topic of model averaging in structured additive regression models also requires further research.

Appendix A

Details about ANOVA type interaction models

A.1 Decomposition of a tensor product spline into one–dimensional splines

In this section we want to examine the conditions that permit an exact decomposition of a two–dimensional tensor product B–spline into two one–dimensional B–splines, i.e.

$$f(x_1, x_2) = \sum_{j,k=1}^{p} \beta_{jk} B_j(x_1) B_k(x_2) \stackrel{!}{=} \sum_{j=1}^{p} a_j B_j(x_1) + \sum_{k=1}^{p} b_k B_k(x_2) = f_1(x_1) + f_2(x_2).$$

In order to show these conditions, we have to reformulate the formulae of both one– and two–dimensional B–splines first.
One–dimensional B–spline basis functions of degree $l \geq 0$ possess two important characteristics which hold at every point in the range of the variable:

1. Only $l + 1$ of the p basis functions $B_1, \ldots, B_p$ are positive at every sinlge point.

2. The $l + 1$ positive basis functions sum up to the value one at a single point.

Suppose at value $x_0 \in [x_{min}; x_{max}]$ the basis functions B_a to B_{a+l} are positive where index a can take every number between 1 and $p - l$ (dependent on value x_0). Then, by using the

two facts mentioned above, we get

$$\begin{aligned}\sum_{j=1}^{p} a_j B_j &\overset{1.}{=} \sum_{j=0}^{l} a_{a+j} B_{a+j} \\ &\overset{2.}{=} \sum_{j=0}^{l-1} a_{a+j} B_{a+j} + a_{a+l}\left(1 - \sum_{j=0}^{l-1} B_{a+j}\right) \\ &= \sum_{j=0}^{l-1} (a_{a+j} - a_{a+l}) B_{a+j} + a_{a+l}, \end{aligned} \tag{A.1}$$

where we set $B_j := B_j(x_0)$ for simplicity.

This is similar for two–dimensional tensor product B–splines. We use the same degree $l \geq 0$ for the one–dimensional basis functions of x_1 and x_2. Then, the characteristics 1. and 2. for one–dimensional B–splines are still valid so that altogether $(l+1)^2$ two–dimensional basis functions are positive at every point in the common range of x_1 and x_2. Suppose, the basis functions $B_a^{(1)}$ to $B_{a+l}^{(1)}$ are positive for variable x_1 and $B_b^{(2)}$ to $B_{b+l}^{(2)}$ for variable x_2, where the upper index indicates the respective covariate. Each of the indices a and b can take some value between 1 and $p-l$ (dependent on the value of the covariate). Using these facts, we get

$$\begin{aligned} &\sum_{j=1}^{p}\sum_{k=1}^{p} \beta_{jk} B_j^{(1)} B_k^{(2)} \\ \overset{1.}{=}\ &\sum_{j=0}^{l}\sum_{k=0}^{l} \beta_{a+j,b+k} B_{a+j}^{(1)} B_{b+k}^{(2)} \\ \overset{2.}{=}\ &\sum_{j=0}^{l-1}\sum_{k=0}^{l-1} \beta_{a+j,b+k} B_{a+j}^{(1)} B_{b+k}^{(2)} + \sum_{k=0}^{l-1} \beta_{a+l,b+k}\left(1 - \sum_{j=0}^{l-1} B_{a+j}^{(1)}\right) B_{b+k}^{(2)} + \\ &\sum_{j=0}^{l-1} \beta_{a+j,b+l} B_{a+j}^{(1)}\left(1 - \sum_{k=0}^{l-1} B_{b+k}^{(2)}\right) + \beta_{a+l,b+l}\left(1 - \sum_{j=0}^{l-1} B_{a+j}^{(1)}\right)\left(1 - \sum_{k=0}^{l-1} B_{b+k}^{(2)}\right) \\ =\ &\sum_{j=0}^{l-1}\sum_{k=0}^{l-1} \beta_{a+j,b+k} B_{a+j}^{(1)} B_{b+k}^{(2)} + I + II + III \end{aligned}$$

Formula I can be simplified to

$$\begin{aligned} I &= \sum_{k=0}^{l-1} \beta_{a+l,b+k}\left(1 - \sum_{j=0}^{l-1} B_{a+j}^{(1)}\right) B_{b+k}^{(2)} \\ &= \sum_{k=0}^{l-1} \beta_{a+l,b+k} B_{b+k}^{(2)} - \sum_{k=0}^{l-1} \beta_{a+l,b+k}\left(\sum_{j=0}^{l-1} B_{a+j}^{(1)}\right) B_{b+k}^{(2)} \end{aligned}$$

$$= \sum_{k=0}^{l-1} \beta_{a+l,b+k} B_{b+k}^{(2)} - \sum_{j=0}^{l-1} \sum_{k=0}^{l-1} \beta_{a+l,b+k} B_{a+j}^{(1)} B_{b+k}^{(2)}.$$

Similarly, formula *II* can be rewritten as

$$\begin{aligned} II &= \sum_{j=0}^{l-1} \beta_{a+j,b+l} B_{a+j}^{(1)} \left(1 - \sum_{k=0}^{l-1} B_{b+k}^{(2)}\right) \\ &= \sum_{j=0}^{l-1} \beta_{a+j,b+l} B_{a+j}^{(1)} - \sum_{j=0}^{l-1} \beta_{a+j,b+l} B_{a+j}^{(1)} \left(\sum_{k=0}^{l-1} B_{b+k}^{(2)}\right) \\ &= \sum_{j=0}^{l-1} \beta_{a+j,b+l} B_{a+j}^{(1)} - \sum_{j=0}^{l-1} \sum_{k=0}^{l-1} \beta_{a+j,b+l} B_{a+j}^{(1)} B_{b+k}^{(2)}. \end{aligned}$$

And formula *III* can be rewritten as

$$\begin{aligned} III &= \beta_{a+l,b+l} \left(1 - \sum_{j=0}^{l-1} B_{a+j}^{(1)}\right) \left(1 - \sum_{k=0}^{l-1} B_{b+k}^{(2)}\right) \\ &= \beta_{a+l,b+l} \sum_{j=0}^{l-1} B_{a+j}^{(1)} \sum_{k=0}^{l-1} B_{b+k}^{(2)} - \beta_{a+l,b+l} \sum_{j=0}^{l-1} B_{a+j}^{(1)} - \beta_{a+l,b+l} \sum_{k=0}^{l-1} B_{b+k}^{(2)} + \beta_{a+l,b+l} \\ &= \sum_{j=0}^{l-1} \sum_{k=0}^{l-1} \beta_{a+l,b+l} B_{a+j}^{(1)} B_{b+k}^{(2)} - \sum_{j=0}^{l-1} \beta_{a+l,b+l} B_{a+j}^{(1)} - \sum_{k=0}^{l-1} \beta_{a+l,b+l} B_{b+k}^{(2)} + \beta_{a+l,b+l} \end{aligned}$$

By summarising corresponding terms, a two–dimensional tensor product B–spline can be written as

$$\begin{aligned} &\sum_{j=1}^{p} \sum_{k=1}^{p} \beta_{jk} B_j^{(1)} B_k^{(2)} \\ = &\sum_{j=0}^{l-1} \sum_{k=0}^{l-1} (\beta_{a+j,b+k} - \beta_{a+l,b+k} - \beta_{a+j,b+l} + \beta_{a+l,b+l}) B_{a+j}^{(1)} B_{b+k}^{(2)} + \\ &\sum_{j=0}^{l-1} (\beta_{a+j,b+l} - \beta_{a+l,b+l}) B_{a+j}^{(1)} + \sum_{k=0}^{l-1} (\beta_{a+l,b+k} - \beta_{a+l,b+l}) B_{b+k}^{(2)} + \beta_{a+l,b+l}. \end{aligned} \quad \text{(A.2)}$$

Using the alternative respresentations (A.1) and (A.2), we can easily see under which conditions a two–dimensional spline decomposes into two one–dimensional splines, i.e. under which conditions (A.2) is equal to formula

$$\sum_{j=1}^{p} a_j B_j^{(1)} + \sum_{k=1}^{p} b_k B_k^{(2)} = \sum_{j=0}^{l-1} (a_{a+j} - a_{a+l}) B_{a+j}^{(1)} + a_{a+l} + \sum_{k=0}^{l-1} (b_{b+k} - b_{b+l}) B_{b+k}^{(2)} + b_{b+l}.$$

The two formulas are equal if the corresponding coefficients are equal, i.e. if

1. $\beta_{a+l,b+l} = a_{a+l} + b_{b+l}$ which follows from the constant term.
2. $\beta_{a+j,b+l} - \beta_{a+l,b+l} = a_{a+j} - a_{a+l}$ which follows from the coefficients belonging to $B^{(1)}_{a+j}$. Together with 1. we get $\beta_{a+j,b+l} = a_{a+j} + b_{b+l}$ for $j = 0, \ldots, l-1$.
3. $\beta_{a+l,b+k} - \beta_{a+l,b+l} = b_{b+k} - b_{b+l}$ which follows from the coefficients belonging to $B^{(2)}_{b+k}$. Together with 1. we get $\beta_{a+l,b+k} = a_{a+l} + b_{b+k}$ for $k = 0, \ldots, l-1$.
4. $\beta_{a+j,b+k} - \beta_{a+l,b+k} - \beta_{a+j,b+l} + \beta_{a+l,b+l} = 0$ which follows from the coefficients for the mixed terms. Together with 1., 2. and 3. we get $\beta_{a+j,b+k} = a_{a+j} + b_{b+k}$ for $j, k = 0, \ldots, l-1$.

