Bayesian Regularization and Model Choice in Structured Additive Regression

Fabian Scheipl

München 2011

Bayesian Regularization and Model Choice in Structured Additive Regression

Dissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
am Institut für Statistik
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Vorgelegt von
Fabian Scheipl
am 31. Januar 2011
in München

Erstgutachter:	Prof. Ludwig Fahrmeir, Ludwig-Maximilians-Universität München
Zweitgutachter:	PD Dr. Christian Heumann, Ludwig-Maximilians-Universität München
Externer Gutachter:	Prof. Thomas Kneib, Carl von Ossietzky Universität Oldenburg
Rigorosum:	17. März 2011

Abstract

In regression models with a large number of potential model terms, the selection of an appropriate subset of covariates and their interactions is an important challenge for data analysis, as is the choice of the appropriate representation of their impact on the quantities to be estimated such as deciding between linear or smooth non-linear effects. The main part of this work is dedicated to the development, implementation and validation of an extension of stochastic search variable selection (SSVS) for structured additive regression models aimed at finding and estimating appropriate and parsimonious model representations. The approach described here is the first implementation of fully Bayesian variable selection and model choice for general responses from the exponential family in generalized additive mixed models (GAMM) available in free and open source software. It is based on a spike-and-slab prior on the regression coefficients with an innovative multiplicative parameter expansion that induces desirable shrinkage properties. This thesis points out a possible reason why previous attempts at extending SSVS algorithms for the selection of parameter vectors have not been entirely successful, discusses the regularization properties of the novel prior structure, investigates sensitivity of observed results with regard to the choice of hyperparameters and compares the performance on real and simulated data in a variety of scenarios to that of established methods such as boosting, conventional generalized additive mixed models and LASSO estimation. Some case studies show the usefulness as well as the limitations of the approach.

The second part of this work presents a method for locally adaptive function estimation for functions with spatially varying roughness properties. An implementation of locally adaptive penalized spline smoothing using a class of heavy-tailed shrinkage priors for the estimation of functional forms with highly varying curvature or discontinuities is presented. These priors utilize scale mixtures of normals with locally varying exponential-gamma distributed variances for the differences of the P-spline coefficients. A fully Bayesian hierarchical structure is derived with inference about the posterior being based on Markov Chain Monte Carlo techniques. Three increasingly flexible and automatic approaches are introduced to estimate the spatially varying structure of the variances. Extensive simulation studies for Gaussian, Poisson, and Binomial responses shows that the performance of this approach on a number of benchmark functions is competitive to that of previous approaches. Results from two applications support the conclusions of the simulation studies.

Zusammenfassung

In Regressionsmodellen mit einer großen Zahl von potentiellen Modelltermen ist die Auswahl einer angemessenen Teilmenge an Kovariablen sowie ihrer Interaktionen eine wichtige Herausforderung der angewandten Statistik. Zusätzlich muss zwischen linearen und glatten funktionalen Formen der Effekte unterschieden werden. Der Hauptteil dieser Arbeit befasst sich mit der Entwicklung, Implementierung und Validierung einer Erweiterung des Stochastic Search Variable Selection-Ansatzes (SSVS) um in strukturierten additiven Regressionsmodellen geeignete parametersparsame Modelle auszuwählen und zu schätzen. Die entwickelten Methoden sind der erste in frei verfügbarer Software implementierte Ansatz der voll-Bayesianische Variablenselektion und Modellwahl für Zielvariablen aus Exponentialfamilien in generalisierten additiven gemischten Modellen erlaubt. Er basiert auf einer Spike-and-Slab Priori mit einer innovativen multiplikativen Parameterexpansion, die besonders günstige Regularisierungseigenschaften besitzt. Die vorliegende Arbeit diskutiert mögliche Ursachen, warum bisherige Versuche SSVS-Algorithmen auf die Auswahl von Parameterblöcken auszudehnen wenig erfolgreich waren, leitet die Regularisierungseigenschaften der eingeführten Prioristruktur her, untersucht die Sensitivität der erzielten Ergebnisse im Bezug auf die Wahl von Hyperparametern und vergleicht die erzielten Ergebnisse auf echten und simulierten Daten mit den Ergebnissen anderer Methoden wie Boosting, LASSO oder konventionellen generalisierten additiven gemischten Modellen. Fallstudien zeigen das Anwendungspotenzial und die Leistungsgrenzen des eingeführten Ansatzes.

Der zweite Teil der Arbeit befasst sich mit einer Methode zur lokal adaptiven Funktionsschätzung bei Funktionen, deren Rauheit sich über ihren Wertebereich verändert. Die beschriebene Implementation benutzt Regularisierungsprioris mit dicken Schwänzen zur Schätzung funktionaler Formen mit stark variierender Krümmung oder Unstetigkeitsstellen. Diese Prioris nutzen Skalenmischungen der Normalverteilung mit lokal variierenden Exponential-Gamma verteilten Varianzen für die Differenzen der Splinekoeffizienten. Es werden drei zunehmend flexible und automatische Ansätze zur Schätzung der lokal variierenden Struktur der Varianzen beschrieben. Breit angelegte Simulationsstudien für Normal-, Poisson- und binomialverteilte Zielvariablen zeigen, dass die Leistung des beschriebenen Ansatzes konkurrenzfähig zu früheren, in der Literatur beschriebenen Ansätzen ist. Ergebnisse in Anwendungen mit Normal- und Poissonverteilten Zielvariablen untermauern die Ergebnisse der Simulationsstudien.

Acknowledgements

To Ludwig Fahrmeir and Thomas Kneib, for giving me this opportunity and pointing me in suitable directions.

To Helmut Küchenhoff, for getting me on track at the very beginning.

To Sonja Greven, for opening a door and prompting me to finally get it done.

To my parents, for their unflinching support.

To Nadja, for hugs, kisses and giggles.

Contents

Introduction

> We are drowning in information, but we are starved for knowledge.
>
> *John Naisbitt*

One of the largest challenges of modern applied statistical modeling arises from the confluence of the following two factors:

1. Methodological advances in applied statistics of the last decades and ever more powerful computers make the estimation of increasingly flexible and sophisticated models that include nonlinear, temporal, spatial or tempo-spatial effects and dependency structures feasible and accessible for practitioners.

2. Modern methods of data acquisition provide researchers in many fields of science with ever increasing amounts of data. These data sets often have as many features as observations or even more features than observations.

Combined, these two trends lead to regression models that are heavily overparameterized, both because of the large number of features themselves and because of the increase in parameters due to semiparametric terms. This in turn calls for both reliable regularization of the resulting inverse problems and principled and general methods of model choice and model simplification in order to be able to make sense of the estimated structures in appropriately complex, yet parsimonious model representations.

The first part of this dissertation describes the theoretical development and validates the implementation of a first step towards such a method in a fully Bayesian framework. We describe a novel generalization of stochastic search variable selection in the context of structured additive regression leading towards a class of Bayesian prior structures that offers good regularization properties and simultaneously accomplishes selection of model terms for complex regression models with additive predictors. The approach we describe is aimed not only at identifying relevant covariates and interactions and removing those with negligible effects, it can additionally distinguish between linear and nonlinear effects and interactions in order to fit models that are as parsimonious and easy to interpret as possible. Our method is implemented for Gaussian, binomial and Poisson distributed responses in an open source software package `spikeSlabGAM` for the R environment.

The theoretical motivation behind our approach and the properties of the proposed prior structure are discussed in Chapter 2 of this first part. Chapter 3 then describes the implementation of our ideas in the software package `spikeSlabGAM`, followed by applications on real and simulated data sets in Chapter 4. Our results indicate that this implementation can improve upon previous approaches in terms of predictive performance and function selection in structured additive models for both Gaussian and non-Gaussian responses. Selection of very flexible terms associated with large coefficient blocks such as random effects or Markov random fields, however, is strongly biased towards inclusion for non-Gaussian responses.

While the first part aims at discovering adequate and flexible but parsimonious models, the second part of this dissertation focuses on discovering and accounting for additional complexity: We describe a fully Bayesian approach for locally adaptive estimation of univariate functions with locally varying roughness, that is, functional forms with highly varying curvature or discontinuities.

The main innovation of our approach is to estimate a locally varying smoothing parameter in the shape of a step function, with the option to estimate the locations and number of steps as well as their heights. Additionally, we use a heavy tailed scale mixture of Gaussians with a sharp peak in zero instead of the conventional Gaussian or Student priors for the spline coefficients. This is crucial in order to achieve good performance for functions of both smoothly and abruptly varying roughness.

Chapters 6 and 7 motivate the prior structures we use for the spatially varying structure of the smoothing parameters and discuss their properties. We consider three increasingly flexible and automatic approaches. Chapters 8 and 9 describe results for Gaussian, binomial and Poisson response on artificial and real data sets, respectively.

From a methodological point of view, both parts of this dissertation share a focus on flexible and robust regularization priors that are parameterized as scale mixtures of Gaussians.

This dissertation is based in part on the following publications and working papers:

- F. Scheipl, L. Fahrmeir, T. Kneib (2011). Function Selection in Structured Additive Regression Models based on Spike-and-Slab Priors. *In preparation.* (Chapter 2, Sections 4.2, 4.4)

- S.N. Wood, F. Scheipl*, and J.J. Faraway (2011). On intermediate Rank Tensor Product Smoothing. *Submitted.* ($\star$: minor contribution; Section 3.1.2)

- F. Scheipl (2011). `spikeSlabGAM`: Bayesian Variable Selection, Model Choice and Regularization for Generalized Additive Mixed Models in R. *Submitted.* (Chapter 3)

- F. Scheipl (2010). Normal-Mixture-of-inverse-Gamma Priors for Bayesian Regularization and Model Selection in Generalized Additive Models. Technical Report 84, Department of Statistics, LMU München. (Chapter 4)

- F. Scheipl, T. Kneib (2009): Locally adaptive Bayesian P-splines with a Normal-Exponential-Gamma Prior. *Computational Statistics & Data Analysis*, 53(10):3533-3552. (Chapter 9)

- F. Scheipl, T. Kneib (2008): Locally adaptive Bayesian P-Splines with a Normal-Exponential-Gamma Prior. Technical Report 22, Department of Statistics, LMU München. (Chapters 6-10)

The following software packages were created as part of this work:

- R-package `spikeSlabGAM`

- R-package `negspline`

Part I.

Bayesian Variable Selection and Model Choice for Structured Additive Regression

1. Introduction

In data sets with many potential predictors, choosing an appropriate subset of covariates and their interactions at the same time as determining whether linear or more flexible functional forms are required to model the relationship between covariates and response is a challenging and important task. From a Bayesian perspective, it can be translated into a question of estimating marginal posterior probabilities whether a variable should be in the model and in what form (i.e. linear or smooth; as main effect and/or as effect modifier).

The following Chapter 2 lays out the theoretical background for a new method to select or deselect single coefficients as well as blocks of coefficients associated with factor variables, interactions or basis expansions of smooth functions. It is based on a spike-and-slab prior structure similar to Ishwaran and Rao (2005). We use bimodal priors for the hyper-variances of the regression coefficients that result in a two component mixture of a narrow "spike" around zero and a "slab" with wide support as the marginal prior for the coefficients. The mixture weights for the "spike" component can be interpreted as posterior probabilities of exclusion of a coefficient or coefficient block from the model. This is the basic idea that unites all the different flavors of stochastic search variable selection (SSVS) (George and McCulloch, 1993).

The main contribution of the present work is the extension of the spike-and-slab or SSVS approach for selection of single coefficients in Gaussian models to the selection of potentially large blocks of coefficients for general responses from an exponential family. We use an innovative sampling procedure based on a multiplicative parameter expansion (Gelman, Van Dyk, Huang, and Boscardin, 2008) in order to improve the exceedingly slow mixing of conventional samplers that make a direct extension of the spike-and-slab approach for function selection (or, more generally, selection of coefficient blocks) infeasible. We also show that this parameter expansion leads to a prior with desirable regularization properties similar to $\mathcal{L}_q$-penalization with $q < 1$. The proposed approach is immediately applicable, since it is implemented in publicly available software (R-package `spikeSlabGAM` (Scheipl, 2010d)) and the presented results are reproducible. Our proposal improves on previous approaches in that it fulfills all of the following criteria simultaneously:

i. it accommodates all types of regularized effects with a (conditionally) Gaussian prior such as simple covariates (both metric and categorical), penalized splines (uni- or multivariate), random effects or ridge-penalized factors/interaction effects,

ii. it scales reasonably well to intermediate datasets with thousands of observations and hundreds of covariates,

iii. it accommodates non-Gaussian responses from the exponential family,

iv. it is implemented in publicly available and user-friendly open source software.

Fitting the practical importance of the topic, a vast literature on Bayesian approaches for selection of single coefficients based on mixture priors for the coefficients exists. In a recent review paper, O'Hara and Sillanpää (2009) compare the spike-and-slab approach in Kuo and Mallick (1998), the Gibbs variable selection approach (Carlin and Chib, 1995; Dellaportas, Forster, and Ntzoufras, 2002), and stochastic search variable selection (SSVS) approaches in George and McCulloch (1993), among other methods.

Bayesian function selection, similar to the frequentist COSSO (Lin and Zhang, 2006), is usually based on decomposing an additive model into orthogonal functions in the spirit of a smoothing spline ANOVA (Wahba, Wang, Gu, Klein, and Klein, 1995). Wood, Kohn, Shively, and Jiang (2002) and Yau, Kohn, and Wood (2003) describe implementations using a data-based prior that requires two MCMC runs, a pilot run to obtain a data-based prior for the "slab" part and a second one to estimate parameters and select model components. A more general approach based on double exponential regression models that also allows for flexible modeling of the dispersion is described by Cottet, Kohn, and Nott (2008). They use a reduced rank representation of cubic smoothing splines (i.e a "pseudo-spline" (Hastie, 1996)) with a very small number of basis functions to model the smooth terms in order to reduce the complexity of the fitted models, and, presumably, to avoid the mixing problems detailed in Section 2.2.2. Since the authors were unable to provide their software for this work, it was not possible to compare their approach to the one described in the following. Reich, Storlie, and Bondell (2009) also use the smoothing spline ANOVA framework and perform variable and function selection via SSVS for Gaussian responses, but their implementation is very slow. To the best of our knowledge, none of the above-mentioned approaches was implemented in publicly available software in a useable form at the time of writing and none are able to select between smooth nonlinear and linear effects.

The remainder of this part is structured as follows: Section 2.1 summarizes structured additive regression models and introduces the notation. Section 2.2 describes the prior structure and the parameter expansion trick used to improve mixing and discusses shrinkage properties of the marginal prior for the regression coefficients. Subsequent chapters describe the implementation in more detail (Ch. 3) and summarize results from extensive simulation studies and applications to real data sets (Ch. 4).

2. Normal-Mixture-of-Inverse-Gamma Priors for Bayesian Regularization and Model Selection in Structured Additive Regression Models

This chapter lays out the theoretical background for the methods implemented in the R package `spikeSlabGAM`.

2.1. Structured additive regression

2.1.1. Model structure

Structured additive regression (Fahrmeir, Kneib, and Lang, 2004), a broad model class that contains generalized additive mixed models, is among the most widely used approaches in applied statistics due to its flexibility and generality.

We give a short summary of structured additive regression: The distribution of the responses $\boldsymbol{y}$ given a set of covariates $\boldsymbol{z}$ belongs to an exponential family, i.e

$$p(\boldsymbol{y}|\boldsymbol{z},\phi) = \prod_{i=1}^{n} c(y_i,\phi) \exp\left(\frac{y_i\theta_i - b(\theta_i)}{\phi}\right), \tag{2.1}$$

with $\boldsymbol{\theta}, \phi, b(\cdot)$ and $c(\cdot)$ determined by the type of distribution. The additive predictor $\boldsymbol{\eta}$ determines the conditional expected value of the response via

$$\mathrm{E}(\boldsymbol{y}|\boldsymbol{z}) = h(\boldsymbol{\eta}) \tag{2.2}$$

with a fixed response function $h(\cdot)$. In the class of models we consider in the following, the additive predictor itself is given by

$$\eta_i = \eta_{0i} + \sum_{j=1}^{P} f_j(\boldsymbol{z}_i), \quad i = 1, \dots, n, \tag{2.3}$$

where η_{0i} includes offsets and/or a global intercept term, $\boldsymbol{z}$ represents the complete covariate information and the regression functions f_j are generic representatives for different types of model terms. Important special cases of regression functions in structured additive regression models are

- *Linear effects* $f(\boldsymbol{z}) = \boldsymbol{x}'\boldsymbol{\beta}$ of single covariates or covariate blocks arising for example from dummy coding for categorical covariates or polynomial expansions.

- *Smooth nonlinear effects* $f(\boldsymbol{z}) = f(x)$ of continuous covariates x based on penalized splines (Brezger and Lang, 2006). For function selection, it may be useful to split the effect of covariate x into a component lying in the nullspace of the associated penalty (i.e., a linear part $f_0(x) = x\beta$) and a component for the deviation from linearity $f_{\text{pen}}(x)$ such that we can not only include or exclude the complete effect of x but also linear and nonlinear effects separately. We will show in Section 3.1.2 how such a reparameterization can be achieved for all types of penalized effects with partially improper priors.

- Penalized splines can also be employed in *varying coefficient terms* $f(\boldsymbol{z}) = uf(x),\ \boldsymbol{z} = (u, x)$ where the effect of interaction variable u varies with respect to the effect modifier x, or in *interaction surface estimation* $f(\boldsymbol{z}) = f(x_1, x_2),\ \boldsymbol{z} = (x_1, x_2)$ based on penalized tensor product splines. Again it may be useful to split such terms into linear and nonlinear marginal effects and their respective interaction surfaces.

- *Gaussian Markov random fields* (Rue and Held, 2005) for spatial effects $f(\boldsymbol{z}) = f(s)$ based on (discrete) geographical information s.

- *Random effects* $f(\boldsymbol{z}) = \beta_g$ representing subject- or cluster-specific effects for a grouping factor $z = g$.

- *Surface smoothers* $f(\boldsymbol{z}) = f(\boldsymbol{x})$ for vector valued covariates $\boldsymbol{x}$ based on radial basis functions constructed for example in the context of reproducing kernel Hilbert spaces (Wood, 2006).

Interactions between these basic function types can also be included in structured additive regression models. Flexible terms need to be regularized in order to avoid overfitting and are associated with appropriate shrinkage priors. These shrinkage or regularization priors are usually Gaussian or can be parameterized as scale mixtures of Gaussians (e.g. the Bayesian Lasso with a Laplace prior on the coefficients is a Normal-Exponential scale mixture (Park and Casella, 2008)), so that they are conditionally Gaussian given their variance parameters.

In our approach, model structure (2.3) defines a candidate set of model terms that define a model of maximum complexity and we are interested in finding simple special cases of (2.3) where some of the functions are identified as having negligible impact on the response and therefore drop out of the model.

2.1.2. Model term structure

All of the term types given in the previous section can be represented as $f(\boldsymbol{z}) = \boldsymbol{B}\boldsymbol{\delta}$ with a (possibly partially improper) Gaussian prior $\boldsymbol{\delta} \sim N_K\left(\mathbf{0}, s^2\boldsymbol{P}^-\right)$ with fixed (possibly rank-deficient) precision matrix $\boldsymbol{P}$ and an associated design matrix $\boldsymbol{B}$. The remainder of this section describes this framework in detail for the important special case of penalized splines (P-splines):

Smooth functions $f(\cdot)$ of continuous covariates are commonly modeled via basis function expansions, i.e. $\tilde{f}(\boldsymbol{x}) = \sum_{k=1}^{K} \delta_k \tilde{B}_k(\boldsymbol{x})$ where $\boldsymbol{\delta}$ is a vector of coefficients associated with (nonlinear) basis functions $B_k(\cdot)$; $k = 1, \dots, K$. Many different basis functions and associated regularization approaches exist. Knot-free methods include e.g. thin plate splines (Wood, 2003) or smoothing splines (Wood et al., 2002) and their reduced rank representations (Cottet et al., 2008) based on the dominating eigenvalues and -vectors of the covariance of the equivalent Gaussian process.

In the following, we use Bayesian P-splines as introduced by Lang and Brezger (2004), similar to the approach chosen in Panagiotelis and Smith (2008). In this approach, $B_k(x), k = 1, \dots, K$ is a collection of B-spline basis functions (Eilers and Marx, 1996) and the shrinkage prior on the associated coefficient vector $\boldsymbol{\delta}$ is a Gaussian random walk prior of order d:

$$\Delta^d \boldsymbol{\delta} \sim N_{K-d}\left(\mathbf{0}, \tau^2 \boldsymbol{I}_{K-d}\right),$$

where Δ^d is the d-th difference operator matrix. Unless specified otherwise we use cubic B-splines with a second order difference penalty. Note that this formulation implies a partially improper prior for $\boldsymbol{\delta}$: $p(\boldsymbol{\delta}|\tau^2, \boldsymbol{P}) \propto \exp\left(-0.5\boldsymbol{\delta}'\boldsymbol{P}\boldsymbol{\delta}/\tau^2\right)$, with rank-deficient $\boldsymbol{P} = \Delta^{d\prime}\Delta^d$ that leaves linear functions unpenalized.

2.2. The NMIG model with parameter expansion

The following section describes the prior structure of the conventional Normal-mixture of Inverse Gamma (NMIG) model (Section 2.2.1) and shows that this setup is not well suited for the simultaneous selection of coefficient groups (Section 2.2.2). Section 2.2.3 describes a parameter expansion that changes the prior structure and enables simultaneous selection of coefficient groups. Ishwaran and Rao (2005) originally proposed an empirical Bayes analogue of this prior for selection of single coefficients in the linear model for Gaussian data. Note that this Section glosses over the fact that subvectors of coefficients associated with the different model terms will be associated with the complicated dependence structures given in 2.1.2 and instead assumes marginal independence of coefficients in a given subvector. Section 3.1.2 describes the steps used to achieve this simpler representation.

2.2.1. Model hierarchy

This section discusses the basic model hierarchy for structured additive regression models with the NMIG prior. In most cases, the linear predictor $\boldsymbol{\eta}$ will contain terms that are forced into the model (e.g. a global intercept term) and are not associated with a variable selection prior. We write $\boldsymbol{\eta} = \boldsymbol{\eta}_u + \boldsymbol{X\beta}$, where

$$\boldsymbol{\eta}_u = \boldsymbol{\eta}_0 + \boldsymbol{X}_u\boldsymbol{\beta}_u \tag{2.4}$$

represents the part of the linear predictor not associated with an NMIG prior, consisting of an optional known offset vector $\boldsymbol{\eta}_0$ and the design matrix $\boldsymbol{X}_u$ with associated coefficients $\boldsymbol{\beta}_u$ for the covariates not under selection. In the following, we focus on the part $\boldsymbol{X\beta}$ associated with NMIG priors.

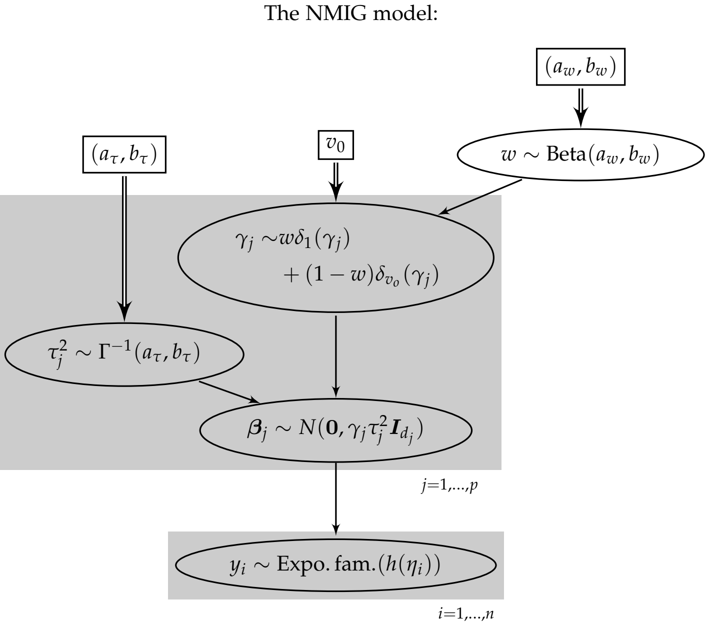

The NMIG model:
(a_w, b_w)
(a_τ, b_τ)
v_0
$w \sim \text{Beta}(a_w, b_w)$
$\gamma_j \sim w\delta_1(\gamma_j) + (1-w)\delta_{v_o}(\gamma_j)$
$\tau_j^2 \sim \Gamma^{-1}(a_\tau, b_\tau)$
$\boldsymbol{\beta}_j \sim N(\boldsymbol{0}, \gamma_j\tau_j^2\boldsymbol{I}_{d_j})$
$j=1,\ldots,p$
$y_i \sim \text{Expo. fam.}(h(\eta_i))$
$i=1,\ldots,n$

Figure 2.1.: Directed acyclic graph of NMIG model. Ellipses are stochastic nodes, rectangles are deterministic/logical nodes. Single arrows are stochastic edges, double arrows are logical/deterministic edges. Subvectors $\boldsymbol{\beta}_j$ are associated with different components of the predictor, e.g. a spline basis or a group of dummy variables coding the different levels of a factor. d_j is the length of subvector $\boldsymbol{\beta}_j$. $h()$ is a known response function. $\delta_y(x)$ is zero for any value of x other than y and 1 at y.

Figure 2.1 shows the hierarchy of the basic NMIG prior model. At the lowest level of the hierarchy, the data y_i, $i = 1, \dots, n$ come from a distribution in the exponential family such as the Gaussian, binomial or Poisson distributions. The canonical parameter of this distribution is connected to the linear predictor via a known response function $h()$. The regression coefficients have independent Gaussian priors with mean zero. Subvectors β_j, $j = 1, \dots, p$ are associated with different components of the predictor, i.e. different covariates, unpenalized and penalized parts of a reparameterized spline basis or a set of indicator variables encoding the levels of a factor. The prior variance for β is constant within subvectors and given by the product of an indicator variable γ_j and the hypervariance τ_j^2. The indicator variable γ_j takes the value 1 with probability w or some (very) small value v_0 with probability $1 - w$. The hypervariance τ_j^2 has an inverse gamma-prior with shape parameter a_τ and scale parameter b_τ with $b_\tau \gg a_\tau$, so that the mode b_τ / a_τ is significantly greater than 1. The implied prior for the effective hypervariance $v_j^2 = \gamma_j \tau_j^2$ is a bimodal mixture of inverse gamma distributions, with one component strongly concentrated on very small values – the *spike* with $\gamma_j = v_0$ and effective scale parameter $v_0 b_\tau$ – and a second more diffuse component with most mass on larger values – the *slab* with $\gamma_j = 1$ and scale b_τ. A coefficient associated with a hypervariance that is primarily sampled from the *spike*-part of the prior will be strongly shrunk towards zero if v_0 is sufficiently small, so that the posterior probability for $\gamma_j = v_0$ can be interpreted as the probability of exclusion of $\boldsymbol{\beta}_j$ from the model. The Beta prior for the mixture weights w can be used to incorporate the analyst's prior knowledge about the sparsity of $\boldsymbol{\beta}$ or, more practically, enforce sufficiently sparse solutions for overparameterized models. In the following, we write $\boldsymbol{\beta}_j \sim \text{NMIG}(v_0, w, a_\tau, b_\tau)$ to denote this prior hierarchy for the regression coefficients.

Expressions for the full conditionals resulting from this prior structure are given in Section 3.2. This prior hierarchy is very well suited for selection of model terms for non-Gaussian data because the selection (i.e. the sampling of indicator variables γ) occurs on the level of the hypervariances for the coefficients. This means that the likelihood itself is not in the Markov blanket of γ and consequently does not occur in the full conditionals for the indicator variables. Since the full conditionals for γ are thus available in closed form regardless of the likelihood, this results in comparatively easy and fast model averaging for non-Gaussian models without the need to delve into the intricacies of estimating marginal likelihoods.

2.2.2. Simultaneous selection of multiple coefficients

Previous approaches for Bayesian variable selection have primarily concentrated on selection of single coefficients (George and McCulloch, 1993; Kuo and Mallick, 1998; Dellaportas et al., 2002; Ishwaran and Rao, 2005) or used very low dimensional bases for the representation of smooth effects. E.g. Cottet et al. (2008) use a pseudo-spline representation of their cubic smoothing spline bases with only 3 to 4 basis functions. In the following, we argue that conventional blockwise Gibbs sampling is ill suited for updating the state of the Markov chain when sampling from the posterior of an NMIG model even for moderately large coefficient blocks. We show that mixing for γ_j will be very slow for blocks of coefficients $\boldsymbol{\beta}_j$ with $d_j \gg 1$. We suppress the index j in the following.

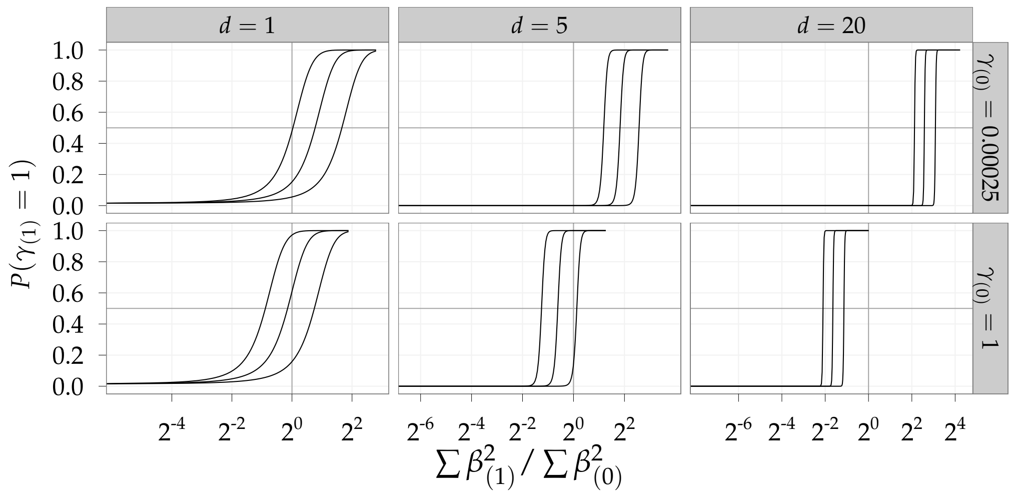


Figure 2.2.: $P(\gamma)$ as a function of the relative change in $\sum^d \beta^2$ for varying $d, \gamma_{(0)}$: Inclusion probability in iteration (1) as a function of the ratio between the sum of squared coefficients in iteration (1) and (0). Lines in each panel correspond to $\tau^2_{(1)}$ equal to the median of its full conditional and the .1- and .9-quantiles. Upper row is for $\gamma_{(0)} = 1$, lower row for $\gamma_{(0)} = v_0$. Columns correspond to $d = 1,\ 5,\ 20$. Fat gray grid lines denote inclusion probability $=$ .5 and ratio of coefficient sum of squares $= 1$

The following analysis will show that, even if the blockwise sampler is initially in an ideal state for switching between the spike and the slab parts of the prior, i.e. a parameter constellation so that the full conditional probability $P(\gamma = 1|\cdot) = .5$, such a switch is very unlikely in subsequent iterations for coefficient vectors with more than a few entries given the NMIG prior hierarchy.

Assume that the sampler starts out in iteration (0) with a parameter configuration of $a_t, b_t, v_0, w, \tau^2_{(0)}$ and $\beta_{(0)}$ so that $P(\gamma_{(0)} = 1|\cdot) = .5$. We set $w = .5$.

The parameters for which $P(\gamma = 1|\cdot) = .5$ satisfy the following relations:

$$\frac{P(\gamma = 1|\cdot)}{P(\gamma = v_0|\cdot)} = v_0^{d/2} \exp\left(\frac{(1-v_0)}{2v_0}\frac{\sum^d \beta^2}{\tau^2}\right) = 1,$$

so that $P(\gamma = 1|\cdot) > .5$ if

$$\begin{aligned} \frac{\sum^d \beta^2}{d\tau^2} &> -\frac{v_0}{1-v_0}\log(v_0) \\ \Leftrightarrow \sum^d \beta^2 &> -\frac{dv_0}{1-v_0}\log(v_0)\tau^2 \\ \Leftrightarrow \tau^2 &> -\frac{(1-v_0)\sum^d \beta^2}{dv_0 \log(v_0)}. \end{aligned}$$

Assuming a given value $\tau^2_{(0)}$, set

$$\sum^d \beta^2_{(0)} = \frac{dv_0}{1-v_0}\log(v_0)\tau^2_{(0)}.$$

Now $\gamma_{(0)}$ takes on both values v_0 and 1 with equal probability, conditional on all other parameters.

In the following iteration, $\tau^2_{(1)}$ is drawn from its full conditional $\Gamma^{-1}(a_t + d/2, b_t + \frac{\sum^d \beta^2_{(0)}}{2\gamma_{(0)}})$. Figure 2.2 shows $P(\gamma_{(1)} = 1|\tau^2_{(1)}, \sum^d \beta^2_{(1)})$ as a function of $\sum^d \beta^2_{(1)} / \sum^d \beta^2_{(0)}$ for various values of d. The 3 lines in each panel correspond to $P(\gamma_{(1)} = 1|\tau^2_{(1)}, \sum^d \beta^2_{(1)})$ for values of $\tau^2_{(1)}$ equal to the median of its full conditional as well as the .1- and .9-quantiles. The upper row in the Figure plots the function for $\gamma_{(0)} = 1$, the lower row for $\gamma_{(0)} = v_0$.

So, if we start in this "equilibrium state" we begin iteration (0) with v_0, w, $\tau^2_{(0)}$, and $\boldsymbol{\beta}_{(0)}$ so that $P(\gamma_{(0)} = 1|\cdot) = .5$. We then determine $P(\gamma_{(1)} = 1|\tau^2_{(1)}, \sum^d \beta^2_{(1)})$ as a function of $\sum^d \beta^2_{(1)} / \sum^d \beta^2_{(0)}$ for

- various values of $\dim(\boldsymbol{\beta}_j) = d$,
- $\gamma_{(0)} = 1$ and $\gamma_{(0)} = v_0$,
- $\tau^2_{(1)}$ at the .1, .5, .9-quantiles of its conditional distribution given $\boldsymbol{\beta}_{(0)}, \gamma_{(0)}$.

The leftmost column in Figure 2.2 shows that moving between $\gamma = 1$ and $\gamma = v_0$ is easy for $d = 1$: For a large range of realistic values for

$\sum^d \beta^2_{(1)} / \sum^d \beta^2_{(0)}$, moving back to $\gamma_{(1)} = v_0$ from $\gamma_{(0)} = 1$ (upper panel) has reasonably large probability, just as moving from $\gamma_{(0)} = v_0$ to $\gamma_{(1)} = 1$ (lower panel) is fairly likely for realistic values of $\sum^d \beta^2_{(1)} / \sum^d \beta^2_{(0)}$. For $d = 5$, however, $P(\gamma_{(1)} = 1|\cdot)$ already resembles a step function. For $d = 20$, if $\sum^d \beta^2_{(1)} / \sum^d \beta^2_{(0)}$ is not smaller than 0.48, the probability of moving from $\gamma_{(0)} = 1$ to $\gamma_{(1)} = v_0$ (upper panel) is practically zero for 90% of the values drawn from $p(\tau^2_{(1)}|\cdot)$. However, draws of β that reduce $\sum^d \beta^2$ by more than a factor of 0.48 while $\gamma = 1$ are unlikely to occur in real data. It is also extremely unlikely to move back to $\gamma_{(1)} = 1$ when $\gamma_{(0)} = v_0$, unless $\sum^d \beta^2_{(1)} / \sum^d \beta^2_{(0)}$ is larger than 2.9. Since the full conditional for β is very concentrated if $\gamma = v_0$, such moves are highly improbable and correspondingly the sampler is unlikely to move away from $\gamma = v_0$. Numerical values for the graphs in Figure 2.2 were computed for $a_\tau = 5$, $b_\tau = 50$, $v_0 = 0.005$ but similar problems arise for all suitable hyperparameter configurations.