As these relationships have to apply to each combination of values for the indices $a, b = 1, \ldots, p-l$, we get the following general condition for a decomposition of a two–dimensional tensor product in two main effects:

$$\beta_{j,k} = a_j + b_k, \tag{A.3}$$

for $j, k = 1, \ldots, p$.
Alternatively, condition (A.3) can either be rewritten as

$$\beta_{1,i} - \beta_{1,i+1} = \ldots = \beta_{p,i} - \beta_{p,i+1} = b_i - b_{i+1}$$

or as

$$\beta_{i,1} - \beta_{i+1,1} = \ldots = \beta_{i,p} - \beta_{i+1,p} = a_i - a_{i+1}$$

for $i = 1, \ldots, p-1$. Both of these alternative formulations can be equivalently written in form of the following $(p-1)^2$ conditions

$$\begin{aligned}
\beta_{1,1} - \beta_{1,2} - \beta_{2,1} + \beta_{2,2} &= 0, \\
\beta_{2,1} - \beta_{2,2} - \beta_{3,1} + \beta_{3,2} &= 0, \\
&\vdots \\
\beta_{p-1,1} - \beta_{p-1,2} - \beta_{p,1} + \beta_{p,2} &= 0, \\
\beta_{1,2} - \beta_{1,3} - \beta_{2,2} + \beta_{2,3} &= 0, \\
&\vdots \\
\beta_{p-1,p-1} - \beta_{p-1,p} - \beta_{p,p-1} + \beta_{p,p} &= 0.
\end{aligned}$$

Using a two–dimensional difference operator Δ these conditions can be generalised as differences of differences by

$$\Delta^{(1,0)}\Delta^{(0,1)}\beta_{j,k} = \beta_{j,k} - \beta_{j-1,k} - \beta_{j,k-1} + \beta_{j-1,k-1} = 0,$$

for $j, k = 2, \ldots, p$. Summarising these $(p-1)^2$ conditions in a difference matrix $\mathbf{D}$ such that $\mathbf{D}\boldsymbol{\beta} = 0$ leads to difference matrix (2.34).

A.2 Comparison of one– and two–dimensional penalty matrices

In this section we show that matrices $\mathbf{P}_{x_1} = \mathbf{P}_{k_1} \otimes \mathbf{I}_p$ and $\mathbf{P}_{x_2} = \mathbf{I}_p \otimes \mathbf{P}_{k_2}$ used in the overall penalty of the two-dimensional function are p times as strong as the corresponding matrices $\mathbf{P}_{k_1}$ or $\mathbf{P}_{k_2}$. For this purpose, we suppose that the surface exactly decomposes into two main effects, i.e. we suppose that $\boldsymbol{\beta} = (\beta_{11}, \beta_{12}, \ldots, \beta_{pp})' = (a_1 + b_1, a_1 + b_2, \ldots, a_p + b_p)'$. Then we have

$$
\begin{aligned}
& \boldsymbol{\beta}' \mathbf{P}_{x_2} \boldsymbol{\beta} \\
= \; & \boldsymbol{\beta}' \cdot (\mathbf{I}_p \otimes \mathbf{P}_{k_2}) \cdot \boldsymbol{\beta} \\
= \; & (a_1 + b_1, a_1 + b_2, \ldots, a_p + b_p) \cdot (\mathbf{I}_p \otimes \mathbf{P}_{k_2}) \cdot (a_1 + b_1, a_1 + b_2, \ldots, a_p + b_p)' \\
= \; & (a_1 + b_1, a_1 + b_2, \ldots, a_p + b_p) \cdot (\mathbf{I}_p \otimes \mathbf{D}_{k_2})'(\mathbf{I}_p \otimes \mathbf{D}_{k_2}) \cdot (a_1 + b_1, a_1 + b_2, \ldots, a_p + b_p)' \\
= \; & ||(\mathbf{I}_p \otimes \mathbf{D}_{k_2}) \cdot (a_1 + b_1, a_1 + b_2, \ldots, a_p + b_p)'||^2 \\
= \; & \left|\left| \begin{pmatrix} \mathbf{D}_{k_2} \cdot (a_1 + b_1, \ldots, a_1 + b_p)' \\ \vdots \\ \mathbf{D}_{k_2} \cdot (a_p + b_1, \ldots, a_p + b_p)' \end{pmatrix} \right|\right|^2 \\
= \; & \left|\left| \begin{pmatrix} \mathbf{D}_{k_2} \cdot (b_1, \ldots, b_p)' \\ \vdots \\ \mathbf{D}_{k_2} \cdot (b_1, \ldots, b_p)' \end{pmatrix} \right|\right|^2 \\
= \; & p \cdot (b_1, \ldots, b_p) \cdot \mathbf{P}_{k_2} \cdot (b_1, \ldots, b_p)'.
\end{aligned}
$$

This confirms, that the penalty term using the two–dimensional matrix $\mathbf{P}_{x_2}$ is p times as large as the penalty using $\mathbf{P}_{k_2}$.
For the calculation we use the fact that the line total of $\mathbf{D}_{k_2}$ is zero, so that

$$\mathbf{D}_{k_2}(a_i, \ldots, a_i)' = 0$$

for $i = 1, \ldots, p$. From the analogous calculation for matrix $\mathbf{P}_{x_1}$ we get the result

$$(a_1 + b_1, a_1 + b_2, \ldots, a_p + b_p) \cdot (\mathbf{P}_{k_1} \otimes \mathbf{I}_p) \cdot (a_1 + b_1, a_1 + b_2, \ldots, a_p + b_p)' = p \cdot a' \mathbf{P}_{k_1} a.$$

A.3 Extraction of the main effects

In this section we show that the main effects that are extracted from the overall surface are P–splines. For that purpose we use the tensor product representation of the two–dimensional spline. Apart from additive and multiplicative constants function $\hat{f}_2(x_2)$ is

based on the integral $\int_{x_{1,min}}^{x_{1,max}} f(x_1, x_2)dx_1$. This integral can be transformed in the following way:

$$\begin{aligned}
&\int_{x_{1,min}}^{x_{1,max}} f(x_1, x_2)dx_1 \\
= &\int_{x_{1,min}}^{x_{1,max}} \left(\sum_{j=1}^{p} \sum_{k=1}^{p} \beta_{jk} B_j(x_1) B_k(x_2) \right) dx_1 \\
= &\sum_{j=1}^{p} \sum_{k=1}^{p} \beta_{jk} B_k(x_2) \int_{x_{1,min}}^{x_{1,max}} B_j(x_1)dx_1 \\
= &\sum_{k=1}^{p} \underbrace{\left(\sum_{j=1}^{p} \beta_{jk} \int_{x_{1,min}}^{x_{1,max}} B_j(x_1)dx_1 \right)}_{=:\beta_k} B_k(x_2).
\end{aligned}$$

This calculation applies likewise to the integral regarding x_2, i.e. to function $\hat{f}_1(x_1)$.
Thus, the coefficients of the main effect splines are a linear combination of the coefficients β_{jk} of the two–dimensional function with weights based on the integrals over one–dimensional basis functions. The values of these integrals depend on the degree of the respective basis functions and are calculated by using the recursive B–spline definition (2.13).

A.4 Examples for different combinations of smoothing parameters

This section shows examples for different combinations of smoothing parameters in the overall penalty matrix $\mathbf{P}_{comp}$. Table A.1 gives the overview of the combinations whose estimated functions are shown in figures A.1–A.5. The examples are based on the data used for the simulation study in section 7.5.1. The true components f_1, f_2 and f_{inter} are shown in figure 7.37.

	λ	λ_1	λ_2	$f_1(x_1)$	$f_2(x_2)$	$f_{inter}(x_1, x_2)$	
(1)	∞	∞ (rw1)	∞ (rw1)	const. ($\equiv 0$)	const. ($\equiv 0$)	const. ($\equiv 0$)	
(2)	∞	∞ (rw2)	∞ (rw2)	linear	linear	const. ($\equiv 0$)	
(3)	∞	3	∞	smooth nonlin.	linear	const. ($\equiv 0$)	
(4)	∞	3	3	smooth nonlin.	smooth nonlin.	const. ($\equiv 0$)	
(5)	∞	0	∞	rough nonlin.	linear	const. ($\equiv 0$)	
(6)	∞	0	3	rough nonlin.	smooth nonlin.	const. ($\equiv 0$)	
(7)	∞	0	0	rough nonlin.	rough nonlin.	const. ($\equiv 0$)	
(8)	0.6	∞ (rw1)	∞ (rw1)	const. ($\equiv 0$)	const. ($\equiv 0$)	const. ($\equiv 0$)	
(9)	0.6	∞ (rw2)	∞ (rw2)	linear	linear	linear	
(10)	0.6	∞	3	linear	smooth nonlin.	smooth nonlin.	
(11)	0.6	3	3	smooth nonlin.	smooth nonlin.	smooth nonlin.	
(12)	0.6	∞	0	linear	rough nonlin.	smooth nonlin.	
(13)	0.6	3	0	smooth nonlin.	rough nonlin.	rough nonlin.	
(14)	0.6	0	0	rough nonlin.	rough nonlin.	rough nonlin.	
(15)	0	∞ (rw1)	∞ (rw1)	const. ($\equiv 0$)	const. ($\equiv 0$)	const. ($\equiv 0$)	**
(16)	0	∞ (rw2)	∞ (rw2)	linear	linear	linear	**
(17)	0	∞	3	linear	smooth nonlin.	smooth nonlin.	*
(18)	0	3	3	smooth nonlin.	smooth nonlin.	smooth nonlin.	**
(19)	0	∞	0	linear	rough nonlin.	rough nonlin.	*
(20)	0	3	0	smooth nonlin.	rough nonlin.	rough nonlin.	*
(21)	0	0	0	rough nonlin.	rough nonlin.	rough nonlin.	**

Table A.1: Different combinations of smoothing parameters for the ANOVA type interaction. If not stated otherwise, a second order penalty is used for the main effects. For each main effect a spline of third degree with 12 basis functions is used. The values $\lambda_1 = 3$ or $\lambda_2 = 3$ each correspond to a spline with $df = 5$. The value $\lambda = 0.6$ corresponds to a two–dimensional function with $df = 50$ if $\lambda_1 = \lambda_2 = 0$. Symbol $*$ indicates cases in which the complete two–dimensional function is equal to the approach by Eilers and Marx (2003) and symbol $**$ indicates cases in which the complete two–dimensional function is additionally equal to Lang and Brezger (2004).