In summary, mixing of the indicator variables γ will be very slow for long subvectors. In experiments, we observed posterior means of $P(\gamma = 1)$ to be either ≈ 0 or ≈ 1 across a wide variety of settings, even for very long chains, largely depending on the starting values of the chains. The following section describes a possible remedy.

2.2.3. Parameter expansion: the peNMIG model

The mixing problem analyzed in the previous section is similar to the mixing problems encountered in other samplers for hypervariances of regression coefficients: a small variance for a batch of coefficients implies small coefficient values and small coefficient values in turn imply a small variance so that a blockwise sampling scheme is unlikely to exit a basin of attraction around the origin. This problem has been previously described in Gelman et al. (2008), where the issue is framed as one of strong dependence between a block of coefficients and their associated hypervariance. A bimodal prior for the variance such as the NMIG prior where the Markov chain must additionally be able to switch between the different components of the mixture prior associated with the two modes of course exacerbates these difficulties. A promising strategy to reduce this dependence is the introduction of working parameters that are only partially identifiable along the lines of *parameter expansion* or *marginal augmentation* introduced for the EM-algorithm in Meng and van Dyk (1997) and developed further for Bayesian inference for hierarchical models in Gelman et al. (2008). While Gelman et al. (2008) focus on speeding up convergence for conventional hierarchical models, we use parameter expansion to enable simultaneous selection or deselection of coefficient subvectors and improve the shrinkage properties of the resulting marginal prior.

We add a non-identifiable multiplicative parameter expansion to the spike-and-slab prior. We set

$$\boldsymbol{\beta}_j = \alpha_j \boldsymbol{\xi}_j; \quad \boldsymbol{\xi}_j \in \mathbb{R}^{d_j}$$

for a subvector $\boldsymbol{\beta}_j$ with length d_j and use a scalar parameter

$$\alpha_j \sim \mathrm{NMIG}(v_0, w, a_\tau, b_\tau),$$

where NMIG denotes the prior hierarchy given in Figure 2.1. A similar prior hierarchy has recently been suggested for the selection of variance components in logistic models (Frühwirth-Schnatter and Wagner, 2010). Entries of the vector $\boldsymbol{\xi}_j$ are a priori distributed as

$$\xi_{jk} \overset{\text{i.i.d.}}{\sim} \frac{1}{2}N(1,1) + \frac{1}{2}N(-1,1),\ k = 1, \ldots, d_j,$$

and prior independence between α_j and $\boldsymbol{\xi}_j$ is assumed to hold. We write

$$\boldsymbol{\beta}_j \sim \mathrm{peNMIG}(v_0, w, a_\tau, b_\tau)$$

as shorthand for this prior structure.

The effective dimension of the coefficient vector associated with updating γ_j and τ_j^2 is then equal to one in every penalization group, since the Markov blankets of both γ_j and τ_j now only contain the scalar parameter α_j instead of the vector $\boldsymbol{\beta}_j$. This is crucial in order to avoid the mixing problems described in the previous Section, because instead of

$$\frac{P(\gamma = 1|\cdot)}{P(\gamma = v_0|\cdot)} = v_0^{d/2} \exp\left(\frac{(1-v_0)}{2v_0} \frac{\sum_i^d \beta_i^2}{\tau^2}\right)$$

for the conventional NMIG prior, we now have

$$\frac{P(\gamma = 1|\cdot)}{P(\gamma = v_0|\cdot)} = \sqrt{v_0} \exp\left(\frac{(1-v_0)}{2v_0} \frac{\alpha^2}{\tau^2}\right),$$

which is less susceptible to result in extreme values and behaves more like the probabilities in the leftmost column of Figure 2.2.

In our parameter expansion, the parameter α_j parameterizes the "importance" of the j-th coefficient block, while $\boldsymbol{\xi}_j$ "distributes" α_j across the entries in $\boldsymbol{\beta}_j$. Setting the conditional expectation of ξ to either positive or negative one shrinks the absolute value of ξ towards 1, the multiplicative identity, so that the interpretation of α_j as the "importance" of the j-th coefficient block can be maintained and yields a marginal prior for $\boldsymbol{\beta}_j$ that is less concentrated on small absolute values than $\xi \sim N(0,1)$.

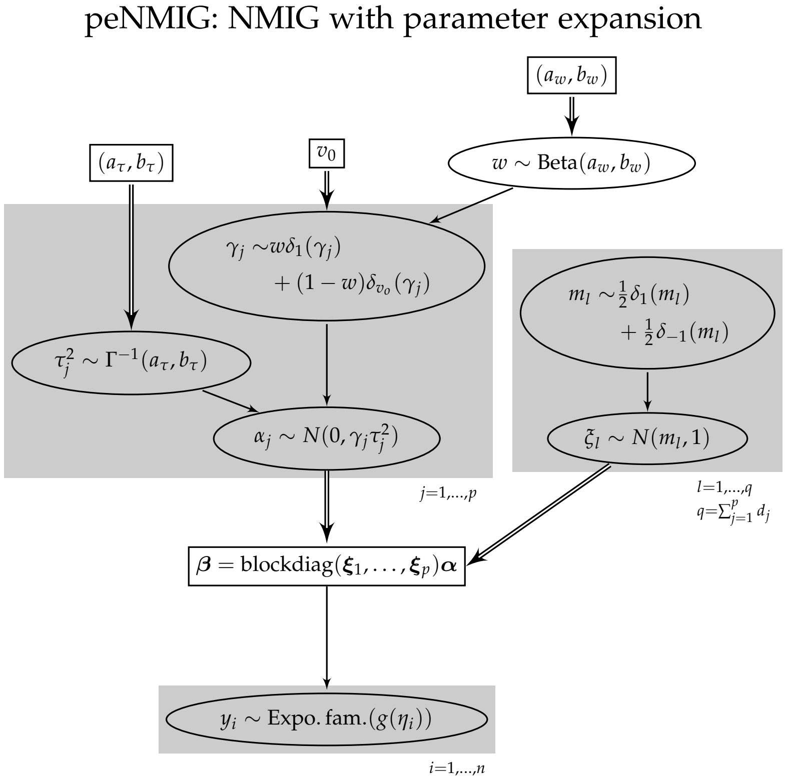

peNMIG: NMIG with parameter expansion
(a_w, b_w)
$w \sim \text{Beta}(a_w, b_w)$
(a_τ, b_τ)
v_0
$\gamma_j \sim w\delta_1(\gamma_j) + (1-w)\delta_{v_0}(\gamma_j)$
$\tau_j^2 \sim \Gamma^{-1}(a_\tau, b_\tau)$
$\alpha_j \sim N(0, \gamma_j\tau_j^2)$
$j{=}1,\ldots,p$
$m_l \sim \frac{1}{2}\delta_1(m_l) + \frac{1}{2}\delta_{-1}(m_l)$
$\xi_l \sim N(m_l, 1)$
$l{=}1,\ldots,q$
$q{=}\sum_{j=1}^{p} d_j$
$\boldsymbol{\beta} = \text{blockdiag}(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_p)\boldsymbol{\alpha}$
$y_i \sim \text{Expo. fam.}(g(\eta_i))$
$i{=}1,\ldots,n$

Figure 2.3.: Directed acyclic graph of NMIG model with parameter expansion. Ellipses are stochastic nodes, rectangles are deterministic/logical nodes. Single arrows are stochastic edges, double arrows are logical/deterministic edges.

Figure 2.3 shows the prior hierarchy for the model with parameter expansion. In the following, this model will be denoted as peNMIG. The vector $\boldsymbol{\xi} = (\boldsymbol{\xi}_1', \ldots, \boldsymbol{\xi}_p)'$ is decomposed into subvectors $\boldsymbol{\xi}_j$ associated with the different penalization groups and their respective entries α_j, $j = 1, \ldots, p$ in $\boldsymbol{\alpha}$.

2.2.4. Shrinkage properties

Marginal priors

This section investigates the regularization properties of the marginal prior for the regression coefficients $\boldsymbol{\beta}$ implied by the hierarchical prior structures given in Figs. 2.1 and 2.3. To distinguish between the conventional NMIG prior and its parameter expanded version we write β if the parameter has an

NMIG prior and β_{pe} if it has the parameter expanded peNMIG prior. In the following, we analyze the univariate marginal priors

$$\begin{aligned} p(\beta|a_\tau, b_\tau, a_w, b_w, v_0) &= \\ &= \int p(\beta|\gamma, \tau^2) p(\tau^2|a_\tau, b_\tau) p(\gamma|w, v_0) p(w|a_w, b_w) d\tau^2 d\gamma dw \end{aligned}$$

for the conventional NMIG model and

$$\begin{aligned} p(\beta_{pe} = \alpha\xi|a_\tau, b_\tau, a_w, b_w, v_0) & \\ = \int p(\alpha|\gamma, \tau^2) p\Bigg(\underbrace{\frac{\beta_{pe}}{\alpha}}_{=\xi}\Bigg) \frac{1}{|\alpha|} p(\tau^2|a_\tau, b_\tau) p(\gamma|a_w, b_w, v_0) & \\ p(w|a_w, b_w) d\alpha d\tau^2 d\gamma dw & \end{aligned}$$

for the peNMIG prior.

These are the univariate marginal priors for a single regression coefficient with and without parameter expansion with the intermediate quantities τ^2, γ and w integrated out. We analyze the marginal priors because it has been shown that the shrinkage properties of the resulting posterior means are dependent on their shape and less on that of the conditional priors (Fahrmeir, Kneib, and Konrath, 2010; Kneib, Konrath, and Fahrmeir, 2010). We use $v^2 = \gamma\tau^2 \sim \Gamma^{-1}(a_\tau, \gamma b_\tau)$ so that the marginal prior for β in the conventional NMIG-model is a mixture of scaled t-distributions with $2a_\tau$ degrees of freedom and scale factors $\sqrt{v_0 b_\tau / a_\tau}$ and $\sqrt{b_\tau / a_\tau}$ with weights $\frac{b_w}{a_w + b_w}$ and $\frac{a_w}{a_w + b_w}$, respectively:

$$\begin{aligned} & p(\beta|a_\tau, b_\tau, a_w, b_w, v_0) = \\ & = \frac{a_w}{a_w + b_w} \int_0^\infty p(\beta|v^2) p(v^2|a_\tau, b_\tau) dv^2 \\ & + \frac{b_w}{a_w + b_w} \int_0^\infty p(\beta|v^2) p(v^2|a_\tau, v_0 b_\tau) dv^2 \\ & = \frac{a_w}{a_w + b_w} \frac{b_\tau^{a_\tau}}{\sqrt{2\pi}\Gamma(a_\tau)} \int_0^\infty v^{-2(a+\frac{3}{2})} e^{-\frac{\frac{\beta^2}{2} + b_\tau}{v^2}} dv^2 \\ & + \frac{b_w}{a_w + b_w} \frac{(v_0 b_\tau)^{a_\tau}}{\sqrt{2\pi}\Gamma(a_\tau)} \int_0^\infty v^{-2(a+\frac{3}{2})} e^{-\frac{\frac{\beta^2}{2} + v_0 b_\tau}{v^2}} dv^2 \\ & = K_1 \int_0^\infty \left(\frac{v^2}{\frac{\beta^2}{2} + b_\tau} \right)^{-(a+\frac{3}{2})} e^{-\frac{\frac{\beta^2}{2} + b_\tau}{v^2}} \left(\frac{\beta^2}{2} + b_\tau \right)^{-(a_\tau + \frac{1}{2})} d\frac{v^2}{\frac{\beta^2}{2} + b_\tau} \end{aligned}$$

$$
\begin{aligned}
&+ K_2 \int_0^\infty \left(\frac{v^2}{\frac{\beta^2}{2} + v_0 b_\tau} \right)^{-(a_\tau + \frac{3}{2})} e^{-\frac{\frac{\beta^2}{2} + v_0 b_\tau}{v^2}} \left(\frac{\beta^2}{2} + v_0 b_\tau \right)^{-(a_\tau + \frac{1}{2})} d \frac{v^2}{\frac{\beta^2}{2} + v_0 b_\tau} \\
&= \frac{a_w}{a_w + b_w} \frac{b_\tau^{a_\tau} \Gamma(a_\tau + \frac{1}{2})}{\sqrt{2\pi}\Gamma(a_\tau) \left(\frac{\beta^2}{2} + b_\tau \right)^{a_\tau + \frac{1}{2}}} + \frac{b_w}{a_w + b_w} \frac{(v_0 b_\tau)^{a_\tau} \Gamma(a_\tau + \frac{1}{2})}{\sqrt{2\pi}\Gamma(a_\tau) \left(\frac{\beta^2}{2} + v_0 b_\tau \right)^{a + \frac{1}{2}}} \\
&= \frac{a_w}{a_w + b_w} \frac{\Gamma\left(\frac{2a_\tau + 1}{2} \right)}{\Gamma\left(\frac{2a_\tau}{2} \right) \sqrt{2 a_\tau \pi \frac{b_\tau}{a_\tau}}} \left(1 + \frac{\beta^2}{2 a_\tau \frac{b_\tau}{a_\tau}} \right)^{-\frac{2a_\tau + 1}{2}} \\
&+ \frac{b_w}{a_w + b_w} \frac{\Gamma\left(\frac{2a_\tau + 1}{2} \right)}{\Gamma\left(\frac{2a_\tau}{2} \right) \sqrt{2 a_\tau \pi \frac{v_0 b_\tau}{a_\tau}}} \left(1 + \frac{\beta^2}{2 a_\tau \frac{v_0 b_\tau}{a_\tau}} \right)^{-\frac{2a_\tau + 1}{2}} . \qquad (2.5)
\end{aligned}
$$

The marginal prior for β_{pe} in the peNMIG model has no closed form. The density given in (2.5) is also the marginal prior $p(\alpha | a_\tau, b_\tau, a_w, b_w, v_0)$ for α in the peNMIG model so that a density transform yields

$$
\begin{aligned}
&p(\beta_{pe} = \alpha\xi | a_\tau, b_\tau, a_w, b_w, v_0) = \\
&\qquad = \int p(\alpha | a_\tau, b_\tau, a_w, b_w, v_0) p \Bigg(\underbrace{\frac{\beta_{pe}}{\alpha}}_{=\xi} \Bigg) \frac{1}{|\alpha|} d\alpha \\
&\qquad = \int p \Bigg(\underbrace{\frac{\beta_{pe}}{\xi}}_{=\alpha} | a_\tau, b_\tau, a_w, b_w, v_0 \Bigg) p\left(\xi \right) \frac{1}{|\xi|} d\xi. \qquad (2.6)
\end{aligned}
$$

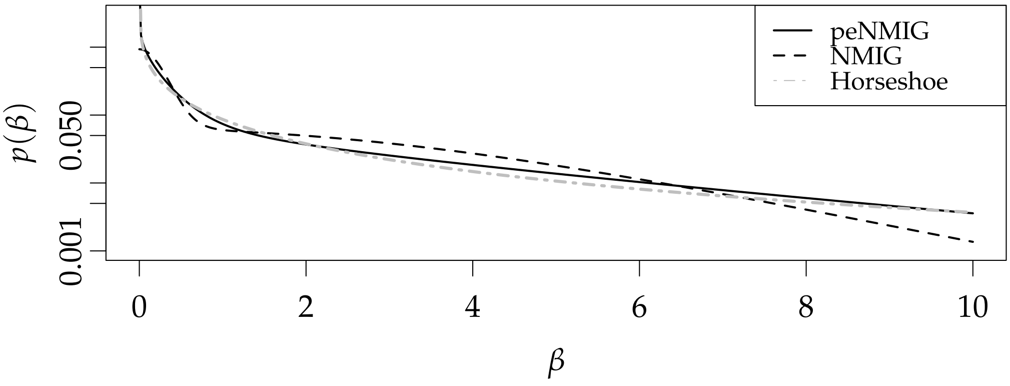


Figure 2.4.: Marginal priors for β as given in (2.5) and (2.6) with $(a_\tau, b_\tau) = (5, 50)$, $v_0 = 0.005$, $a_w = b_w$. Horseshoe prior in dashed grey. Vertical axis on log-scale.

Figure 2.4 shows the two marginal priors for $v_0 = 0.005$, $(a_\tau, b_\tau) = (5, 50)$ and $a_w = b_w$. Values for peNMIG were determined by numerical integration. Note the characteristic shape of the spike-and-slab prior for the marginal prior without parameter expansion: There is a – fairly rounded – "spike" around zero which corresponds to the contribution of the t-distribution with scale factor $\sqrt{v_0 b_\tau / a_\tau}$ and a "slab" which corresponds to the contribution of the t-distribution with scale factor $\sqrt{b_\tau / a_\tau}$. The prior for peNMIG has heavier tails and an infinite spike at zero (see (2.7)). It looks similar to the original spike-and-slab prior suggested by Mitchell and Beauchamp (1988), which used a mixture of a point mass in 0 and a uniform distribution on a finite interval, but sampling for our approach has the benefit of conjugate and proper priors. The similarity to the horseshoe prior (Carvalho, Polson, and Scott, 2010) is even more striking.

The following shows that the marginal prior $p(\beta_{pe})$ diverges in 0. We use

$$p(\beta_{pe}|a_\tau, b_\tau, a_w, b_w, v_0) = \int_{-\infty}^{+\infty} p_\alpha\left(\frac{\beta_{pe}}{\xi}\right) p_\xi(\xi) \frac{1}{|\xi|} d\xi,$$

so that

$$p(\beta_{pe}|a_\tau, b_\tau, a_w, b_w, v_0)|_{\beta_{pe}=0} = p_\alpha(0) \int_{-\infty}^{+\infty} p_\xi(\xi) \frac{1}{|\xi|} d\xi.$$

It is enough to show that $I = \int_{-\infty}^{+\infty} p_\xi(\xi) \frac{1}{|\xi|} d\xi$ diverges, since $p_\alpha(0)$ is finite and strictly positive. The prior $p_\xi()$ is a mixture of normal densities with variance 1 and means ± 1, so

$$\begin{aligned} I &= K \int_{-\infty}^{+\infty} \frac{1}{|\xi|} \left(\exp\left(-\frac{(\xi+1)^2}{2}\right) + \exp\left(-\frac{(\xi-1)^2}{2}\right)\right) d\xi \\ &= K(I_1 + I_2 + I_3 + I_4) \end{aligned}$$

with

$$\begin{aligned} I_1 &= \int_{-\infty}^{0} -\frac{1}{\xi} \exp\left(-\frac{(\xi+1)^2}{2}\right) d\xi, \qquad I_2 = \int_{0}^{+\infty} \frac{1}{\xi} \exp\left(-\frac{(\xi+1)^2}{2}\right) d\xi, \\ I_3 &= \int_{-\infty}^{0} -\frac{1}{\xi} \exp\left(-\frac{(\xi-1)^2}{2}\right) d\xi, \text{ and } I_4 = \int_{0}^{+\infty} \frac{1}{\xi} \exp\left(-\frac{(\xi-1)^2}{2}\right) d\xi. \end{aligned}$$

Note that $I_1 = I_4$ and $I_2 = I_3$. Since all 4 integrals are positive, it is enough to show that one of them diverges:

$$
\begin{aligned}
I_4 &= \int_0^1 \frac{1}{\xi} \underbrace{\exp\left(-\frac{(\xi-1)^2}{2}\right)}_{\geq e^{-\frac{1}{2}} \text{ for } \xi\in[0,1]} d\xi + \underbrace{\int_1^{+\infty} \frac{1}{\xi} \exp\left(-\frac{(\xi-1)^2}{2}\right) d\xi}_{=\tilde{K}\geq 0} \\
&\geq e^{-\frac{1}{2}} \int_0^1 \frac{1}{\xi} d\xi + \tilde{K} \\
&= e^{-\frac{1}{2}} \left[\ln(\xi)\right]_0^1 + \tilde{K} = +\infty. \qquad (2.7)
\end{aligned}
$$

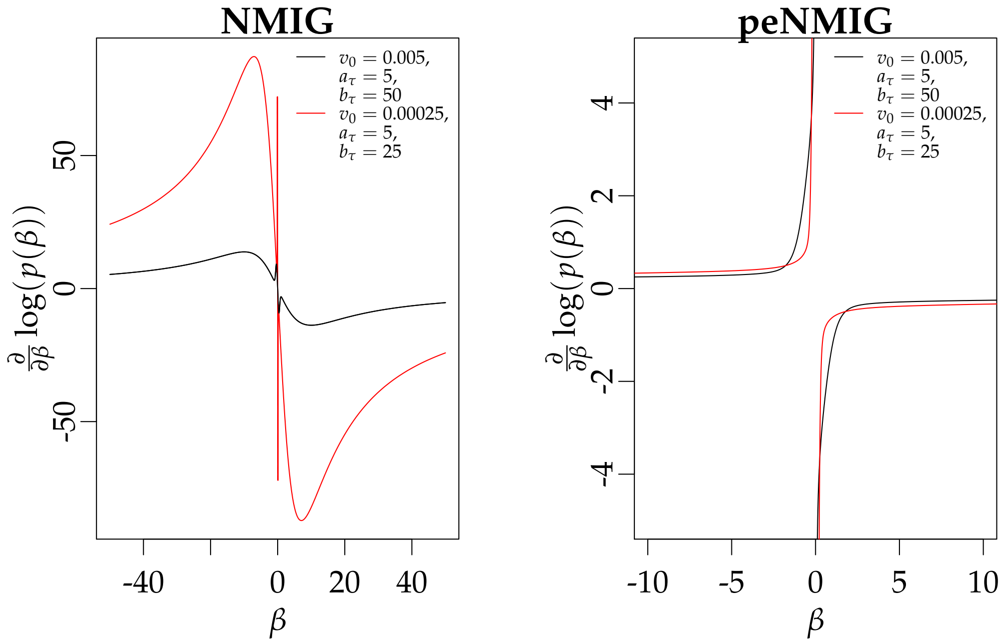


Figure 2.5.: Score functions for marginal priors for beta as given in (2.5) and (2.6). Note the different scales for the conventional NMIG and peNMIG.

For both NMIG and peNMIG, the tails of the marginal priors are heavy enough so that they have redescending score functions (see Figure 2.5) which ensures Bayesian robustness of the resulting estimators. While the shape of peNMIG's score function is similar to that of an $\mathcal{L}_q$-prior with $q \to 0$ and is fairly robust towards different combinations of hyperparameters, the conventional NMIG score function has a complicated shape determined by the interaction of a_τ, b_τ and v_0. Note that the score function of the marginal prior under parameter expansion descends monotonously and much faster.

The marginal prior of the hypervariances for $\beta_{pe} = \alpha\xi$ is given by the density of the product $\gamma\tau^2\xi^2$ since $\beta_{pe}|\gamma, \tau^2, \xi \sim N(0, \gamma\tau^2\xi^2)$. This marginal prior, which is the integral over the product of a mixture of scaled inverse gamma distributions with a noncentral χ^2_1 distribution

$$\begin{aligned} p(\lambda^2 =&\gamma\tau^2\xi^2) = \\ &= \int_0^\infty \left(\frac{a_w}{a_w + b_w} \Gamma^{-1} \left(\frac{\lambda^2}{\xi^2} | a_\tau, b_\tau \right) + \frac{b_w}{a_w + b_w} \Gamma^{-1} \left(\frac{\lambda^2}{\xi^2} | a_\tau, v_0 b_\tau \right) \right) \\ &\qquad \frac{1}{\xi^2} \chi^2_1(\xi^2 | \mu = 1) d\xi^2, \\ \Gamma^{-1}(x|a,b) &= \frac{a^b}{\Gamma(a)} x^{-(a+1)} \exp\left(-\frac{b}{x} \right), \\ \chi^2_1(x|\mu = 1) &= \frac{1}{2} \exp\left(-\frac{x+1}{2} \right) x^{-\frac{1}{4}} I_{-\frac{1}{2}}\left(\sqrt{x} \right), \end{aligned}$$

($I_v(y)$ denotes the modified Bessel function of the first kind) is intractable, so we are unable to verify whether conditions for Theorem 1 in Polson and Scott (2010) apply. Simulation results indicate that the peNMIG prior has similar robustness for large coefficient values and better sparsity recovery as the horseshoe prior (see p. 61 f.), for which the theorem applies.

The peNMIG prior combines an infinite spike at zero with heavy tails. This desirable combination is similar to other shrinkage priors such as the horseshoe prior and the normal-Jeffreys prior (Bae and Mallick, 2004) for which both robustness for large values of β and very efficient estimation of sparse coefficient vectors have been shown (Carvalho et al., 2010; Polson and Scott, 2010).

Constraint regions

The shapes of the 2-d constraint regions $\log p((\beta_1, \beta_2)') \leq$ const implied by the NMIG and peNMIG priors provide some further intuition about their shrinkage properties. The contours of the NMIG prior, depicted on the left in Figure 2.6, have different shapes depending on the distance from the origin. Close to the origin ($\beta <$ .3), they are circular and very closely spaced, implying strong ridge-type shrinkage – coefficient values this small fall into the "spike"-part of the prior and will be strongly shrunk towards zero. Moving away from the origin (.3 $< \beta <$ .8), the shape of the contours defining the constraint region morphs into a rhombus shape with rounded corners that is similar to that produced by a Cauchy prior. Still further from the origin ($1 < \beta < 2$), the contours become convex and resemble those of the contours of an $\mathcal{L}_q$ penalty function, i.e. a prior with $p(\beta) \propto \exp(-|\beta|^q)$, with $q < 1$. Coefficient pairs in this region will be shrunk towards one of the axes, depending on their posterior correlation and which of their maximum likelihood estimators is bigger. For even larger β, the shape of the contours is a

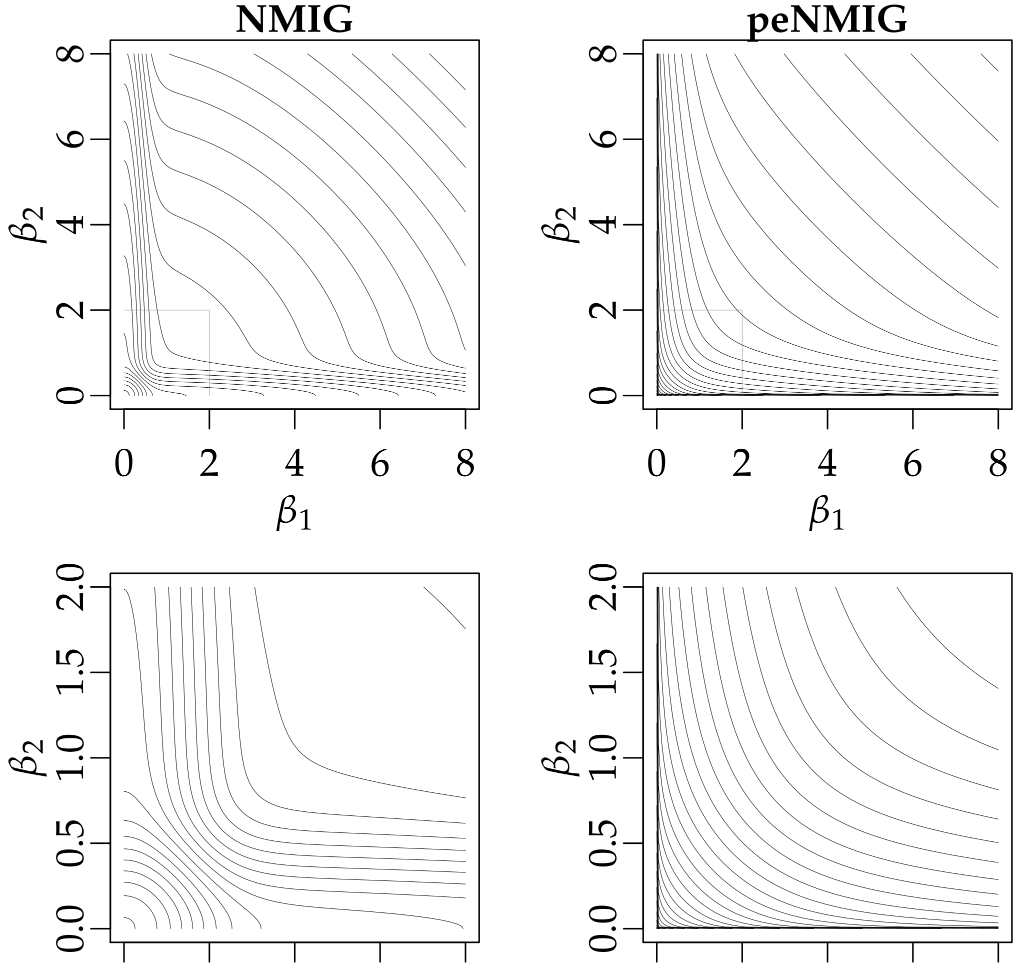


Figure 2.6.: Contour plots of $\log p((\beta_1, \beta_2)')$ for $a_\tau = 5$, $b_\tau = 50$, $v_0 = 0.005$, $a_w = b_w$ for the standard NMIG model and the model with parameter expansion. Lower panels are zooms into the region around the origin (indicated in the upper panels).

mixture of a ridge-type circular shape around the bisecting angle with pointy ends close to the axes. The concave shape of the contours in the areas far from the axes implies proportional (i.e. ridge-type) shrinkage of very large coefficient pairs. This corresponds to the comparatively smaller tail robustness of the conventional NMIG prior observed in simulations.

The shape of the constraint region implied by the peNMIG prior has the convex shape of a $\mathcal{L}_q$-penalty function with $q < 1$, which has the desirable properties of simultaneous strong shrinkage of small coefficients and weak shrinkage of large coefficients due to its closeness to the $\mathcal{L}_0$ penalty (see also Figure 2.8).

Until now, the discussion has been limited to bivariate shrinkage properties applied to single coefficients from *separate* penalization groups. In the following, we discuss shrinkage properties for coefficients from the *same* penalization group, i.e. two entries from the same subvector $\boldsymbol{\beta}_j$ in the notation of Figures 2.1 and 2.3. The shape of the peNMIG prior for 2 coefficients from the same penalization group is quite different. Recall that two coefficients (β_1, β_2) from the same penalization group share the same α, e.g. in this case $(\beta_1, \beta_2)' = \alpha(\xi_1, \xi_2)'$. This results in a very different shape of $\log p((\beta_1, \beta_2)') \leq$ const shown in Figure 2.7 (values determined by numerical integration). The prior in this case is

$$
\begin{aligned}
p(\boldsymbol{\beta}_{pe} = \alpha(\xi_1, \xi_2)' | a_\tau, b_\tau, a_w, b_w, v_0) = \\
&= \int p(\alpha | a_\tau, b_\tau, a_w, b_w, v_0) p\left(\frac{\boldsymbol{\beta}_{pe}}{\alpha}\right) \frac{1}{|\alpha|} d\alpha \\
&= \int p(\alpha | a_\tau, b_\tau, a_w, b_w, v_0) \frac{1}{|\alpha|} \cdot \\
&\quad \cdot \frac{1}{4}\left(N\left(\frac{\beta_1}{\alpha} | \mu = 1\right) + N\left(\frac{\beta_1}{\alpha} | \mu = -1\right)\right) \cdot \\
&\quad \cdot \left(N\left(\frac{\beta_2}{\alpha} | \mu = 1\right) + N\left(\frac{\beta_2}{\alpha} | \mu = -1\right)\right) d\alpha,
\end{aligned}
$$

where $N(x|\mu)$ denotes the normal density with variance 1 and mean μ. The shape of the constraint region for grouped coefficients is that of a square with rounded corners. Compared with the convex shape of the constraint region, this shape induces less shrinkage toward the axes and more towards the origin or along the bisecting angle.

Figure 2.8 illustrates the difference in shrinkage behavior between grouped and ungrouped coefficients for a simple toy example. We simulated design matrices $\boldsymbol{X}$ with $n = 15$ observations and 2 covariates so that $(\boldsymbol{X}'\boldsymbol{X})^{-1} = \left(\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right)$ with $\rho = -0.8, 0, 0.8$. Coefficients $\boldsymbol{\beta}$ were either $(1, 1)'$ (two intermediate effect sizes) or $(0, 2)'$ (one null, one large effect) and observations y were

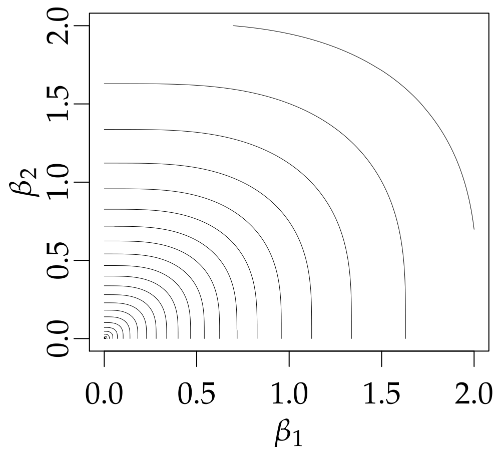


Figure 2.7.: Constraint region for $\boldsymbol{\beta} = (\beta_1, \beta_2)'$ from the same penalization group.

generated with a signal-to-noise ratio of 2. We generated 100 datasets for each combinations of ρ and $\boldsymbol{\beta}$ and compared OLS estimates to the posterior means for a peNMIG model as returned by `spikeSlabGAM`.

The different shrinkage properties for grouped and ungrouped coefficients are most apparent for uncorrelated coefficients (middle column): Shrinkage in this case occurs in directions orthogonal to the contours of the prior, so while the shape of the grouped prior causes shrinkage toward the origin in the direction of the bisecting angle or parallel to the axes, the ungrouped coefficients are shrunk more toward the nearest axis. Consequently, we expect estimation error for sparse coefficient vectors with few large and many small or zero entries (like $\boldsymbol{\beta} = (0, 2)'$) to be smaller for ungrouped coefficients, while the grouped prior should have a smaller bias for coefficient vectors with many entries of similar (absolute) size (like $\boldsymbol{\beta} = (1, 1)'$): While most of the mass of the multivariate prior for ungrouped coefficients is concentrated along the axes (i.e. on sparse coefficient vectors), the multivariate prior for grouped coefficients is concentrated in a cube around the origin.