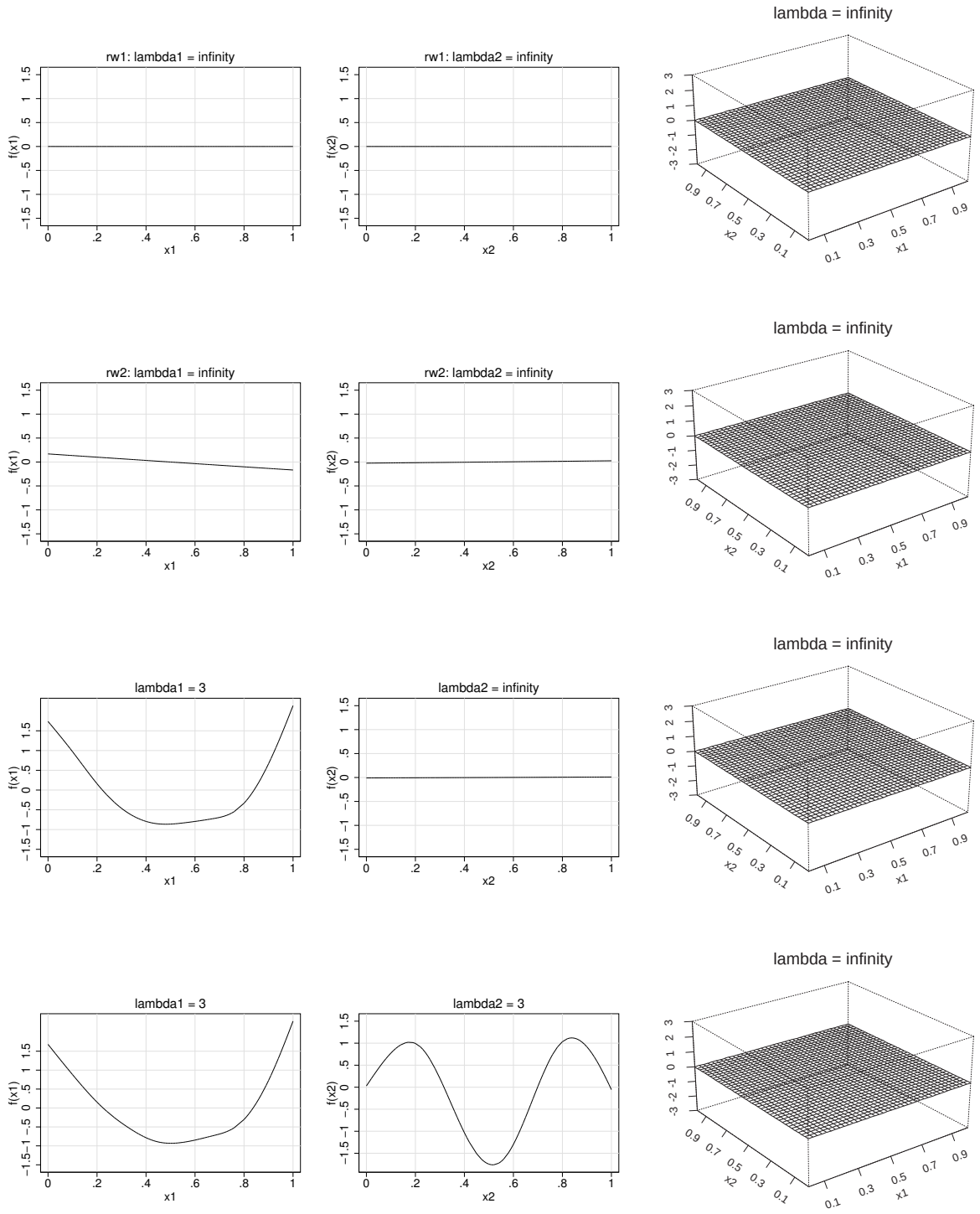

Figure A.1: Examples for different combinations of smoothing parameters in ANOVA type interaction models. Shown are cases (1) to (4) from table A.1.

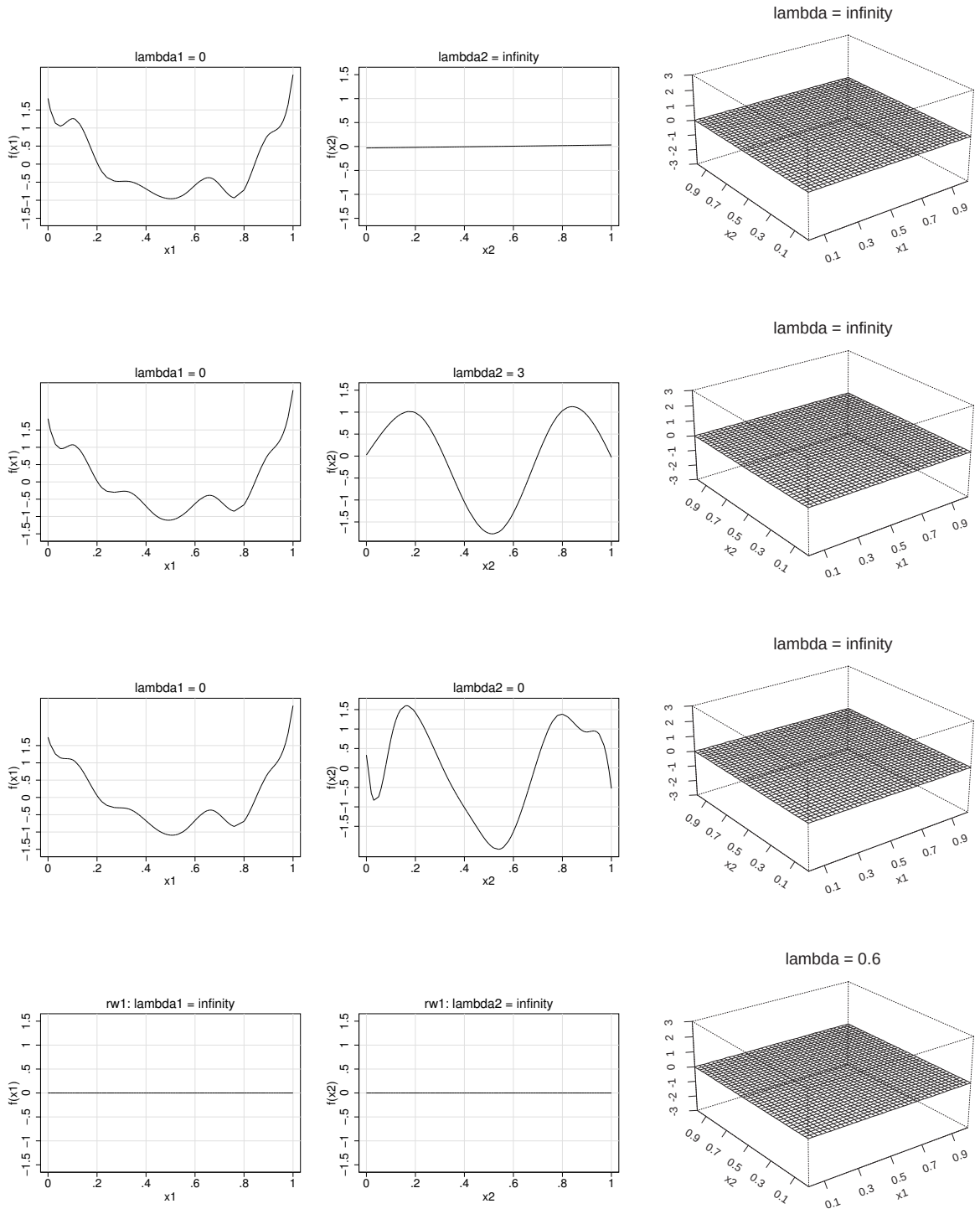

Figure A.2: Examples for different combinations of smoothing parameters in ANOVA type interaction models. Shown are cases (5) to (8) from table A.1.