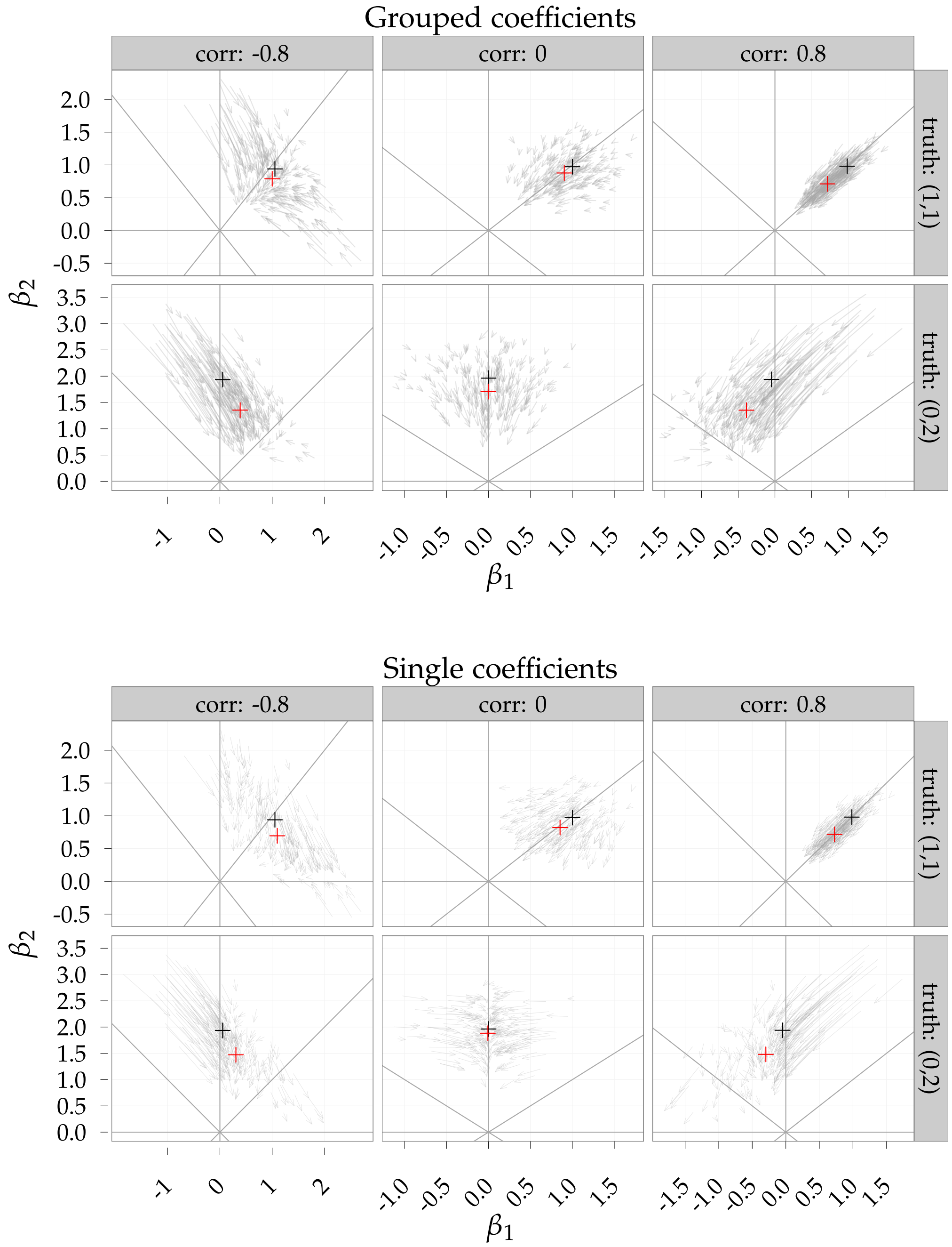


Figure 2.8.: Shrinkage for grouped (top) and ungrouped coefficients (bottom). Arrows connect OLS estimates with peNMIG posterior means on identical data sets. Black crosses denote means of OLS estimators over all replications, red crosses means of posterior means from peNMIG models fit with `spikeSlabGAM`. Top rows in each graph are for $\boldsymbol{\beta} = (1,1)'$, bottom rows for $\boldsymbol{\beta} = (0,2)'$. Columns show results for $\rho = -0.8, 0, 0.8$. ρ is the correlation of the OLS estimators.

3. spikeSlabGAM: Implementing Bayesian Variable Selection, Model Choice and Regularization for Generalized Additive Mixed Models in R

This chapter describes the implementation of the ideas developed in Chapter 2 available in the R package spikeSlabGAM. Section 3.1 discusses the types of terms available in for model fitting with spikeSlabGAM and describes the steps necessary to reparameterize general structured additive regression terms as given in (2.3) so that they can be included in the prior structure of Figure 2.3. Section 3.2 provides a detailed description of the MCMC sampler used in spikeSlabGAM. Section 3.3 concludes this chapter with a demonstration of and code examples for spikeSlabGAM's capabilities in terms of model fitting, model checking and visualization.

3.1. Setting up the design

All of the terms implemented in spikeSlabGAM have the following structure, as described in Section 2.1.2: First, their contribution to the predictor $\boldsymbol{\eta}$ (cf. (2.3)) is represented as a linear combination of basis functions, i.e., the term associated with a covariate or set of covariates $\boldsymbol{z}$ is represented as $f(\boldsymbol{z}) = \sum_{k=1}^{K} \delta_k B_k(\boldsymbol{z}) = \boldsymbol{B}\boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is a vector of coefficients associated with the basis functions $B_k(\cdot)$ $(k = 1, \dots, K)$ evaluated in $\boldsymbol{z}$. Second, $\boldsymbol{\delta}$ has a (conditionally) multivariate Gaussian prior, i.e., $\boldsymbol{\delta}|v^2 \sim N(\boldsymbol{0}, v^2\boldsymbol{P}^{-})$, with a fixed scaled precision matrix $\boldsymbol{P}$ that is often positive *semi*-definite.

3.1.1. Available terms

Table 3.1 gives an overview of the model terms available in spikeSlabGAM and how they fit into this framework.

R-syntax	Description	$\boldsymbol{B}$	$\boldsymbol{P}$
`lin(x, degree)`	linear/polynomial trend: basis functions are orthogonal polynomials of degree 1 to `degree` evaluated in `x`; defaults to `degree`$=1$	`poly(x, degree)`	identity matrix
`fct(x)`	factor: defaults to sum-to-zero contrasts	depends on contrasts	identity matrix
`rnd(x, C)`	random intercept: defaults to i. i. d.; i.e., correlation `C`$= \boldsymbol{I}$	indicator variables for each level of `x`	`C`$^{-1}$
`sm(x)`	univariate penalized spline: defaults to cubic B-splines with 2$^{\text{nd}}$ order difference penalty	B-spline basis functions	$\boldsymbol{\Delta}^{d^\top} \boldsymbol{\Delta}^d$ with $\boldsymbol{\Delta}^d$ the d^{th} diff. operator matrix
`srf(xy)`	penalized surface estimation on 2-D coordinates `xy`: defaults to tensor product cubic B-spline with first order difference penalties	(radial) basis functions (thin plate / tensor product B-spline)	depends on basis function
`mrf(x, N)`	first order intrinsic Gauss-Markov random field: factor `x` defines the grouping of observations, `N` defines the neighborhood structure of the levels in `x`	indicator variables for regions in `x`	precision matrix of MRF defined by (weighted) adjacency matrix `N`

Table 3.1.: Term types in `spikeSlabGAM`. The semiparametric terms (`sm()`, `srf()`, `mrf()`) only parameterize the proper part of their respective regularization priors (see Section 3.1.2). Unpenalized terms not associated with a peNMIG prior (i.e., the columns in $\boldsymbol{X}_u$ in (2.4)) are specified with term type `u()`.

3.1.2. Decomposition and reparameterization of regularized terms

In Section (2.2), we glossed over the fact that every coefficient batch $\boldsymbol{\delta}$ associated with a specific term $f(\boldsymbol{z}) = \boldsymbol{B}\boldsymbol{\delta}$ will have some kind of prior dependency structure determined by $\boldsymbol{P}$, since $\boldsymbol{\delta} \sim N(0, s^2\boldsymbol{P}^-)$. Moreover, if $\boldsymbol{P}$ is only positive *semi*-definite, the prior is partially improper. For example, the precision matrix for a B-spline with second order difference penalty implies an improper flat prior on the linear and constant components of the estimated function (Lang and Brezger, 2004), and the precision matrix for an intrinsic Gauss-Markov random field (IGMRF) of first order puts an improper flat prior on the mean level of the IGMRF (Rue and Held, 2005, ch. 3). These partially improper priors for splines and IGMRFs are problematic for spikeSlabGAM's purpose for two reasons: In the first place, if e.g., coefficient vectors that parameterize linear functions are in the nullspace of the prior precision matrix, the linear component of the function is estimated entirely unpenalized. This means that it is unaffected by the variable selection property of the peNMIG prior and thus always included in the model, but we want to be able to not only remove the entire effect of a covariate (i.e., both its penalized and unpenalized parts) from the model, but also to select or deselect its penalized and unpenalized parts separately. The second issue is that, as the nullspaces of these precision matrices usually also contain coefficient vectors that parameterize constant effects, terms in multivariate models are not identifiable, since adding a constant to one term and subtracting it from another does not affect the posterior.

Two strategies to resolve these issues are implemented in spikeSlabGAM. Both involve two steps:

1. Splitting terms $f(\boldsymbol{x})$ with partially improper priors into two parts – $f_0(\boldsymbol{x})$ associated with the improper/unpenalized part of the prior and $f_{\text{pen}}(\boldsymbol{x})$ associated with the proper/penalized part of the prior

2. Absorbing the fixed prior correlation structure of the coefficients implied by $\boldsymbol{P}$ into a transformed design matrix $\boldsymbol{X}_{\text{pen}}$ associated with then *a priori* independent coefficients $\boldsymbol{\beta}_{\text{pen}}$ for the penalized part.

Constant functions contained in the unpenalized part of a term are subsumed into a global intercept. This removes the identifiability issue. The remainder of the unpenalized component enters the model in a separate term $f_0(\boldsymbol{x})$, e.g., P-splines (term type `sm()`, see Table 3.1) leave polynomial functions of a certain order unpenalized and these enter the model in a separate `lin()`-term.

Orthogonal reduced rank decomposition

The first strategy, used by default, employs a reduced rank approximation of the implied covariance of $f(\boldsymbol{x})$ to construct a design $\boldsymbol{X}_{\text{pen}}$ for the penalized part of the function, similar to the approaches used in Reich et al. (2009) and Cottet et al. (2008):

Since

$$f(\boldsymbol{x}) = \boldsymbol{B}\delta \sim N(0, v^2\boldsymbol{B}\boldsymbol{P}^-\boldsymbol{B}^\top),$$

we can use the spectral decomposition $\boldsymbol{B}\boldsymbol{P}^-\boldsymbol{B}^\top = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top$ with orthonormal $\boldsymbol{U}$ and diagonal $\boldsymbol{D}$ with entries ≥ 0 to find an orthogonal basis representation for $\text{Cov}\,(f(\boldsymbol{x}))$. For $\boldsymbol{B}$ with d columns and full column rank and $\boldsymbol{P}$ with rank $d - n_P$, where n_P is the dimension of the nullspace of $\boldsymbol{P}$, all eigenvalues of $\text{Cov}\,(f(\boldsymbol{x}))$ except the first $d - n_P$ are zero. Now write

$$\text{Cov}\,(f(\boldsymbol{x})) = [\boldsymbol{U}_+ \boldsymbol{U}_0]^\top \left[\begin{smallmatrix} \boldsymbol{D}_+ & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{smallmatrix}\right] [\boldsymbol{U}_+ \boldsymbol{U}_0],$$

where $\boldsymbol{U}_+$ is a matrix of eigenvectors associated with the positive eigenvalues in $\boldsymbol{D}_+$, and $\boldsymbol{U}_0$ are the eigenvectors associated with the zero eigenvalues. With $\boldsymbol{X}_{\text{pen}} = \boldsymbol{U}_+\boldsymbol{D}_+^{1/2}$ and $\boldsymbol{\beta}_{\text{pen}} \sim N(0, v^2\boldsymbol{I})$, $f_{\text{pen}}(\boldsymbol{x}) = \boldsymbol{X}_{\text{pen}}\boldsymbol{\beta}_{\text{pen}}$ has a proper Gaussian distribution that is proportional to that of the partially improper prior of $f(\boldsymbol{x})$ (Rue and Held, 2005, eq. (3.16)) and parameterizes only the penalized part of $f(\boldsymbol{x})$, while the unpenalized part of the function is represented by $f_0(\boldsymbol{x}) = \boldsymbol{X}_0\boldsymbol{\beta}_0$ with $\boldsymbol{X}_0 = \boldsymbol{U}_0$.

In practice, it is unnecessary and impractically slow to compute all eigenvectors and -values for a full spectral decomposition $\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top$. Only the first $d - n_P$ are needed for $\boldsymbol{X}_{\text{pen}}$, and of those the first few typically represent most of the variability in $f_{\text{pen}}(\boldsymbol{x})$. `spikeSlabGAM` makes use of a fast truncated bidiagonalization algorithm (Baglama and Reichel, 2006) implemented in `irlba` (Lewis, 2009) to compute only the largest $d - n_P$ eigenvalues of $\text{Cov}\,(f(\boldsymbol{x}))$ and their associated eigenvectors. Only the first $\tilde{d}$ eigenvectors and -values whose sum represents at least .995 of the sum of all eigenvalues are used to construct the reduced rank orthogonal basis $\boldsymbol{X}_{\text{pen}}$. For a cubic P-spline with second order difference penalty and 20 basis functions (i.e., $d = 20$ columns in $\boldsymbol{B}$ and $n_P = 2$), $\boldsymbol{X}_{\text{pen}}$ will typically have only $\tilde{d}$=8 to 12 columns.

Mixed model decomposition

The second strategy reparameterizes via a decomposition of the coefficient vector $\boldsymbol{\delta}$ into an **u**npenalized part and a **p**enalized part: $\boldsymbol{\delta} = \boldsymbol{X}_u\boldsymbol{\beta}_0 + \boldsymbol{X}_p\boldsymbol{\beta}_{\text{pen}}$, where $\boldsymbol{X}_u$ is a basis of the n_P-dimensional nullspace of $\boldsymbol{P}$ and $\boldsymbol{X}_p$ is a basis of its complement.

`spikeSlabGAM` uses a spectral decomposition of $\boldsymbol{P}$ with

$$\boldsymbol{P} = [\boldsymbol{\Lambda}_{+}\boldsymbol{\Lambda}_{0}]^{\top} \left[\begin{smallmatrix} \boldsymbol{\Gamma}_{+} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix}\right] [\boldsymbol{\Lambda}_{+}\boldsymbol{\Lambda}_{0}],$$

where $\boldsymbol{\Lambda}_{+}$ is the matrix of eigenvectors associated with the positive eigenvalues in $\boldsymbol{\Gamma}_{+}$, and $\boldsymbol{\Lambda}_{0}$ are the eigenvectors associated with the zero eigenvalues. This decomposition yields $\boldsymbol{X}_u = \boldsymbol{\Lambda}_0$ and $\boldsymbol{X}_p = \boldsymbol{L}(\boldsymbol{L}^{\top}\boldsymbol{L})^{-1}$ with $\boldsymbol{L} = \boldsymbol{\Lambda}_{+}\boldsymbol{\Gamma}_{+}^{1/2}$. The model term can then be expressed as

$$\begin{aligned} \boldsymbol{B}\boldsymbol{\delta} &= \boldsymbol{B}(\boldsymbol{X}_u\boldsymbol{\beta}_0 + \boldsymbol{X}_p\boldsymbol{\beta}_{\text{pen}}) \\ &= \boldsymbol{X}_0\boldsymbol{\beta}_0 + \boldsymbol{X}_{\text{pen}}\boldsymbol{\beta}_{\text{pen}}, \end{aligned}$$

with $\boldsymbol{X}_0$ as the design matrix associated with the unpenalized part and $\boldsymbol{X}_{\text{pen}}$ as the design matrix associated with the penalized part of the term.

The prior for the coefficients associated with the penalized part after reparameterization is then $\boldsymbol{\beta}_{\text{pen}} \sim N(\mathbf{0}, v^2\boldsymbol{I})$, while $\boldsymbol{\beta}_0$ has a flat prior (cf. Kneib, 2006, ch. 5.1). For the purpose of term selection, this flat prior is subsequently replaced by a conditionally Gaussian prior. As for the other decomposition, `spikeSlabGAM` by default only uses the first $\tilde{d}$ eigenvectors and -values whose sum represents at least .995 of the sum of all eigenvalues to construct $\boldsymbol{X}_p$. This usually results in much less dimension reduction than the previous method, e.g., for a cubic P-spline with second order difference penalty and 20 basis functions, $\boldsymbol{X}_{\text{pen}}$ will typically have 14 to 16 columns.

3.1.3. Interactions

Design matrices for interaction effects are constructed from tensor products (i.e., column-wise Kronecker products) of the bases for the respective main effect terms or lower order interactions. A detailed discussion of constructing tensor product splines in this way is given in Wood, Scheipl, and Faraway (2011). A more rigorous derivation based on reproducing kernel Hilbert spaces in the context of smoothing spline ANOVA is in Gu (2002, Ch. 2.4).

For example, the complete interaction between two numeric covariates x_1 and x_2 with smooth effects modeled as P-splines with second order difference penalty consists of the interactions of their unpenalized parts (i.e., linear x_1-linear x_2), two varying-coefficient terms (i.e., smooth $x_1\times$ linear x_2, linear $x_1\times$ smooth x_2) and a 2-D nonlinear effect (i.e., smooth $x_1\times$ smooth x_2).

By default, `spikeSlabGAM` uses a reduced rank representation of these tensor product bases derived from their partial singular value decomposition as described above for the "orthogonal" decomposition in order to reduce the posterior's dimensionality and to speed up the sampling. As the marginal dependency structures for the main effects have been absorbed into their design matrices, the precision matrix associated with the coefficients of the interaction effect is the identity matrix, so the implied covariance of the interaction effect is simply the crossproduct of the interaction design matrix. Consequently, its low-rank approximation can be computed based on the singular value decomposition of the interaction design matrix instead of that of the product between the design, the associated covariance of the interaction coefficients and the transposed design in order to save some computational effort.

3.1.4. "Centering" and scaling the effects

By default, `spikeSlabGAM` makes the estimated effects of all terms orthogonal to the nullspace of their associated penalty and, for interaction terms, against the corresponding main effects. This is similar to the method described in Yau et al. (2003), where it was used to simplify expressions for the marginal likelihoods of candidate models.

In `spikeSlabGAM`, every $\boldsymbol{X}$ is transformed via

$$\boldsymbol{X} \to \boldsymbol{X} \left(\boldsymbol{I} - \boldsymbol{Z}(\boldsymbol{Z}^\top \boldsymbol{Z})^{-1} \boldsymbol{Z}^\top \right).$$

For simple terms (i.e., `fct()`, `lin()`, `rnd()`), $\boldsymbol{Z} = \mathbf{1}$ and the projection above simply enforces a sum-to-zero constraint on the estimated effect. For semi-parametric terms, $\boldsymbol{Z}$ is a basis of the nullspace of the implied prior on the effect. For interactions between d main effects,

$$\boldsymbol{Z} = \left[\mathbf{1}\ \boldsymbol{X}_{0,1}\ \boldsymbol{X}_{0,2} \ldots \boldsymbol{X}_{0,d}\ \boldsymbol{X}_{\text{pen},1}\ \boldsymbol{X}_{\text{pen},2} \ldots \boldsymbol{X}_{\text{pen},d} \right],$$

where $\boldsymbol{X}_{0,1}, \ldots, \boldsymbol{X}_{\text{pen},d}$ are the design matrices for the involved main effects. This centering improves separability between main effects and their interactions by removing any overlap of their respective column spaces. All uncertainty about the mean response level is shifted into the global intercept. The projection uses the QR decomposition of $\boldsymbol{Z}$ for speed and stability. Note that to ensure identifiability, it would suffice to center all design matrices for functions from the nullspace, so in a sense we are imposing additional constraints on the fit. Simulation results in Wood et al. (2011) indicate that these additional constraints do not affect the performance of fits in conventional GAMMs adversely in a relevant way. The simulation results discussed on p. 83 f. indicate that this centering, in fact, improves estimation results for `spikeSlabGAM`.

Since `spikeSlabGAM` uses the same prior for all model terms, we have to make sure that similar coefficient sizes imply similar effect sizes, otherwise term selection will be biased towards excluding terms for which relatively smaller coefficients translate into relatively larger contributions to the additive predictor and including terms for which the opposite is true. To do this, we scale the design matrices of all terms so that they have a Frobenius norm of 0.5.

3.1.5. Summary

In summary, `spikeSlabGAM` performs the following reparameterization from

$$\boldsymbol{\eta} = \boldsymbol{\eta}_0 + \sum_{j=1}^{P} \boldsymbol{B}_j \boldsymbol{\delta}_j$$

with $\boldsymbol{\delta}_j | s_j^2 \sim N(\mathbf{0}, s_j^2 \boldsymbol{P}_j^-)$ to

$$\boldsymbol{\eta} = \boldsymbol{\eta}_0 + \sum_{j=1}^{p} \boldsymbol{X}_j \boldsymbol{\beta}_j$$

with $\boldsymbol{\beta}_j | v_j^2 \sim N(\mathbf{0}, v_j^2 \boldsymbol{I})$ and $p \geq P$:

1. Split up all main effects $f_j(\boldsymbol{z}) = \boldsymbol{B}_j \boldsymbol{\delta}_j$ into a penalized part $f_{j,\text{pen}}(\boldsymbol{z}) = \boldsymbol{X}_{j,\text{pen}} \boldsymbol{\beta}_{j,\text{pen}}$ and, if the associated precision $\boldsymbol{P}_j$ is rank deficient, an unpenalized part $f_{j,0}(\boldsymbol{x}) = \boldsymbol{X}_{j,0} \boldsymbol{\beta}_{j,0}$. (cf. Section 3.1.2)

2. Orthogonalize all design matrices $\boldsymbol{X}_{j,\text{pen}}$ created in the previous step against the respective $\boldsymbol{X}_{j,0}$ or the intercept column if $\boldsymbol{X}_{j,0}$ does not exist in the case of positive definite $\boldsymbol{P}_j$.

3. Orthogonalize all design matrices $\boldsymbol{X}_{j,0}$ created in the first step against the intercept column. (cf. Section 3.1.4)

4. construct columnwise Kronecker products of the design matrices of the involved main effects for all interaction effects (cf. Section 3.1.3)

5. Calculate reduced rank representations of the interaction designs based on their truncated singular value decompositions.

6. Orthogonalize all design matrices for interaction effects created in the previous step against the main effect design matrices used in their construction. (cf. Section 3.1.3)

7. Scale all design matrices to have Frobenius norm 0.5.

Note that the prior for $\boldsymbol{\beta}_j$ is Gaussian *conditional* on the hypervariance $v_j^2 = \gamma_j \tau_j^2 \xi_j^2$. Also note that this reparameterization does not yield a one-to-one correspondence to the original priors since we place proper priors on functions in the nullspace of the original penalty. However, it has two distinct advantages: First, the conditionally i.i.d. Gaussian prior somewhat reduces the computational complexity of the MCMC sampler. More importantly, by assigning separate and proper priors on both penalized and unpenalized parts of all model terms, we can perform term selection on the penalized and the unpenalized parts separately. The resulting models are potentially more parsimonious and easier to interpret.

3.1.6. Computing predictions

While the reduced rank representation and orthogonalization of the model's main effects and interaction terms described above speed up the sampling and seem to improve estimation as well as the precision of variable selection (cf. p. 83 and results in Scheipl (2010b)), they come at a cost if the estimated model is used to generate predictions for covariate values z_{new} not present in the original data: Since the basis functions in the reparameterized design matrix $\boldsymbol{X}$ no longer have a closed form expression, it is not possible to simply generate appropriate design matrices for new data as would be done for, e.g., conventional spline or tensor product spline bases.

At least for the main effect terms, this problem can be overcome by using spline interpolation to evaluate the basis functions of the reduced rank representation at the new covariate locations z_{new}. Interpolation in two or more dimensions can be very unstable, however, so `spikeSlabGAM` uses a different and much more computationally intensive approach to generate predictions for reduced rank interaction effects: For any given reparameterized interaction term $f(z)$, coefficients $\boldsymbol{\beta}$ for the reparameterized basis $\boldsymbol{X}$ are back-transformed into analogous coefficients $\boldsymbol{\delta}$ for the original basis $\boldsymbol{B}$. These can then be used to generate predictions by multiplication with $\boldsymbol{B}_{\text{new}} = \boldsymbol{B}(z_{\text{new}})$, the un-reparameterized design matrix for the new covariate values z_{new}. Since $\hat{\boldsymbol{F}}$, the MCMC realizations from the posterior of $f(\boldsymbol{z})$, can be written as $\hat{\boldsymbol{F}} = \boldsymbol{X}[\boldsymbol{\beta}^{(1)}\ \boldsymbol{\beta}^{(2)} \dots \boldsymbol{\beta}^{(T)}]$, and also, by construction, $\hat{\boldsymbol{F}} = \boldsymbol{B}[\boldsymbol{\delta}^{(1)}\ \boldsymbol{\delta}^{(2)} \dots \boldsymbol{\delta}^{(T)}]$, where $\beta^{(t)}$ and $\delta^{(t)}$ are the t^{th} MCMC samples, we find $[\boldsymbol{\delta}^{(1)} \dots \boldsymbol{\delta}^{(T)}]$ by solving this system of equations. In `spikeSlabGAM`, this is done via the QR decomposition of $\boldsymbol{B}$. The posterior distribution of $f(\boldsymbol{z})$ at z_{new} can then be approximated with $\hat{\boldsymbol{F}}_{\text{new}} = \boldsymbol{B}_{\text{new}}[\boldsymbol{\delta}^{(1)} \dots \boldsymbol{\delta}^{(T)}]$.

3.2. MCMC

This section describes the MCMC sampler implemented in `spikeSlabGAM` that was used for all the simulations and applications in Sections 4.1, 4.3 and 4.4. Algorithm 1 on p. 43 gives a short summary of the blockwise Metropolis-within-Gibbs sampler we use.

3.2.1. Full conditionals

The sampler exploits the fact that the full conditionals of (most of) the parameters are available in closed form:

$$\begin{aligned} w|\cdot &\sim \text{Beta}\left(a_w + \sum_j^p \delta_1(\gamma_j), b_w + \sum_j^p \delta_{v0}(\gamma_j)\right), \\ \tau_j^2|\cdot &\sim \Gamma^{-1}\left(a_t + d_j/2, b_t + \frac{\sum_{i=1}^{d_j} \beta_{ji}^2}{2\gamma_j}\right), \\ \frac{P(\gamma_j = 1|\cdot)}{P(\gamma_j = v_0|\cdot)} &= v_0^{d_j/2} \exp\left(\frac{(1-v_0)}{2v_0}\frac{\sum_{i=1}^{d_j} \beta_{ji}^2}{\tau_j^2}\right). \end{aligned}$$

Full conditionals for $\boldsymbol{\beta}_j$ for Gaussian responses and the conventional NMIG model (given in Figure 2.1) are given by

$$\begin{aligned} &\boldsymbol{\beta}_j|\cdot \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \text{ with} \\ &\boldsymbol{\Sigma}_j = \left(\frac{1}{\sigma_\varepsilon^2}\boldsymbol{X}_j'\boldsymbol{X}_j + \frac{1}{\gamma_j\tau_j^2}\boldsymbol{I}_{d_j}\right)^{-1} \quad \text{and } \boldsymbol{\mu}_j = \frac{1}{\sigma_\varepsilon^2}\Sigma_j\boldsymbol{X}_j'\boldsymbol{y}. \end{aligned} \tag{3.1}$$

In the peNMIG model given in Figure 2.3, updates for $\boldsymbol{\alpha}$ use the "collapsed" design matrix $\boldsymbol{X}_\alpha = \boldsymbol{X}\,\text{blockdiag}(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_p)$, while $\boldsymbol{\xi}$ is updated based on a "rescaled" design matrix $\boldsymbol{X}_\xi = \boldsymbol{X}\,\text{diag}(\text{blockdiag}(\mathbf{1}_{d1}, \ldots, \mathbf{1}_{dp})\boldsymbol{\alpha})$, where $\mathbf{1}_d$ is a $d \times 1$ vector of ones. For Gaussian responses, these are draws from their multivariate normal full conditionals as above. For non-Gaussian responses, we use P-IWLS proposals (Lang and Brezger, 2004) with a Metropolis-Hastings step. The following Section 3.2.2 provides more details on the methods used to sample $\boldsymbol{\beta}$.

Note that

$\frac{P(\gamma_j = 1|\cdot)}{P(\gamma_j = v_0|\cdot)} > v_0^{d_j/2}$ for all values of $\boldsymbol{\beta}_j$, i.e that

$$P(\gamma_j = 1|\cdot) > \frac{v_0^{d_j/2}}{1 + v_0^{d_j/2}} \approx v_0^{d_j/2} \text{ for small } v_0.$$

3.2.2. Updating the coefficients

This section describes the implementation of the updates for the regression coefficients in the peNMIG model. For both Gaussian and non-Gaussian responses, the proposed algorithm does blockwise updates of coefficient subvectors, conditional on the remainder of the coefficient vector and the other parameters in the Markov blanket (i.e. prior covariances, prior means and the relevant likelihood terms). The default is a blocksize of 30 for both $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$ for Gaussian response and smaller blocksizes of 5 and 15 for $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$, respectively, for non-Gaussian response. Blocksizes are smaller for non-Gaussian response since the acceptance probability in the necessary Metropolis-Hastings-step for non-Gaussian responses tends to decrease quickly with increasing dimension of the proposal.

Since $\boldsymbol{\beta} = \text{blockdiag}(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_p)\boldsymbol{\alpha}$, we sample $\boldsymbol{\beta}$ by first updating $\boldsymbol{\alpha}$ based on a "collapsed" $n \times p$ design matrix $\boldsymbol{X}_\alpha = \boldsymbol{X}\,\text{blockdiag}(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_p)$ and then updating $\boldsymbol{\xi}$ based on a "rescaled" $n \times q$ design matrix $\boldsymbol{X}_\xi = \boldsymbol{X}\,\text{blockdiag}(\mathbf{1}_{d1}, \dots, \mathbf{1}_{dp})\boldsymbol{\alpha}$, where $\mathbf{1}_d$ is a $d \times 1$ vector of ones. The j-th column of $\boldsymbol{X}_\alpha$ contains the sum of the original design columns multiplied by the entries in the subvector $\boldsymbol{\xi}_j$ associated with α_j. Each column in $\boldsymbol{X}_\xi$ contains the respective column of the original design matrix multiplied by the associated entry in $\boldsymbol{\alpha}$. The prior means $m_l \in \{\pm 1\}$ for $\xi_l \sim N(m_l, 1)$ are drawn beforehand from their full conditionals via $P(m_l = 1|\cdot) = \frac{1}{1+\exp(-2\xi_l)}$.

Update via QR-decomposition

The following paragraphs describe a general method to update a coefficient vector $\boldsymbol{\delta}$ associated with a conditional Gaussian prior. We use this procedure to update $\boldsymbol{\beta}$ in the NMIG model and to update both $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$ in the peNMIG model.

Regression coefficients $\boldsymbol{\delta}$ with prior $\boldsymbol{\delta} \sim N(\boldsymbol{\mu}^\delta, \boldsymbol{\Sigma}^\delta)$ and associated design matrix $\boldsymbol{X}^\delta$ can be updated by running the regression of an augmented data vector $\tilde{\boldsymbol{y}}$ with covariance $\widetilde{\boldsymbol{\Sigma}}$ on an augmented design matrix $\tilde{\boldsymbol{X}}$ with

$$\tilde{\boldsymbol{y}} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{\mu}^\delta \end{pmatrix}; \quad \widetilde{\boldsymbol{X}} = \begin{pmatrix} \boldsymbol{X}^\delta \\ \boldsymbol{I} \end{pmatrix} \text{ and } \widetilde{\boldsymbol{\Sigma}} = \begin{pmatrix} \text{Cov}(\boldsymbol{y}) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}^\delta \end{pmatrix}. \tag{3.2}$$

If only a subvector $\boldsymbol{\delta}_j$ is updated conditional on the remainder $\boldsymbol{\delta}_{-j}$ of the vector $\boldsymbol{\delta}$, $\boldsymbol{y}$ is replaced by $\boldsymbol{y} - \boldsymbol{X}^\delta_{-j}\boldsymbol{\delta}_{-j}$ and $\boldsymbol{\Sigma}^\delta$ is replaced by $\boldsymbol{\Sigma}^\delta_{-j,-j}$.

Following Gelman et al. (2008), we perform the updates for the regression coefficients via the QR-decomposition $\widetilde{\boldsymbol{\Sigma}}^{-1/2}\tilde{\boldsymbol{X}} = \boldsymbol{QR}$. From this decomposition, we can solve the triangular system $\boldsymbol{R}\hat{\boldsymbol{\delta}} = \boldsymbol{Q}\left(\widetilde{\boldsymbol{\Sigma}}^{-1/2}\tilde{\boldsymbol{y}}\right)$ for the mean of the full conditional $\hat{\boldsymbol{\delta}}$. As long as $\widetilde{\boldsymbol{\Sigma}}^{-1/2}$ is a diagonal matrix, as is the case for all of the models and predictor terms we are considering (see Section 3.1.2), or is known, the computationally demanding step is the computation of the QR-decomposition.

We solve another triangular system $\boldsymbol{R}\boldsymbol{e}_\delta = \boldsymbol{n}$, $n_i \overset{\text{i.i.d.}}{\sim} N(0,1)$ in order to generate a candidate value $\boldsymbol{\delta}^c = \hat{\boldsymbol{\delta}} + \boldsymbol{e}_\delta$ from (the approximation to) the full conditional, so the proposal distribution $q(\boldsymbol{\delta}^c, \boldsymbol{\delta})$ is $N(\hat{\boldsymbol{\delta}}, (\boldsymbol{R}'\boldsymbol{R})^{-1})$.

IWLS updates for non-Gaussian responses

We use a variant of the well-known IWLS proposal scheme (Gamerman, 1997) to do blockwise updates for both $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$ in the non-Gaussian case. We use a penalized IWLS (P-IWLS) proposal scheme based on an approximation of the current posterior mode described in detail in Brezger and Lang (2006) (Sampling scheme 1, Section 3.1.1). This method is a Metropolis-Hastings type update which uses a Gaussian (i.e. second order Taylor) approximation to the full conditional around its approximate mode as its proposal distribution. The approximating Gaussian is obtained by performing a single Fisher scoring step per iteration.

For P-IWLS, $\boldsymbol{y}$ and $\text{Cov}(\boldsymbol{y})$ in (3.2) are replaced by their IWLS equivalents (Gamerman, 1997)

$$\text{Cov}(\boldsymbol{y}) \overset{IWLS}{\approx} \text{diag}\left(b''(\boldsymbol{\theta})g'(\boldsymbol{\mu})^2\right) \text{ and } \boldsymbol{y} \overset{IWLS}{\approx} \boldsymbol{X}_j\boldsymbol{\delta}_j + (\boldsymbol{y}-\boldsymbol{\mu})g'(\boldsymbol{\mu}), \tag{3.3}$$

see (2.1) for notation.

We use the following modification of the IWLS-algorithm in order to decrease the computational complexity of the algorithm somewhat: By using the mean of the proposal distribution of the previous iteration $\hat{\boldsymbol{\delta}}^p$ instead of $\boldsymbol{\delta}$ in (3.3) and recalculating $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$ based on $\hat{\boldsymbol{\delta}}^p$, the proposal distribution $q()$ becomes independent of the current state, which simplifies the calculation of the acceptance probability and can increase acceptance rates (Brezger and Lang, 2006).

Acceptance rates for the sampler strongly depend on the size of the update blocks and on the magnitude of the rescaling performed in each iteration: For large blocks or updates that require drastic rescaling (see paragraph below), acceptance probabilities can occasionally become small, especially for binary responses. To avoid getting stuck, we use a different type of proposal with probability 0.15: Instead of drawing proposals from $N(\hat{\boldsymbol{\delta}}^p, (\boldsymbol{R}'\boldsymbol{R})^{-1})$, we use

$q(\boldsymbol{\delta}^c, \boldsymbol{\delta}) = N(\boldsymbol{\delta}^c, (\boldsymbol{R}'\boldsymbol{R})^{-1})$, i.e. we use the current state as the mean of the proposal. The working observations and IWLS weights that determine $\boldsymbol{R}$ are calculated from the mode of the previous iteration as described above so that the proposal ratio $q(\boldsymbol{\delta}, \boldsymbol{\delta}^c)/q(\boldsymbol{\delta}^c, \boldsymbol{\delta})$ is 1. This type of update tends to result in smaller steps, but it can be useful in order to keep the chain moving. For most datasets, mode switching is not necessary for good sampling performance, and `spikeSlabGAM` provides the option to switch it off entirely.