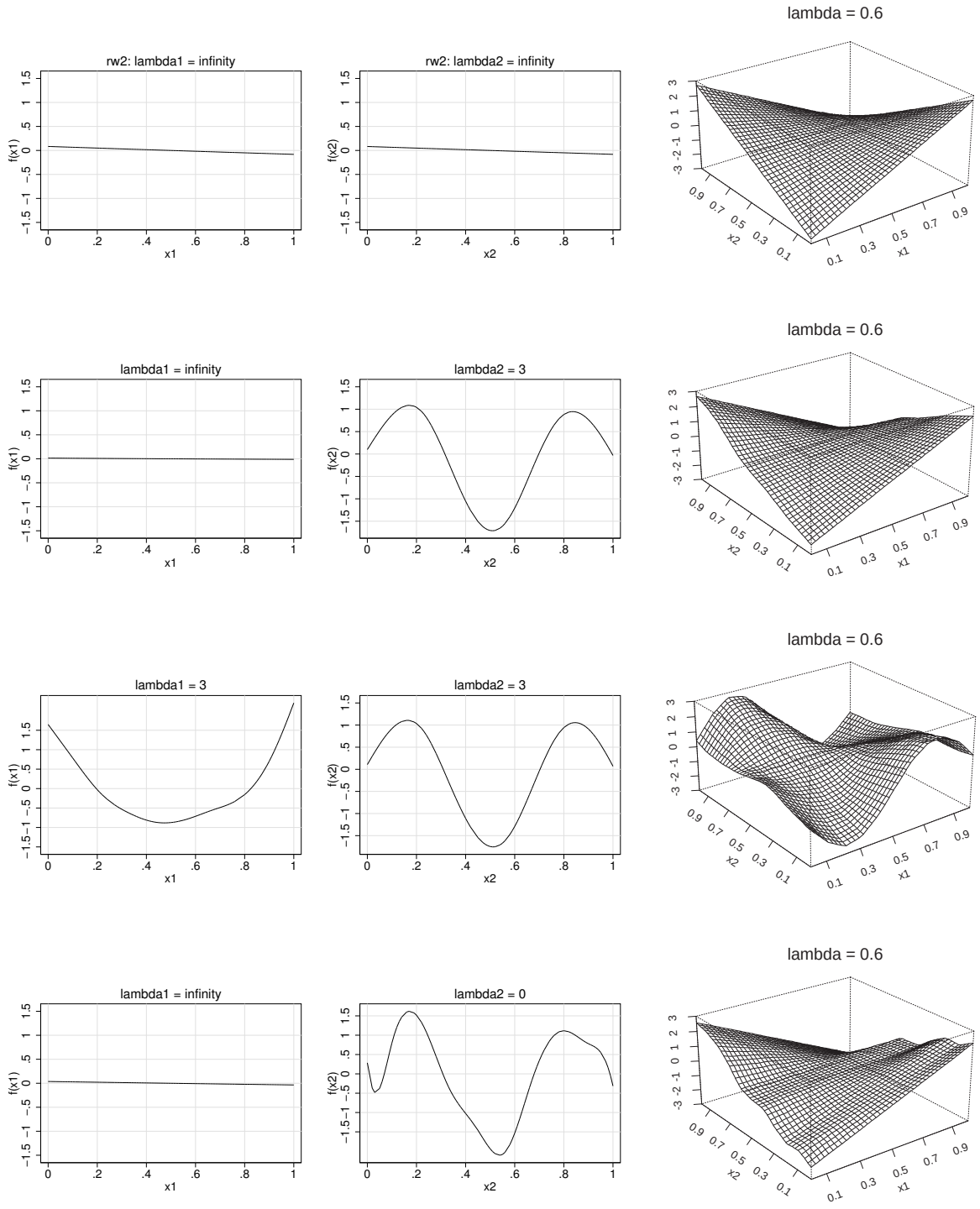

Figure A.3: Examples for different combinations of smoothing parameters in ANOVA type interaction models. Shown are cases (9) to (12) from table A.1.

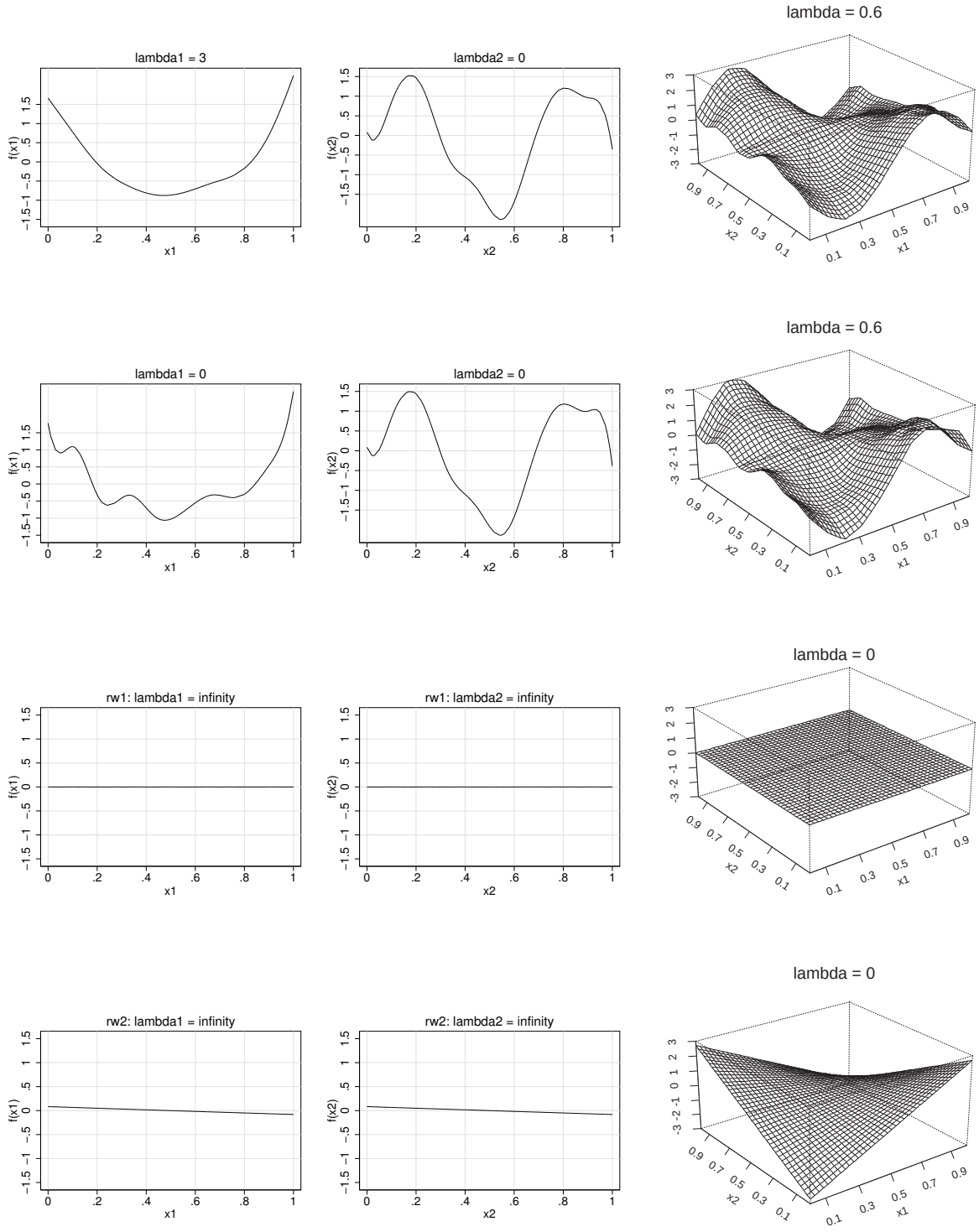

Figure A.4: Examples for different combinations of smoothing parameters in ANOVA type interaction models. Shown are cases (13) to (16) from table A.1.

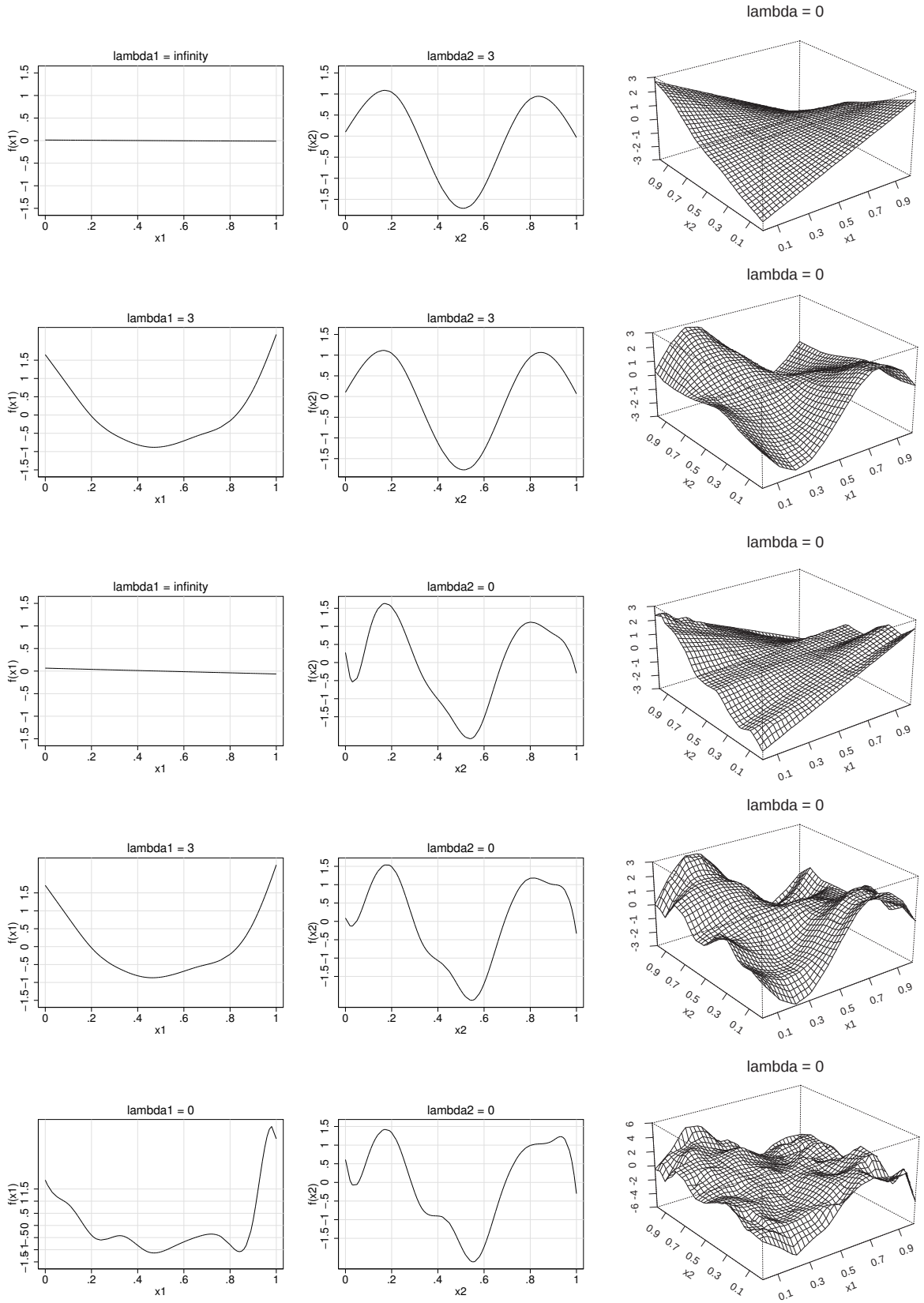

Figure A.5: Examples for different combinations of smoothing parameters in ANOVA type interaction models. Shown are cases (17) to (21) from table A.1.