Rescaling parameter blocks

After updating the entire $\boldsymbol{\alpha}-$ and $\boldsymbol{\xi}-$vectors, each subvector $\boldsymbol{\xi}_j$ is rescaled so that $|\boldsymbol{\xi}_j|$ has mean 1, and the associated α_j is rescaled accordingly so that $\boldsymbol{\beta}_j = \alpha_j \boldsymbol{\xi}_j$ is unchanged:

$$\boldsymbol{\xi}_j \rightarrow \frac{d_j}{\sum_i^{d_j} |\xi_{ji}|} \boldsymbol{\xi}_j \quad \text{and} \quad \alpha_j \rightarrow \frac{\sum_i^{d_j} |\xi_{ji}|}{d_j} \alpha_j.$$

This rescaling is advantageous since α_j and $\boldsymbol{\xi}_j$ are not identifiable and thus their sampling paths can wander off into extreme regions of the parameter space without affecting the fit, e.g. α_j becoming extremely large while entries in $\boldsymbol{\xi}_j$ simultaneously become extremely small. By rescaling, we ensure that the interpretation of α_j as a scaling factor representing the importance of the model term associated with it is valid and avoid numerical problems that can occur for extreme parameter values. For non-Gaussian responses, the posterior modes used in the IWLS-updates are shifted accordingly as well. Note, however, that this shifting of the mode is only approximate. Consequentially, this rescaling can occasionally lead to low ($< .1$) acceptance rates for the P-IWLS proposals since the proposal density may not be well adapted to the posterior anymore after a large rescaling.

Starting values

By default, starting values $\boldsymbol{\beta}^{(0)}$ are drawn randomly in three steps: First, 5 Fisher scoring steps with fixed, large hypervariances are performed to reach a viable region of the parameter space. Second, for each chain run in parallel, Gaussian noise is added to this preliminary $\boldsymbol{\beta}^{(0)}$, and third its constituting p subvectors are scaled with variance parameters $\gamma_j \tau_j^2$ $(j = 1, \dots, p)$ drawn from their priors. This means that, for each of the parallel chains, some of the p model terms are set close to zero initially, and the remainder is in the vicinity of their respective ridge-penalized MLEs. Starting values for $\boldsymbol{\alpha}^{(0)}$ and

$\boldsymbol{\xi}^{(0)}$ are computed via

$$\alpha_j^{(0)} = \frac{\sum_i^{d_j} |\beta_{ji}|}{d_j} \quad \text{and} \quad \boldsymbol{\xi}_j^{(0)} = \frac{\boldsymbol{\beta}_j}{\alpha_j^{(0)}}.$$

Simulation results and applications show that this strategy works well.

Algorithm 1 MCMC sampler for peNMIG

```
Initialize $\boldsymbol{\tau}^{2(0)}, \boldsymbol{\gamma}^{(0)}, \phi^{(0)}, w^{(0)}$ and $\boldsymbol{\beta}^{(0)}$ (via IWLS for non-Gaussian response)
Compute $\boldsymbol{\alpha}^{(0)}, \boldsymbol{\xi}^{(0)}, \boldsymbol{X}_\alpha^{(0)}$
for iterations $t = 1, \dots, T$ do
  for blocks $b = 1, \dots, b_\alpha$ do
    update $\boldsymbol{\alpha}_b^{(t)}$ from its FCD (Gaussian case, see (3.1)) or via P-IWLS
  set $\boldsymbol{X}_\xi^{(t)} = \boldsymbol{X}\,\text{blockdiag}(\mathbf{1}_{d_1}, \dots, \mathbf{1}_{d_p})\boldsymbol{\alpha}^{(t)}$
  update $m_1^{(t)}, ..., m_q^{(t)}$ from their FCD: $P(m_l^{(t)} = 1|\cdot) = \frac{1}{1+\exp(-2\xi_l^{(t)})}$
  for blocks $b = 1, \dots, b_\xi$ do
    update $\boldsymbol{\xi}_b^{(t)}$ from its FCD (Gaussian case, see (3.1)) or via P-IWLS
  for model terms $j = 1, \dots, p$ do
    rescale $\boldsymbol{\xi}_j^{(t)}$ and $\alpha_j^{(t)}$
  set $\boldsymbol{X}_\alpha^{(t)} = \boldsymbol{X}\,\text{blockdiag}(\boldsymbol{\xi}_1^{(t)}, \dots, \boldsymbol{\xi}_p^{(t)})$
  update ${\tau_1}^{2(t)}, ..., {\tau_p}^{2(t)}$ from their FCD:
      $\tau_j^{2(t)}|\cdot \sim \Gamma^{-1}\left(a_\tau + 1/2, b_\tau + \frac{\alpha_j^{2(t)}}{2\gamma_j^{(t)}}\right)$
  update ${\gamma_1}^{(t)}, ..., {\gamma_p}^{(t)}$ from their FCD:
      $\frac{P(\gamma_j^{(t)}=1|\cdot)}{P(\gamma_j^{(t)}=v_0|\cdot)} = v_0^{1/2}\exp\left(\frac{(1-v_0)}{2v_0}\frac{\alpha_j^{2(t)}}{\tau_j^{2(t)}}\right)$
  update $w^{(t)}$ from its FCD:
      $w^{(t)}|\cdot \sim \text{Beta}\left(a_w + \sum_j^p \delta_1(\gamma_j^{(t)}), b_w + \sum_j^p \delta_{v0}(\gamma_j^{(t)})\right)$
  if $\boldsymbol{y}$ is Gaussian then
    update $\phi^{(t)}$ from its FCD:
      $\phi^{(t)}|\cdot \sim \Gamma^{-1}\left(a_\phi + n/2, b_\phi + \frac{\sum_i^n (y_i - \eta_i^{(t)})^2}{2}\right)$
```

3.2.3. Estimating inclusion probabilities

Selection of coefficient blocks $\boldsymbol{\beta}_j$ in the NMIG and peNMIG models is based on the marginal posterior of γ_j. The posterior expectation of $\delta_1(\gamma_j)$ is the posterior inclusion probability $p_{in,j}$, since $p_{in,j} = P(\gamma_j = 1) = E(\delta_1(\gamma_j))$. Inclusion probabilities $p_{in,j}$ are estimated with the Rao-Blackwellized estimator

$$\widehat{p_{in,j}} = T^{-1} \sum_{t=0}^{T} p_{in,j}^{(t)} \,,$$

$$\text{with } p_{in,j}^{(t)} = 1 - \begin{cases} \left(1 + v_0^{d_j/2} \exp\left(\frac{(1-v_0)}{2v_0} \frac{\sum_{i=1}^{d_j} (\beta_{ji}^{(t)})^2}{(\tau_j^2)^{(t)}}\right)\right)^{-1} & \text{for NMIG,} \\ \left(1 + v_0^{1/2} \exp\left(\frac{(1-v_0)}{2v_0} \frac{(\alpha_j^{(t)})^2}{(\tau_j^2)^{(t)}}\right)\right)^{-1} & \text{for peNMIG,} \end{cases}$$

where $\theta^{(t)}$ denotes the realized value of parameter θ in iteration t of an MCMC chain with length T. This estimator uses the MCMC samples of $P(\gamma_j = 1)$ after burn-in, instead of $\widehat{p_{in,j}} = T^{-1} \sum_{t=0}^{T} \delta_1(\gamma_j^{(t)})$.

3.2.4. Algorithm variants

While the default prior for the inclusion indicators γ_j assumes mutual independence, i.e. that inclusion or exclusion of a model term is a priori independent of the inclusion or exclusion of all other model terms, we also implemented a structure of the prior for $\boldsymbol{\gamma}$ that incorporates the hierarchical structure of the model terms themselves. More precisely, the prior structure forces inclusion of e.g. the linear term for a covariate if the corresponding smooth term is included in the model, or the inclusion of main effects if an interaction effect involving them is included in the model. Without changing the sampler per se, this "top-down" approach is implemented as a simple pass over the updated $\boldsymbol{\gamma}$-vector in each iteration, making sure that all low-order terms (i.e. main effects) have $\gamma = 1$ if high-order terms that involve them (i.e. interactions) have $\gamma = 1$. Alternatively, a "bottom-up" variant enforcing more parsimonious models that excludes high-order terms (i.e. sets them to $\gamma = v_0$) unless all low-order terms associated with them are included may be an option worth pursuing, but we have not done so yet.

An alternative to be implemented in future versions of the software is to sample $\boldsymbol{\gamma}$ not via single-site updates, but blockwise with blocks determined by the dependencies induced by the hierarchy (e.g. sample γs for main effects and their interaction together) and then include a Metropolis-Hastings step to reject proposals that violate the hierarchical constraints in a block. Previous work along these lines relied on an Ising prior for γ: Smith and Fahrmeir (2007) use an Ising prior on γ in an fMRI application for spatial smoothing of activation profiles across and Li and Zhang (2010) use it to incorporate prior knowledge and preferences about the predictor structure in an SSVS framework.

3.3. Using spikeSlabGAM

3.3.1. Model specification and post-processing

spikeSlabGAM uses the standard R formula syntax to specify models, with a slight twist: Every term in the model has to belong to one of the term types given in Table 3.1. If a model formula contains "raw" terms not wrapped in one of these term type functions, the package will try to guess appropriate term types: For example, the formula `y ~ x + f` with a numeric `x` and a factor `f` is expanded into `y ~ lin(x) + sm(x) + fct(f)` since the default is to model any numeric covariate as a smooth effect with a `lin()`-term parameterizing functions from the nullspace of its penalty and an `sm()`-term parameterizing the penalized part. The model formula defines the candidate set of model terms that comprise the model of maximal complexity under consideration. Terms are selected or removed without hierarchical constraints, i.e., an interaction effect can be included in the model even if the associated main effects or lower order interactions are not.

We generate some artificial data for a didactic example. We draw $n = 200$ observations from the following data generating process:

- covariates `sm1, sm2, noise2, noise3` are $\overset{\text{i.i.d.}}{\sim} U[0,1]$,
- covariates `f, noise4` are factors with 3 and 4 levels,
- covariates `lin1, lin2, lin3` are $\overset{\text{i.i.d.}}{\sim} N(0,1)$,
- covariate `noise1` is collinear with `sm1`: $\texttt{noise1} = \texttt{sm1} + \boldsymbol{e}$; $e_i \overset{\text{i.i.d.}}{\sim} N(0,1)$,
- $\boldsymbol{\eta} = f(\texttt{sm1}) + f(\texttt{sm2,f}) + 0.1 \cdot \texttt{lin1} + 0.2 \cdot \texttt{lin2} + 0.3 \cdot \texttt{lin3}$ (see Figures 3.3 and 3.4 for the shapes of the nonlinear effects $f(\texttt{sm1})$ and $f(\texttt{sm2,f})$),
- the response vector $\boldsymbol{y} = \boldsymbol{\eta} + \frac{\text{sd}(\boldsymbol{\eta})}{\text{snr}}\boldsymbol{\epsilon}$ is generated under signal-to-noise ratio $\text{snr} = 3$ with i.i.d. t_5-distributed errors ϵ_i $(i = 1, \dots, n)$.

```
R> set.seed(1312424)
R> n <- 200
```

```
R> snr <- 3
R> sm1 <- runif(n)
R> fsm1 <- dbeta(sm1, 7, 3)/2
R> sm2 <- runif(n, 0, 1)
R> f <- gl(3, n/3)
R> ff <- as.numeric(f)/2
R> fsm2f <- ff + ff * sm2 + ((f == 1) * -dbeta(sm2, 6, 4) +
+      (f == 2) * dbeta(sm2, 6, 9) + (f == 3) * dbeta(sm2,
+      9, 6))/2
R> lin <- matrix(rnorm(n * 3), n, 3)
R> colnames(lin) <- paste("lin", 1:3, sep = "")
R> noise1 <- sm1 + rnorm(n)
R> noise2 <- runif(n)
R> noise3 <- runif(n)
R> noise4 <- sample(gl(4, n/4))
R> eta <- fsm1 + fsm2f + lin %*% c(0.1, 0.2, 0.3)
R> y <- eta + sd(eta)/snr * rt(n, df = 5)
R> d <- data.frame(y, sm1, sm2, f, lin, noise1, noise2,
+      noise3, noise4)
```

We fit an additive model with all covariates as main effects and first-order interactions between the first 4 as potential model terms:

```
R> f1 <- y ~ (sm1 + sm2 + f + lin1)^2 + lin2 + lin3 + noise1 +
+      noise2 + noise3 + noise4
```

The function `spikeSlabGAM` sets up the design matrices, calls the sampler and returns the results:

```
R> m <- spikeSlabGAM(formula = f1, data = d)
```

The following output shows the first part of the `summary` of the fitted model. Note that the numeric covariates have been split into `lin()`- and `sm()`-terms and that the factors have been correctly identified as `fct()`-terms. The joint effect of the two numerical covariates `sm1` and `sm2` has been decomposed into 8 components: the 4 marginal linear and smooth terms, their linear-linear interaction, two "varying coefficient" terms (i.e., linear-smooth interactions) and a smooth interaction surface. This decomposition can be helpful in constructing parsimonious models. If a decomposition into marginal and joint effects is irrelevant or inappropriate, bivariate smooth terms can alternatively be specified with a `srf()`-term. `Mean posterior deviance` is $\frac{1}{T}\sum_t^{\top} -2l(\boldsymbol{y}|\boldsymbol{\eta}^{(t)}, \phi^{(t)})$, the average of twice the negative log-likelihood of the observations over the saved MCMC iterations, the `null deviance` is twice the negative log-likelihood of an intercept model without covariates.

```
R> summary(m)

Spike-and-Slab STAR for Gaussian data
Model:
y ~ ((lin(sm1) + sm(sm1)) + (lin(sm2) + sm(sm2)) + fct(f) + (lin(lin1) +
    sm(lin1)))^2 + (lin(lin2) + sm(lin2)) + (lin(lin3) + sm(lin3)) +
    (lin(noise1) + sm(noise1)) + (lin(noise2) + sm(noise2)) +
    (lin(noise3) + sm(noise3)) + fct(noise4) - lin(sm1):sm(sm1) -
    lin(sm2):sm(sm2) - lin(lin1):sm(lin1)
200 observations; 257 coefficients in 37 model terms.

Prior:
    a[tau]     b[tau]       v[0]       a[w]       b[w] a[sigma^2]
   5.0e+00    2.5e+01    2.5e-04    1.0e+00    1.0e+00    1.0e-04
b[sigma^2]
   1.0e-04

MCMC:
Saved 1500 samples from 3 chain(s), each ran 2500 iterations after a
  burn-in of 100 ; Thinning: 5


Null deviance:           704
Mean posterior deviance: 285

Marginal posterior inclusion probabilities and term importance:
                   P(gamma=1)     pi dim
u                          NA     NA   1
lin(sm1)                1.000  0.096   1 ***
sm(sm1)                 1.000  0.066   8 ***
lin(sm2)                0.999  0.028   1 ***
sm(sm2)                 0.976  0.016   8 ***
fct(f)                  1.000  0.579   2 ***
lin(lin1)               0.087 -0.002   1
sm(lin1)                0.037  0.001   9
lin(lin2)               0.997  0.029   1 ***
sm(lin2)                0.063  0.001   9
lin(lin3)               1.000  0.042   1 ***
sm(lin3)                0.039  0.000   9
lin(noise1)             0.053  0.002   1
sm(noise1)              0.028  0.000   9
lin(noise2)             0.019  0.000   1
sm(noise2)              0.039  0.000   8
lin(noise3)             0.025  0.000   1
sm(noise3)              0.039  0.000   8
fct(noise4)             0.078  0.001   3
lin(sm1):lin(sm2)       0.021  0.000   1
lin(sm1):sm(sm2)        0.067  0.000   7
lin(sm1):fct(f)         0.117 -0.003   2
lin(sm1):lin(lin1)      0.023  0.000   1
lin(sm1):sm(lin1)       0.056  0.000   7
sm(sm1):lin(sm2)        0.039  0.000   7
sm(sm1):sm(sm2)         0.120 -0.001  27
sm(sm1):fct(f)          0.068  0.000  13
sm(sm1):lin(lin1)       0.042  0.000   7
sm(sm1):sm(lin1)        0.053  0.000  28
```

```
lin(sm2):fct(f)          1.000  0.054   2 ***
lin(sm2):lin(lin1)       0.023  0.000   1
lin(sm2):sm(lin1)        0.054  0.000   8
sm(sm2):fct(f)           1.000  0.090  13 ***
sm(sm2):lin(lin1)        0.065  0.000   7
sm(sm2):sm(lin1)         0.195  0.000  28
fct(f):lin(lin1)         0.095  0.000   2
fct(f):sm(lin1)          0.137  0.001  14
*:P(gamma=1)>.25 **:P(gamma=1)>.5 ***:P(gamma=1)>.9
```

In most applications, the primary focus will be on the marginal posterior inclusion probabilities *P(gamma = 1)*. They are given along with a measure of term importance *pi* and the size of the associated coefficient batch *dim*. Term importance *pi* is defined as $\pi_j = \bar{\boldsymbol{\eta}}_j^\top \bar{\boldsymbol{\eta}}_{-1} / \bar{\boldsymbol{\eta}}_{-1}^T \bar{\boldsymbol{\eta}}_{-1}$, where $\bar{\boldsymbol{\eta}}_j$ is the posterior expectation of the linear predictor associated with the j^{th} term, and $\bar{\boldsymbol{\eta}}_{-1}$ is the linear predictor minus the intercept. Since $\sum_j^p \pi_j = 1$, the *pi* values provide a rough percentage decomposition of the (non-constant) linear predictor (Gu, 1992). Note that they can assume negative values as well. The summary shows that almost all true effects have a high posterior inclusion probability (i.e., *lin()* for *lin2*, *lin3*; *lin(),sm()* for *sm1*, *sm2*; *fct(f)*; and the interaction terms between *sm2* and *f*). All the terms associated with noise variables and the superfluous smooth terms for *lin1*, *lin2*, *lin3* as well as the superfluous interaction terms have a very low posterior inclusion probability. The small linear influence of *lin1* has not been recovered.