Appendix B

Details about the calculation of degrees of freedom

B.1 Degrees of freedom for i.i.d. Gaussian random effects

In section 3.3, we consider the simple predictor

$$\eta = \gamma_0 + f_{ran}(x),$$

containing an intercept term and an i.i.d. Gaussian random effect with p individuals. For $\lambda_{ran} > 0$, the true degrees of freedom for this simple predictor can be calculated from the overall hat matrix by using formula (3.24)

$$df_{ran} = \operatorname{tr}\left\{ (\mathbf{X}_{ran},\, \mathbf{1}) \left[(\mathbf{X}_{ran},\, \mathbf{1})' \mathbf{W} (\mathbf{X}_{ran},\, \mathbf{1}) + \lambda_{ran}\, \operatorname{diag}(\mathbf{I}_p, 0) \right]^{-1} (\mathbf{X}_{ran},\, \mathbf{1})' \mathbf{W} \right\} - 1,$$

where $\mathbf{1}$ is the vector containing value one only. In this section we show the derivation of the efficient formula (3.25) that allows to calculate the degrees of freedom by computing only the necessary elements of the respective hat matrix.
An equivalent representation of formula (3.24) is

$$df_{ran} = \operatorname{tr}\left\{ (\mathbf{X}_{ran},\, \mathbf{1})' \mathbf{W} (\mathbf{X}_{ran},\, \mathbf{1}) \left[(\mathbf{X}_{ran},\, \mathbf{1})' \mathbf{W} (\mathbf{X}_{ran},\, \mathbf{1}) + \lambda_{ran}\, \operatorname{diag}(\mathbf{I}_p, 0) \right]^{-1} \right\} - 1$$

which is used later for computing the trace. But first, we have to calculate the inverse matrix

$$\begin{aligned} \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B}' & \mathcal{C} \end{pmatrix}^{-1} &:= \left[(\mathbf{X}_{ran},\, \mathbf{1})' \mathbf{W} (\mathbf{X}_{ran},\, \mathbf{1}) + \lambda_{ran}\, \operatorname{diag}(\mathbf{I}_p, 0) \right]^{-1} \\ &= \begin{pmatrix} \mathbf{X}'_{ran} \mathbf{W} \mathbf{X}_{ran} + \lambda_{ran} \mathbf{I} & \mathbf{X}'_{ran} \mathbf{W} \mathbf{1} \\ \mathbf{1}' \mathbf{W} \mathbf{X}_{ran} & \mathbf{1}' \mathbf{W} \mathbf{1} \end{pmatrix}^{-1}. \end{aligned}$$

As can easily be verified, the inverse matrix of a matrix containing four submatrices is given by (compare Magnus & Neudecker (1991))

$$\begin{pmatrix} \mathcal{X} & \mathcal{Y}' \\ \mathcal{Y} & \mathcal{Z} \end{pmatrix} := \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B}' & \mathcal{C} \end{pmatrix}^{-1} = \begin{pmatrix} \mathcal{A}^{-1}(\mathbf{I} + \mathcal{B}(\mathcal{C} - \mathcal{B}'\mathcal{A}^{-1}\mathcal{B})^{-1}\mathcal{B}'\mathcal{A}^{-1}) & -\mathcal{A}^{-1}\mathcal{B}(\mathcal{C} - \mathcal{B}'\mathcal{A}^{-1}\mathcal{B})^{-1} \\ -(\mathcal{C} - \mathcal{B}'\mathcal{A}^{-1}\mathcal{B})^{-1}\mathcal{B}'\mathcal{A}^{-1} & (\mathcal{C} - \mathcal{B}'\mathcal{A}^{-1}\mathcal{B})^{-1} \end{pmatrix} \quad \text{(B.1)}$$

We start with calculating submatrix $\mathcal{Z}$ as this is contained in the other two submatrices. Matrix $\mathcal{Z}$ is actually a scalar and obtained by

$$\begin{aligned} \mathcal{Z} &= (\mathbf{1}'\mathbf{W}\mathbf{1} - \mathbf{1}'\mathbf{W}\mathbf{X}_{ran}(\mathbf{X}'_{ran}\mathbf{W}\mathbf{X}_{ran} + \lambda_{ran}\mathbf{I})^{-1}\mathbf{X}'_{ran}\mathbf{W}\mathbf{1})^{-1} \\ &= \left(n - (n_1, \ldots, n_p)\ \mathrm{diag}\left(\frac{1}{n_1 + \lambda_{ran}}, \ldots, \frac{1}{n_p + \lambda_{ran}}\right)(n_1, \ldots, n_p)'\right)^{-1} \\ &= \left(n - \sum_{k=1}^{p} \frac{n_k^2}{n_k + \lambda_{ran}}\right)^{-1} =: c, \end{aligned}$$

with $n_k = \sum_{1 \le i \le n : x_{ik} = 1} w_i$ for $k = 1, \ldots, p$ and $n = \sum_{k=1}^{p} n_k$. For random effects, the design matrix $\mathbf{X}_{ran}$ is a 0/1–incidence matrix so that matrix $(\mathbf{X}'_{ran}\mathbf{W}\mathbf{X}_{ran} + \lambda_{ran}\mathbf{I})$ and its inverse are diagonal.

Using the above result, matrix $\mathcal{Y}$ can be transformed to

$$\begin{aligned} \mathcal{Y} &= -c\,\mathbf{1}'\mathbf{W}\mathbf{X}_{ran}(\mathbf{X}'_{ran}\mathbf{W}\mathbf{X}_{ran} + \lambda_{ran}\mathbf{I})^{-1} \\ &= -c\left(\frac{n_1}{n_1 + \lambda_{ran}}, \ldots, \frac{n_p}{n_p + \lambda_{ran}}\right). \end{aligned}$$

The most complex submatrix is $\mathcal{X}$ that can be reformulated as

$$\begin{aligned} \mathcal{X} &= (\mathbf{X}'_{ran}\mathbf{W}\mathbf{X}_{ran} + \lambda_{ran}\mathbf{I})^{-1}(\mathbf{I} - \mathbf{X}'_{ran}\mathbf{W}\mathbf{1} \cdot \mathcal{Y}) \\ &= \mathrm{diag}\left(\frac{1}{n_1 + \lambda_{ran}}, \ldots, \frac{1}{n_p + \lambda_{ran}}\right)\left[\mathbf{I} + (n_1, \ldots, n_p)'\left(\frac{n_1}{n_1 + \lambda_{ran}}, \ldots, \frac{n_p}{n_p + \lambda_{ran}}\right)c\right] \\ &= \mathrm{diag}\left(\frac{1}{n_1 + \lambda_{ran}}, \ldots, \frac{1}{n_p + \lambda_{ran}}\right)\left[\mathbf{I} + \begin{pmatrix} \frac{n_1^2}{n_1+\lambda_{ran}} & \frac{n_1 n_2}{n_2+\lambda_{ran}} & \cdots & \frac{n_1 n_p}{n_p+\lambda_{ran}} \\ \vdots & & & \vdots \\ \frac{n_1 n_p}{n_1+\lambda_{ran}} & \cdots & \cdots & \frac{n_p^2}{n_p+\lambda_{ran}} \end{pmatrix} c\right] \\ &= \mathrm{diag}\left(\frac{1}{n_1 + \lambda_{ran}}, \ldots, \frac{1}{n_p + \lambda_{ran}}\right)\begin{pmatrix} \frac{cn_1^2+n_1+\lambda_{ran}}{n_1+\lambda_{ran}} & \frac{cn_1 n_2}{n_2+\lambda_{ran}} & \cdots & \frac{cn_1 n_p}{n_p+\lambda_{ran}} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{cn_1 n_p}{n_1+\lambda_{ran}} & \cdots & \cdots & \frac{cn_p^2+n_p+\lambda_{ran}}{n_p+\lambda_{ran}} \end{pmatrix} \\ &= \begin{pmatrix} \frac{cn_1^2+n_1+\lambda_{ran}}{(n_1+\lambda_{ran})^2} & \frac{cn_1 n_2}{(n_1+\lambda_{ran})(n_2+\lambda_{ran})} & \cdots & \frac{cn_1 n_p}{(n_1+\lambda_{ran})(n_p+\lambda_{ran})} \\ \vdots & & & \vdots \\ \frac{cn_1 n_p}{(n_1+\lambda_{ran})(n_p+\lambda_{ran})} & & \cdots & \frac{cn_p^2+n_p+\lambda_{ran}}{(n_p+\lambda_{ran})^2} \end{pmatrix}. \end{aligned}$$

Altogether the inverse matrix is given by

$$\begin{pmatrix} \mathbf{X}'_{ran}\mathbf{W}\mathbf{X}_{ran} + \lambda_{ran}\mathbf{I} & \mathbf{X}'_{ran}\mathbf{W}\mathbf{1} \\ \mathbf{1}'\mathbf{W}\mathbf{X}_{ran} & \mathbf{1}'\mathbf{W}\mathbf{1} \end{pmatrix}^{-1}$$

$$= \begin{pmatrix} \frac{cn_1^2+n_1+\lambda_{ran}}{(n_1+\lambda_{ran})^2} & \dots & \frac{cn_1n_p}{(n_1+\lambda_{ran})(n_p+\lambda_{ran})} & -\frac{cn_1}{n_1+\lambda ran} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{cn_1n_p}{(n_1+\lambda_{ran})(n_p+\lambda_{ran})} & \dots & \frac{cn_p^2+n_p+\lambda_{ran}}{(n_p+\lambda_{ran})^2} & -\frac{cn_p}{n_p+\lambda ran} \\ -\frac{cn_1}{n_1+\lambda ran} & \dots & -\frac{cn_p}{n_p+\lambda ran} & c \end{pmatrix}.$$

The second part in formula (3.24) for calculating the degrees of freedom is product matrix $(\mathbf{X}_{ran}, \mathbf{1})'\mathbf{W}(\mathbf{X}_{ran}, \mathbf{1})$ which is equal to

$$\begin{pmatrix} n_1 & \dots & 0 & n_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & n_p & n_p \\ n_1 & \dots & n_p & n \end{pmatrix}.$$

When computing the trace of matrix

$$\mathbf{H} := \begin{pmatrix} n_1 & \dots & 0 & n_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & n_p & n_p \\ n_1 & \dots & n_p & n \end{pmatrix} \cdot \begin{pmatrix} \frac{cn_1^2+n_1+\lambda_{ran}}{(n_1+\lambda_{ran})^2} & \dots & \frac{cn_1n_p}{(n_1+\lambda_{ran})(n_p+\lambda_{ran})} & -\frac{cn_1}{n_1+\lambda ran} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{cn_1n_p}{(n_1+\lambda_{ran})(n_p+\lambda_{ran})} & \dots & \frac{cn_p^2+n_p+\lambda_{ran}}{(n_p+\lambda_{ran})^2} & -\frac{cn_p}{n_p+\lambda ran} \\ -\frac{cn_1}{n_1+\lambda ran} & \dots & -\frac{cn_p}{n_p+\lambda ran} & c \end{pmatrix}$$

we only need to calculate its diagonal elements. The diagonal elements are given by

$$h_{kk} = n_k \frac{cn_k^2 + n_k + \lambda_{ran}}{(n_k + \lambda_{ran})^2} - \frac{cn_k^2}{n_k + \lambda_{ran}} = \frac{n_k^2 + \lambda_{ran}n_k - cn_k^2\lambda_{ran}}{(n_k + \lambda_{ran})^2}$$

for $k = 1, \dots, p$, whereas the last element is given by

$$h_{p+1,p+1} = -\frac{cn_1^2}{n_1 + \lambda_{ran}} - \dots - \frac{cn_p^2}{n_p + \lambda_{ran}} + nc.$$

Hence $\text{tr}(\mathbf{H})$ can be obtained by

$$\text{tr}(\mathbf{H}) = \sum_{k=1}^{p} \frac{-cn_k^3 + n_k^2 - 2cn_k^2\lambda_{ran} + n_k\lambda_{ran}}{(n_k + \lambda_{ran})^2} + nc$$

with $c = (n - \sum_{k=1}^{p} n_k^2/(n_k + \lambda_{ran}))^{-1}$ leading to formula (3.25).

B.2 Degrees of freedom for spatial functions

For the simple predictor

$$\eta = \gamma_0 + f_{spat}(s) = \gamma_0 + f_{str}(s) + f_{unstr}(s),$$

where the spatial function is divided into a smooth function represented by a Markov random field and an unstructured effect modelled through an i.i.d Gaussian random effect, the true degrees of freedom of the spatial function can be calculated by formula (3.26) using the overall hat matrix, i.e.

$$\begin{aligned} df_{spat} &= df_{str} + df_{unstr} = \operatorname{tr}(\mathbf{H}) - 1 \\ &= \operatorname{tr}\left\{(\mathbf{X},\mathbf{X})\left[(\mathbf{X},\mathbf{X})'\mathbf{W}(\mathbf{X},\mathbf{X}) + \mathbf{P}_{total}\right]^{-1}(\mathbf{X},\mathbf{X})'\mathbf{W}\right\} - 1 \\ &= \operatorname{tr}\left\{(\mathbf{X},\mathbf{X})'\mathbf{W}(\mathbf{X},\mathbf{X})\left[(\mathbf{X},\mathbf{X})'\mathbf{W}(\mathbf{X},\mathbf{X}) + \mathbf{P}_{total}\right]^{-1}\right\} - 1, \end{aligned}$$

with the blockdiagonal penalty matrix $\mathbf{P}_{total} = \operatorname{diag}(\lambda_{unstr}\mathbf{I}_p, \lambda_{str}\mathbf{P}_{str})$. The design matrix $\mathbf{X}$ is identical for both Markov random field and i.i.d. Gaussian random effect.
Similar to the last section, we have to calculate the inverse matrix

$$\begin{pmatrix} \mathcal{X} & \mathcal{Y} \\ \mathcal{Y} & \mathcal{Z} \end{pmatrix} := \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B} & \mathcal{C} \end{pmatrix}^{-1} := \begin{pmatrix} \mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_{unstr}\mathbf{I} & \mathbf{X}'\mathbf{W}\mathbf{X} \\ \mathbf{X}'\mathbf{W}\mathbf{X} & \mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_{str}\mathbf{P}_{str} \end{pmatrix}^{-1}$$

first. This can be done by using formula (B.1) where $\mathcal{B}' = \mathcal{B}$. Again, we start with calculating the least complex submatrix $\mathcal{Z}$ which is contained in the other submatrices:

$$\begin{aligned} \mathcal{Z} &= \left(\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_{str}\mathbf{P}_{str} - \mathbf{X}'\mathbf{W}\mathbf{X}(\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_{unstr}\mathbf{I})^{-1}\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1} \\ &= \left(\lambda_{str}\mathbf{P}_{str} + \operatorname{diag}(n_k) - \operatorname{diag}(n_k)\operatorname{diag}\left(\frac{1}{n_k + \lambda_{unstr}}\right)\operatorname{diag}(n_k)\right)^{-1} \\ &= \left(\operatorname{diag}\left(\frac{n_k\lambda_{unstr}}{n_k + \lambda_{unstr}}\right) + \lambda_{str}\mathbf{P}_{str}\right)^{-1}. \end{aligned}$$

Here we used that $n_k = \sum_{1\le i\le n: x_{ik}=1} w_i$ for $k = 1,\ldots,p$ and that $\mathbf{X}'\mathbf{W}\mathbf{X} = \operatorname{diag}(n_k)$ since $\mathbf{X}$ is a 0/1–incidence matrix. For the other submatrices we get

$$\begin{aligned} \mathcal{Y} &= -\mathcal{Z}\cdot\mathbf{X}'\mathbf{W}\mathbf{X}(\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_{unstr}\mathbf{I})^{-1} \\ &= -\mathcal{Z}\cdot\operatorname{diag}\left(\frac{n_k}{n_k + \lambda_{unstr}}\right) \end{aligned}$$

and

$$\begin{aligned} \mathcal{X} &= (\mathbf{X}'\mathbf{W}\mathbf{X} + \lambda_{unstr}\mathbf{I})^{-1}[\mathbf{I} - \mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Y}] \\ &= \operatorname{diag}\left(\frac{1}{n_k + \lambda_{unstr}}\right)\left[\mathbf{I} + \operatorname{diag}(n_k)\cdot\mathcal{Z}\cdot\operatorname{diag}\left(\frac{n_k}{n_k + \lambda_{unstr}}\right)\right]. \end{aligned}$$

The trace of the overall hat matrix $\mathbf{H}$ for both spatial functions can now be simplified to

$$\begin{aligned}\operatorname{tr}(\mathbf{H}) &= \operatorname{tr}\begin{pmatrix} \mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{X}+\mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Y} & \mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Y}+\mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Z} \\ \mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{X}+\mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Y} & \mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Y}+\mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Z} \end{pmatrix} \\ &= \operatorname{tr}(\mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{X})+\operatorname{tr}(\mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Y})+\operatorname{tr}(\mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Y})+\operatorname{tr}(\mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Z}) \\ &= \quad (3.27)+1,\end{aligned}$$

where

$$\begin{aligned}\operatorname{tr}(\mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Z}) &= \operatorname{tr}\left[\operatorname{diag}(n_k)\left(\operatorname{diag}\left(\frac{n_k\lambda_{unstr}}{n_k+\lambda_{unstr}}\right)+\lambda_{str}\mathbf{P}_{str}\right)^{-1}\right], \\ \operatorname{tr}(\mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{Y}) &= \operatorname{tr}\left[-\operatorname{diag}(n_k)\cdot\mathcal{Z}\cdot\operatorname{diag}\left(\frac{n_k}{n_k+\lambda_{unstr}}\right)\right] \\ &= -\operatorname{tr}\left[\operatorname{diag}\left(\frac{n_k^2}{n_k+\lambda_{unstr}}\right)\left(\operatorname{diag}\left(\frac{n_k\lambda_{unstr}}{n_k+\lambda_{unstr}}\right)+\lambda_{str}\mathbf{P}_{str}\right)^{-1}\right]\end{aligned}$$

and

$$\begin{aligned}&\operatorname{tr}(\mathbf{X}'\mathbf{W}\mathbf{X}\cdot\mathcal{X}) \\ =\ &\operatorname{tr}\left[\operatorname{diag}(n_k)\operatorname{diag}\left(\frac{1}{n_k+\lambda_{unstr}}\right)\left(\mathbf{I}+\operatorname{diag}(n_k)\cdot\mathcal{Z}\cdot\operatorname{diag}\left(\frac{n_k}{n_k+\lambda_{unstr}}\right)\right)\right] \\ =\ &\operatorname{tr}\left[\operatorname{diag}\left(\frac{n_k}{n_k+\lambda_{unstr}}\right)+\operatorname{diag}\left(\frac{n_k^2}{n_k+\lambda_{unstr}}\right)\cdot\mathcal{Z}\cdot\operatorname{diag}\left(\frac{n_k}{n_k+\lambda_{unstr}}\right)\right] \\ =\ &\sum_{k=1}^{p}\frac{n_k}{n_k+\lambda_{unstr}}+\operatorname{tr}\left[\operatorname{diag}\left(\frac{n_k^3}{(n_k+\lambda_{unstr})^2}\right)\left[\operatorname{diag}\left(\frac{n_k\lambda_{unstr}}{n_k+\lambda_{unstr}}\right)+\lambda_{str}\mathbf{P}_{str}\right]^{-1}\right].\end{aligned}$$

Hence, with these formulas $\operatorname{tr}(\mathbf{H})$ can be calculated based on the individual design and penalty matrices so that the sparse structures of these matrices can be fully utilised. The overall hat matrix $\mathbf{H}$ is not needed.

B.3 Degrees of freedom for a seasonal component

Here, we consider the simple predictor

$$\eta = \gamma_0 + f_{season}(t),$$

containing an intercept term and a seasonal effect with p seasons. The true degrees of freedom for this simple predictor can be calculated from the overall hat matrix using formula (3.28)

$$df_s = \operatorname{tr}\left\{(\mathbf{1},\mathbf{X}_s)\left[(\mathbf{1},\mathbf{X}_s)'\mathbf{W}(\mathbf{1},\mathbf{X}_s)+\lambda_s\ \operatorname{diag}(0,\mathbf{P}_{per})\right]^{-1}(\mathbf{1},\mathbf{X}_s)'\mathbf{W}\right\}-1$$

$$= \operatorname{tr}\left\{(\mathbf{1},\mathbf{X}_s)'\mathbf{W}(\mathbf{1},\mathbf{X}_s)\left[(\mathbf{1},\mathbf{X}_s)'\mathbf{W}(\mathbf{1},\mathbf{X}_s)+\lambda_s\ \operatorname{diag}(0,\mathbf{P}_{per})\right]^{-1}\right\}-1.$$

In this section we show the derivation of formula (3.29).
First, we have to calculate the inverse matrix (compare formula (B.1))

$$\begin{pmatrix}\mathcal{X} & \mathcal{Y}'\\ \mathcal{Y} & \mathcal{Z}\end{pmatrix} := \begin{pmatrix}\mathcal{A} & \mathcal{B}\\ \mathcal{B}' & \mathcal{C}\end{pmatrix}^{-1} := \begin{pmatrix}\mathbf{1}'\mathbf{W}\mathbf{1} & \mathbf{1}'\mathbf{W}\mathbf{X}_s\\ \mathbf{X}'_s\mathbf{W}\mathbf{1} & \mathbf{X}'_s\mathbf{W}\mathbf{X}_s+\lambda_s\mathbf{P}_{per}\end{pmatrix}^{-1}.$$

Here, the most complex matrix is matrix $\mathcal{Z}$ which is given by

$$\begin{aligned}\mathcal{Z} &= (\mathbf{X}'_s\mathbf{W}\mathbf{X}_s+\lambda_s\mathbf{P}_{per}-\mathbf{X}'_s\mathbf{W}\mathbf{1}(\mathbf{1}'\mathbf{W}\mathbf{1})^{-1}\mathbf{1}'\mathbf{W}\mathbf{X}_s)^{-1}\\ &= \left(\operatorname{diag}(n_1,\ldots,n_p)-\frac{1}{n}(n_1,\ldots,n_p)'(n_1,\ldots,n_p)+\lambda_s\mathbf{P}_{per}\right)^{-1}\\ &= \left[\begin{pmatrix} n_1-n_1^2/n & -n_1n_2/n & \ldots & n_1n_p/n\\ -n_1n_2/n & \ddots & & \vdots\\ \vdots & & \ddots & n_{p-1}n_p/n\\ -n_1n_p/n & \ldots & n_{p-1}n_p/n & n_p-n_p^2/n\end{pmatrix}+\lambda_s\mathbf{P}_{per}\right]^{-1}\end{aligned}$$

with $n_k=\sum_{1\le i\le n:x_{ik}=1} w_i$ for $k=1,\ldots,p$ and $\mathbf{X}'_s\mathbf{W}\mathbf{X}_s=\operatorname{diag}(n_k)$ since $\mathbf{X}_s$ is a 0/1–incidence matrix. The computation of matrix $\mathcal{Z}$ requires the inversion of a symmetric $p\times p$ matrix which has no sparse structure. However, later we will need all elements of $\mathcal{Z}$ for the degrees of freedom.
Matrix $\mathcal{Y}$ is obtained as

$$\mathcal{Y}=-\mathcal{Z}\cdot\mathbf{X}'_s\mathbf{W}\mathbf{1}(\mathbf{1}'\mathbf{W}\mathbf{1})^{-1}=-\frac{1}{n}\mathcal{Z}\cdot(n_1,\ldots,n_p)'=-\frac{1}{n}\left(\sum_k z_{1k}n_k,\ldots,\sum_k z_{pk}n_k\right)'$$

and matrix $\mathcal{X}$ is given by

$$\mathcal{X}=(\mathbf{1}'\mathbf{W}\mathbf{1})^{-1}[1-\mathbf{1}'\mathbf{W}\mathbf{X}_s\cdot\mathcal{Y}]=\frac{1}{n}\left(1-(n_1,\ldots,n_p)\cdot\mathcal{Y}\right)=\frac{1}{n}\left(1+\frac{1}{n}\sum_{j,k}z_{jk}n_jn_k\right).$$

Based on these matrices the overall hat matrix is obtained as

$$\mathbf{H}=\begin{pmatrix} n & n_1 & \ldots & n_p\\ n_1 & n_1 & & \\ \vdots & & \ddots & \\ n_p & & & n_p\end{pmatrix}\cdot\begin{pmatrix} \frac{1}{n}(1+\frac{1}{n}\sum_{j,k}z_{jk}n_jn_k) & -\frac{1}{n}\sum_k z_{1k}n_k & \ldots & -\frac{1}{n}\sum_k z_{pk}n_k\\ -\frac{1}{n}\sum_k z_{1k}n_k & & & \\ \vdots & & \mathcal{Z} & \\ \frac{1}{n}\sum_k z_{pk}n_k & & & \end{pmatrix}.$$

For computing the trace $\operatorname{tr}(\mathbf{H})$ we need only the diagonal elements of $\mathbf{H}$ and get

$$\begin{aligned}\operatorname{tr}(\mathbf{H}) &= \frac{n}{n}\left(1+\frac{1}{n}\sum_{j,k}z_{jk}n_jn_k\right)-\frac{1}{n}\sum_{j,k}z_{jk}n_jn_k-\sum_j\left(\frac{n_j}{n}\sum_k z_{jk}n_k-n_jz_{jj}\right)\\ &= 1+\sum_k n_kz_{kk}-\frac{1}{n}\sum_k n_k^2z_{kk}-\frac{2}{n}\sum_{j>k}n_kn_jz_{jk}=(3.29)+1.\end{aligned}$$

References

Abe, M. (1999). A generalized additive model for discrete–choice data. *Journal of Business & Economic Statistics 17*(3), 271–284.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & F. Csaki (Eds.), *2nd International Symposium on Information Theory*, pp. 267–281. Akademia Kiao, Budapest.

Augustin, N., Sauerbrei, W., and Schumacher, M. (2005). The practical utility of incorporating model selection uncertainty into prognostic models for survival data. *Statistical Modelling 5*(2), 95–118.

Belitz, C., Hübner, J., Klasen, S., and Lang, S. (2007). Determinants of sex–specific undernutrition in India: A geoadditive semi–parametric regression approach. *Technical Report, University of Munich*.

Belitz, C. and Lang, S. (2007). Simultaneous selection of variables and smoothing parsmeters in geoadditive regression models. In H.-J. Lenz & R. Decker (Eds.), *Advanced Data Analysis*, pp. 189–196. Springer, Berlin–Heidelberg.

Besag, J., York, J., and Mollie, A. (1991). Bayesian image restoration with two applications in spatial statistics (with discussion). *Annals of the Institute of Statistical Mathematics 43*(1), 1–59.

Biller, C. (2000). Adaptive bayesian regression splines in semiparametric generalized linear models. *Journal of Computational and Graphical Statistics* **9**, 122–140.

Bollaerts, K., Eilers, P. H., and Van Mechelen, I. (2006). Simple and Multiple P-splines regression with Shape Constraints. *British Journal of Mathematical and Statistical Psychology 59*(2), 451–469.

Brezger, A. (2004). *Bayesian P–Splines in Structured Additive Regression*. PhD thesis, Dr.Hut–Verlag.

Brezger, A., Kneib, T., and Lang, S. (2005a). Bayesx: Analysing bayesian structured additive regression models. *Journal of Statistical Software 14*(11).

Brezger, A., Kneib, T., and Lang, S. (2005b). *BayesX Manuels*. Available under *http://www.stat.uni-muenchen.de/∼bayesx*.

Brezger, A. and Lang, S. (2006). Generalized structured additive regression based on

Bayesian P-splines. *Computational Statistics and Data Analysis* **50**, 967–991.

Brezger, A. and Steiner, W. (2006). Monotonic Regression based on Bayesian P-Splines: An application to estimating price response functions from store-level scanner data. *Journal of Business and Economic Statistics*, to appear.

Bühlmann, P. (2004). Boosting for high–dimensional linear models. *Technical Report, ETH Zürich*.

Bühlmann, P. and Yu, B. (2003). Boosting with the L_2 loss: regression and classification. *Journal of the American Statistical Association* **98**, 324–339.

Buja, A., Hastie, T., and Tibshirani, R. (1989). Linear smoothers and additive models (with discussion). *The Annals of Statistics 17*(2), 453–510.

Burnham, K. P. and Anderson, D. R. (1998). *Model Selection and Multimodel Inference*. Springer.

Carpenter, J. and Bithell, J. (2000). Bootstrap confidence intervals: when, which, what? a practical guide for medical statisticians. *Statistics in Medicine* **19**, 1141–1164.

Cavanaugh, J. E. (1997). Unifying the Derivations for the Akaike and Corrected Akaike Information Criteria. *Statistics and Probability Letters* **33**, 201–208.

Cavanaugh, J. E. and Neath, A. A. (1999). Generalizing the Derivation of the Schwarz Information Criterion. *Communication in Statistics – Theory and Methods* **28**, 49–66.

Chambers, J. M. and Hastie, T. J. (1992). *Statistical Models in S*. Wadsworth and Brooks.

Chen, Z. (1993). Fitting multivariate regression functions by interaction spline models. *Journal of the Royal Statistical Society B 55*(2), 473–491.

Clyde, M. and George, Edward, I. (2004). Model Uncertainty. *Statistical Science* **19**, 81–94.

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and their Application*. Cambridge University Press.

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

Denuit, M. and Lang, S. (2004). Nonlife Ratemaking with Bayesian GAMs. *Insurance: Mathematics and Economics* **35**, 627–647.

Dierckx, P. (1995). *Curve and Surface Fitting with Splines*. Oxford: Clarendon Press.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics 32*(2), 407–499.

Efron, B. and Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman and Hall.

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

Eilers, P. H. and Marx, B. D. (2003). Multivariate calibration with temperature interaction using two–dimensional penalized signal regression. *Chemometrics and Intelligent*

Laboratory Systems **66**, 159–174.

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

Fahrmeir, L. and Lang, S. (2001a). Bayesian inference for generalized addititve mixed models based on Markov random field priors. *Journal of the Royal Statistical Society C* **50**, 201–220.

Fahrmeir, L. and Lang, S. (2001b). Bayesian semiparametric regression analysis in multicategorical time–space data. *Annals of the Institute of Statistical Mathematics* **53**, 11–30.

Fahrmeir, L. and Tutz, G. (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer.

Fan, J. and Gijbels, I. (1984). *Local Polynomial Modelling and Its Applications*. Chapman and Hall, London.

Fotheringham, A., Brunsdon, C., and Charlton, M. (2002). *Geographically Weighted Regression: The Analysis of Spatially Varying Relationships*. Wiley, Chichester.

Friedman, J. H. (1991). Multivariate adaptive regression splines (with discussion). *Annals of Statistics* **19**, 1–141.

Gamerman, D. (1997). Efficient Sampling from the Posterior Distribution in Generalized Linear Models. *Statistics and Computing* **7**, 57–68.

George, A. and Liu, J. W. (1981). *Computer Solution of Large Sparse Positive Definite Sytems*. Prentice–Hall.

George, E. I. and McCulloch, R. E. (1997). Approaches for bayesian variable selection. *Statistica Sinica* **7**, 339–373.

Geweke, J. (1996). Variable selection and model comparison in regression. In J. M. Bernardo, J. O. Berger, A. P. Dawid, & A. F. M. Smith (Eds.), *Bayesian Statistics 5*, pp. 609–620. Oxford Press.

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

Green, P. J. (2001). A primer on markov chain monte carlo. In O. E. Barndorff-Nielsen, D. R. Cox, & C. Klüppelberg (Eds.), *Complex Stochastic Systems*, pp. 1–62. Chapman and Hall.

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

Gu, C. (2002). *Smoothing Spline ANOVA models*. Springer, New York.

Gu, C. and Wahba, G. (1991). Minimizing GCV/GML scores with multiple smoothing parameters via the newton method. *SIAM Journal of Scientifical Statistical Computation 12*(2), 383–398.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models*. Chapman and

Hall.

Hastie, T. J. and Tibshirani, R. J. (1993). Varying–coefficient Models. *Journal of the Royal Statistical Society B* **55**, 757–796.

Hastie, T. J. and Tibshirani, R. J. (2000). Bayesian backfitting. *Statistical Science* **15**, 193–223.

Hastie, T. J., Tibshirani, R. J., and Buja, A. (1994). Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association 89*(428), 1255–1270.

Hastie, T. J., Tibshirani, R. J., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer.

Hennerfeind, A., Brezger, A., and Fahrmeir, L. (2006). Geoadditive Survival Models. *Journal of the American Statistical Association* **101**, 1065–1075.

Hoeting, J., Madigan, D., Raftery, A., and Volinsky, C. (1999). Bayesian Model Averaging: A Tutorial. *Statistical Science* **14**, 382–401.

Hübner, J. (2003). Statistische Analyse der Einflussfaktoren auf die Unterernährung von Kindern in Indien - Anwendung eines bayesianischen, geo–additiven, semiparametrischen Regressionsmodells. *Diploma thesis, Technical University of Munich*.

Hurvich, C. M., Simonoff, J. S., and Tsai, C.-L. (1998). Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of the Royal Statistical Society B* **60**, 271–293.

Kadane, J. B. and Lazar, N. A. (2004). Methods and Criteria for Model Selection. *Journal of the American Statistical Association* **99**, 279–290.

Kammann, E. and Wand, M. (2003). Geoadditive models. *Journal of the Royal Statistical Society C* **52**, 1–18.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association 90*(430), 773–795.

Kauermann, G., Claeskens, G., and Opsomer, J. D. (2006). Bootstrapping for Penalized Spline Rregression. *Research Report KBI_0609, Faculty of Economics and Applied Economics, Katholieke Universiteit Leuven*.

Kim, Y.-J. and Gu, C. (2004). Smoothing spline Gaussian regression: more scalable computation via efficient approximation. *Journal of the Royal Statistical Society B 6*(2), 337–356.

Kneib, T. (2006). *Mixed model based inference in structured additive regression*. PhD thesis, Dr.Hut–Verlag.

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

Krause, R. and Tutz, G. (2004). Simultaneous Selection of Variables and Smoothing Parameters in Additive Models. In D. Baier & K.-D. Wernecke (Eds.), *Innovations in Classification, Data Science, and Information Systems*, pp. 146–153. Springer,

Berlin–Heidelberg.

Krause, R. and Tutz, G. (2006). Genetic Algorithms for the Selection of Smoothing Parameters in Additive Models. *Computational Statistics* **21**, 8–31.

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

Loader, C. (1999). *Local Regression and Likelihood*. Springer, New York.

Madigan, D. and Raftery, A. E. (1994). Model selection and accounting for model selection uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association* **89**, 1535–1546.

Madigan, D. and York, J. (1995). Bayesian graphical models for discrete data. *International Statistical Review* **63**, 215–232.

Magnus, J. R. and Neudecker, H. (1991). *Matrix differential calculus with applications in statistics and econometrics*. Wiley.

Marx, B. D. and Eilers, P. (1998). Direct generalized additive modeling with penalized likelihood. *Computational Statistics and Data Analysis 28*(2), 193–209.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. Chapman and Hall.

Miller, A. (2002). *Subset Selection in Regression*. Chapman and Hall.

Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian Variable Selection in Linear Regression. *Journal of the American Statistical Assiciation* **83**, 1023–1032.

Raftery, A. E., Madigan, D., and Hoeting, Jennifer, A. (1997). Bayesian Model Averaging for Linear Regression Models. *Journal of the American Statistical Association* **92**, 179–191.

Ramsey, J. (1988). Monotone regression splines in action. *Statistical Science* **3**, 425–461.

Rigby, R. A. and Stasinopoulos, D. M. (2005). Generalized Additive Models for Location, Scale and Shape. *Applied Statistics* **54**, 507–554.

Rue, H. (2001). Fast sampling of Gaussian Markov random fields with application. *Journal of the Royal Statistical Society B* **63**, 325–338.

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*, Volume 104 of *Monographs on Statistics and Applied Probability*. London: Chapman & Hall.

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

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics* **6**, 461–464.

Shao, J. and Dongsheng, T. (1995). *The Jackknife and Bootstrap*. New York: Springer.

Shibata, R. (1989). Statistical aspects of model selection. In J. C. Willems (Ed.), *From Data to Model*, pp. 215–240. Springer–Verlag.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van der Linde, A. (2002). Bayesian

measures of model complexity and fits (with discussion). *Journal of the Royal Statistical Society B* **64**, 583–639.

Steiner, W. J., Belitz, C., and Lang, S. (2006). Semiparametric stepwise regression to estimate sales promotion effects. In M. Spiliopoulou, R. Kruse, A. Nürnberger, C. Borgelt, & W. Gaul (Eds.), *From Data and Information Analysis to Knowledge Engineering*, pp. 590–597. Springer, Berlin–Heidelberg.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B* **58**, 267–288.

Tutz, G. and Binder, H. (2006). Generalized additive modelling with implicit variable selection by likelihood–based boosting. *Biometrics* **62**, 961–971.

Tutz, G. and Leitenstorfer, F. (2006). Generalized smooth monotonic regression in additive modeling. *Journal of Computational and Graphical Statistics* **16**, 165–188.

Wahba, G. (1990). Spline models for observational data. *CBMS-NSF Regional Conference Series in Applied Mathematics* **59**.

Wang, Y. and Wahba, G. (1995). Bootstrap confidence intervals for smoothing splines and their comparison to bayesian confidence intervals. *Journal of Statistical Computation and Simulation* **51**, 263–279.

Wood, S. N. (2000). Modelling and smoothing parameter estimation with multiple quadratic penalties. *Journal of the Royal Statistical Society B 62*(1), 413–428.

Wood, S. N. (2003). Thin-plate regression splines. *Journal of the Royal Statistical Society B 65*(1), 95–114.

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

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

Wood, S. N. (2006b). The mgcv Package, version 1.3–22. *R–Manual*.

Wood, S. N. (2006c). On confidence intervals for GAMs based on penalized regression splines. *Australian and New Zealand Journal of Statistics 48*(4), 445–464.

Yau, P., Kohn, R., and Wood, S. (2003). Bayesian variable selection and model averaging in high dimensional multinomial nonparametric regression. *Journal of Computational and Graphical Statistics 12*(1), 23–54.

Lebenslauf

Christiane Belitz

geboren am 15.6.1980 in München

Schulbildung:

1986–1990	Salier–Grundschule in Waiblingen
1990–1999	Salier–Gymnasium in Waiblingen
Juni 1999	Abitur

Studium:

1999–2004	Studium der Statistik an der Ludwig–Maximilians–Universität München mit dem Anwendungsgebiet Biologie und der speziellen Ausrichtung mathematische Statistik
September 2001	Vordiplom
Mai 2004	Diplom

Beruf:

Juni 2001 – Mai 2004	studentische Hilfskraft am Lehrstuhl von Prof. Fahrmeir, Institut für Statistik der LMU München
Juni 2004 – Sept. 2004	wissenschaftliche Hilfskraft am Lehrstuhl von Prof. Fahrmeir, Institut für Statistik der LMU München
Okt. 2004 – Feb. 2005	wissenschaftliche Mitarbeiterin bei Prof. Czado, Zentrum Mathematik der Technischen Universität München
seit März 2005	wissenschaftliche Mitarbeiterin am Lehrstuhl von Prof. Fahrmeir, Institut für Statistik der LMU München