Figure 3.1 shows an excerpt from the second part of the *summary* output, which summarizes the posterior of the vector of inclusion indicators $\boldsymbol{\gamma}$. The table shows the different configurations of $P(\gamma_j = 1) > .5, j = 1, \dots, p$ sorted by relative frequency, i.e., the models visited by the sampler sorted by decreasing posterior support. For this simulated data, the posterior is concentrated strongly on the (almost) true model missing the small linear effect of *lin1*.

3.3.2. Visualization

spikeSlabGAM offers automated visualizations for model terms and their interactions, implemented with ggplot2 (Wickham, 2009). By default, the posterior mean of the linear predictor associated with each covariate (or combination of covariates if the model contains interactions) along with (pointwise) 80% credible intervals is shown. Figure 3.2 shows the estimated effects for *m1*.

Plots for specific terms can be requested with the *label* argument, Figures 3.3 and 3.4 show code snippets and their output for $f(\texttt{sm1})$ and $f(\texttt{sm2}, \texttt{f})$. The fits are quite close to the truth despite the heavy-tailed errors and the many noise terms included in the model. Full disclosure: The code used to render Figures 3.3 and 3.4 is a little more intricate than the code snippets we

```
Posterior model probabilities (inclusion threshold = 0.5 ):
                       1     2     3     4     5     6     7     8
prob.:             0.306 0.063 0.035 0.027 0.026 0.024 0.021 0.017
lin(sm1)               x     x     x     x     x     x     x     x
sm(sm1)                x     x     x     x     x     x     x     x
lin(sm2)               x     x     x     x     x     x     x     x
sm(sm2)                x     x     x     x     x     x     x     x
fct(f)                 x     x     x     x     x     x     x     x
lin(lin1)                                                x
sm(lin1)
lin(lin2)              x     x     x     x     x     x     x     x
sm(lin2)
lin(lin3)              x     x     x     x     x     x     x     x
sm(lin3)
lin(noise1)
sm(noise1)
lin(noise2)
sm(noise2)
lin(noise3)
sm(noise3)
fct(noise4)
lin(sm1):lin(sm2)
lin(sm1):sm(sm2)                                                 x
lin(sm1):fct(f)                                    x
lin(sm1):lin(lin1)
lin(sm1):sm(lin1)
sm(sm1):lin(sm2)
sm(sm1):sm(sm2)                              x
sm(sm1):fct(f)
sm(sm1):lin(lin1)
sm(sm1):sm(lin1)
lin(sm2):fct(f)        x     x     x     x     x     x     x     x
lin(sm2):lin(lin1)
lin(sm2):sm(lin1)
sm(sm2):fct(f)         x     x     x     x     x     x     x     x
sm(sm2):lin(lin1)
sm(sm2):sm(lin1)             x
fct(f):lin(lin1)                                           x
fct(f):sm(lin1)                    x
cumulative:        0.306 0.369 0.405 0.431 0.457 0.481 0.503 0.519
```

Figure 3.1.: Excerpt of the second part of the output returned by `summary.spikeSlabGAM`, which tabulates the configurations of $P(\gamma_j = 1) > .5$ with highest posterior probability. In the example, the posterior is very concentrated in the true model without `lin1`, which has a posterior probability of 0.31. The correct model that additionally includes `lin1` (column 6) has a posterior probability of 0.02.

```
R> plot(m)
```

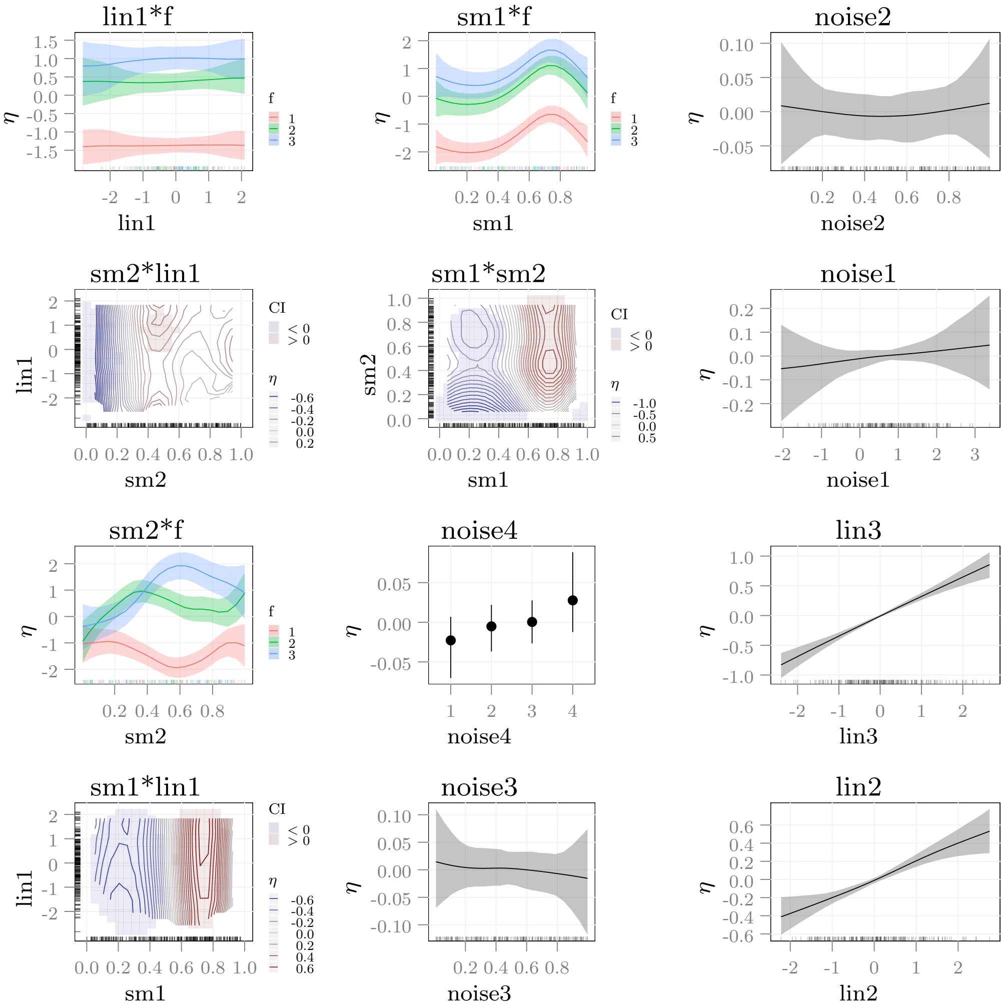


Figure 3.2.: Posterior means and pointwise 80% credible intervals for `m1`. Interaction surfaces of two numerical covariates are displayed as color coded contour plots, with regions in which the credible interval does not overlap zero marked in blue ($\eta < 0$) or red ($\eta > 0$). Each panel contains a marginal rug plot that shows where the observations are located. Note that the default behavior of `plot.spikeSlabGAM` is to cumulate all terms associated with a covariate or covariate combination. In this example, the joint effects of the first 4 covariates *`sm1`*, *`sm2`*, *`f`* and *`lin1`* and the sums of the `lin-` and `sm-`terms associated with *`lin2`*, *`lin3`*, *`noise1`*, *`noise2`* and *`noise3`* are displayed. All effects of the noise variables are ≈ 0, note the different scales on the vertical axes. Vertical axes can be forced to the same range by setting option *`commonEtaScale`*.

show, but the additional code only affects details (font and margin sizes and the arrangement of the panels).

```
R> plot(m, labels = c("lin(sm1)", "sm(sm1)"), cumulative = FALSE)
R> trueFsm1 <- data.frame(truth = fsm1 - mean(fsm1), sm1 = sm1)
R> plot(m, labels = "sm(sm1)", ggElems = list(geom_line(aes(x = sm1,
+       y = truth), data = trueFsm1, linetype = 2)))
```

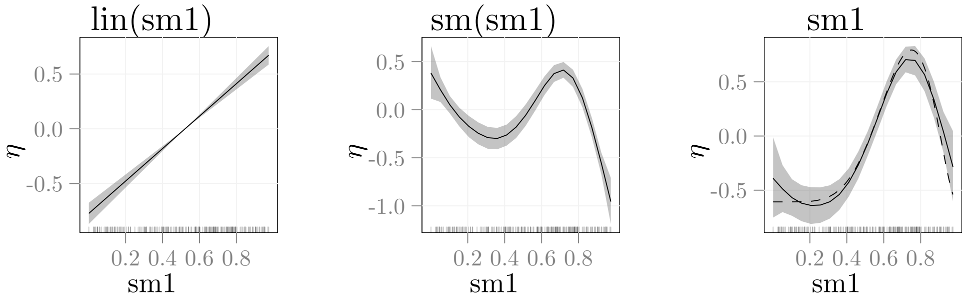


Figure 3.3.: Posterior means and pointwise 80% credible intervals for f(`sm1`) in `m1`. Left and middle panel show the separate `lin()`- and `sm()`-terms returned by the first call to *`plot`*, right panel shows their sum. True shape of f(`sm1`) added as a dashed line with the `ggElems` option of `plot.spikeSlabGAM`.

3.3.3. Assessing convergence

`spikeSlabGAM` uses the convergence diagnostics implemented in `R2WinBUGS` (Sturtz, Ligges, and Gelman, 2005). The function *`ssGAM2Bugs()`* converts the posterior samples for a *`spikeSlabGAM`*-object into a *`bugs`*-object, for which graphical and numerical convergence diagnostics are available via *`plot`* and *`print`*. Note that not all cases of non-convergence should be considered problematic, e.g., if one of the chains samples from a different part of the model space than the others, but has converged on that part of the parameter space.

3.3.4. Example: Diabetes in Pima women

We use the time-honored Pima Indian Diabetes dataset as an example for real non-Gaussian data: This dataset from the UCI repository (Asuncion and Newman, 2007) is provided in package `mlbench` (Leisch and Dimitriadou, 2010) as *`PimaIndiansDiabetes2`*. We remove two columns with a large number of missing values and use the complete measurements of the remaining 7

```
R> trueFsm2f <- data.frame(truth = fsm2f - mean(fsm2f),
+       sm2 = sm2, f = f)
R> plot(m, labels = "sm(sm2):fct(f)", ggElems =
+       list(geom_line(aes(x = sm2, y = truth, colour = f),
+            data = trueFsm2f, linetype = 2)))
```

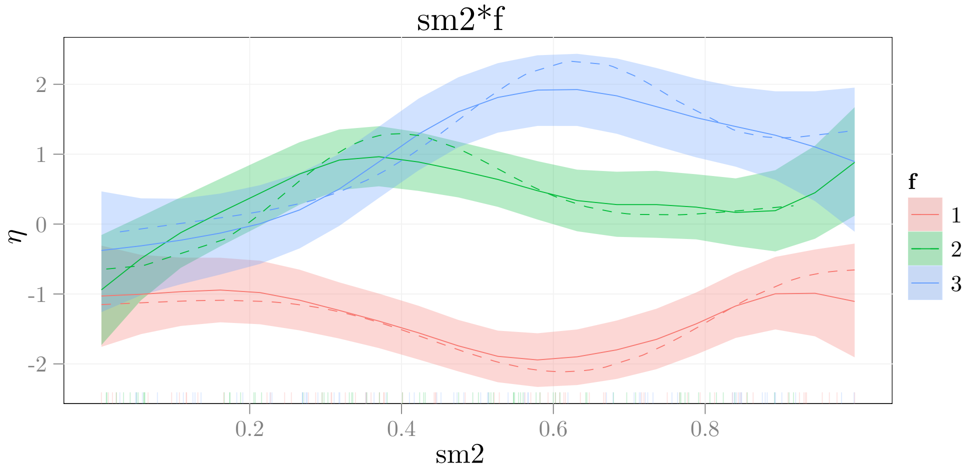


Figure 3.4.: Posterior means and pointwise 80% credible intervals for $f($`sm2`,`f`$)$ in `m1`. True shape of $f($`sm2`$|$`f`$)$ added as dashed line for each level of `f`.

covariates and the response (diabetes Yes/No) for 524 women to estimate the model. We set aside 200 observations as a test set:

```
R> data("PimaIndiansDiabetes2", package = "mlbench")
R> pimaDiab <- na.omit(PimaIndiansDiabetes2[, -c(4, 5)])
R> pimaDiab <- within(pimaDiab, {
+      diabetes <- 1 * (diabetes == "pos")
+  })
R> set.seed(1109712439)
R> testInd <- sample(1:nrow(pimaDiab), 200)
R> pimaDiabTrain <- pimaDiab[-testInd, ]
```

Note that *spikeSlabGAM()* always expects a dataset without any missing values and responses between 0 and 1 for binomial models.

We increase the length of the burn-in phase for each chain from 100 to 500 iterations and run 4 parallel chains for an additive main effects model (if `multicore` (Urbanek, 2010) or `snow` (Tierney, Rossini, Li, and Sevcikova, 2010) are installed, the chains will be run in parallel):

```
R> mcmc <- list(nChains = 4, chainLength = 1000, burnin = 500,
+      thin = 5)
R> m0 <- spikeSlabGAM(diabetes ~ pregnant + glucose + pressure +
+      mass + pedigree + age, family = "binomial", data = pimaDiabTrain,
+      mcmc = mcmc)
```

We compute the posterior predictive means for the test set, and request a summary of the fitted model:

```
R> pr0 <- predict(m0, newdata = pimaDiab[testInd, ])
R> print(summary(m0), printModels = FALSE)

Spike-and-Slab STAR for Binomial data

Model:
diabetes ~ (lin(pregnant) + sm(pregnant)) + (lin(glucose) + sm(glucose)) +
    (lin(pressure) + sm(pressure)) + (lin(mass) + sm(mass)) +
    (lin(pedigree) + sm(pedigree)) + (lin(age) + sm(age))
524 observations; 58 coefficients in 13 model terms.

Prior:
 a[tau]  b[tau]    v[0]    a[w]    b[w]
5.0e+00 2.5e+01 2.5e-04 1.0e+00 1.0e+00

MCMC:
Saved 4000 samples from 4 chain(s), each ran 5000 iterations after a
  burn-in of 500 ; Thinning: 5
P-IWLS acceptance rates: 0.9 for alpha; 0.64 for xi.

Null deviance:           676
Mean posterior deviance: 471

Marginal posterior inclusion probabilities and term importance:
```

```
               P(gamma=1)     pi dim
u                      NA     NA   1
lin(pregnant)       0.013  0.001   1
sm(pregnant)        0.023  0.000   8
lin(glucose)        1.000  0.482   1 ***
sm(glucose)         0.026  0.000   9
lin(pressure)       0.012  0.000   1
sm(pressure)        0.012  0.000   9
lin(mass)           1.000  0.239   1 ***
sm(mass)            0.933  0.065   9 ***
lin(pedigree)       0.013  0.000   1
sm(pedigree)        0.218 -0.002   8
lin(age)            0.486  0.033   1   *
sm(age)             1.000  0.182   8 ***
*:P(gamma=1)>.25 **:P(gamma=1)>.5 ***:P(gamma=1)>.9
```

spikeSlabGAM selects nonlinear effects for *age* and *mass* and a linear trend in *glucose* (and with fairly weak support for a nonlinear effect of *pedigree*). mboost*::gamboost* ranks the variables very similarly, based on the relative selection frequencies of the associated baselearners:

```
R> b <- gamboost(as.factor(diabetes) ~ pregnant + glucose +
+      pressure + mass + pedigree + age, family = Binomial(),
+      data = pimaDiabTrain)[300]
R> aic <- AIC(b, method = "classical")
R> prB <- predict(b[mstop(aic)], newdata = pimaDiab[testInd,
+      ])
```

```
R> summary(b[mstop(aic)])$selprob
```

```
    bbs(mass, df = dfbase)  bbs(glucose, df = dfbase)
                  0.290323                   0.266129
     bbs(age, df = dfbase) bbs(pedigree, df = dfbase)
                  0.209677                   0.120968
bbs(pregnant, df = dfbase) bbs(pressure, df = dfbase)
                  0.072581                   0.040323
```

Finally, we compare the deviance on the test set for the two fitted models:

```
R> dev <- function(y, p) {
+      -2 * sum(dbinom(x = y, size = 1, prob = p, log = T))
+  }
R> c(spikeSlabGAM = dev(pimaDiab[testInd, "diabetes"], pr0),
+      gamboost = dev(pimaDiab[testInd, "diabetes"], plogis(prB)))
```

```
spikeSlabGAM     gamboost
      181.01       194.79
```

So it seems like *spikeSlabGAM*'s model averaged predictions are a little more accurate than the predictions returned by *gamboost* in this case.

We can check the sensitivity of the results to the hyperparameters and refit the model with a larger v_0 to see if/how the results change:

```
R> hyper1 <- list(gamma = c(v0 = 0.005))
R> m1 <- spikeSlabGAM(diabetes ~ pregnant + glucose + pressure +
+      mass + pedigree + age, family = "binomial", data = pimaDiabTrain,
+      mcmc = mcmc, hyperparameters = hyper1)
R> pr1 <- predict(m1, newdata = pimaDiab[testInd, ])

R> print(summary(m1), printModels = FALSE)

Spike-and-Slab STAR for Binomial data

Model:
diabetes ~ (lin(pregnant) + sm(pregnant)) + (lin(glucose) + sm(glucose)) +
    (lin(pressure) + sm(pressure)) + (lin(mass) + sm(mass)) +
    (lin(pedigree) + sm(pedigree)) + (lin(age) + sm(age))
524 observations; 58 coefficients in 13 model terms.

Prior:
a[tau] b[tau]   v[0]   a[w]   b[w]
 5.000 25.000  0.005  1.000  1.000

MCMC:
Saved 4000 samples from 4 chain(s), each ran 5000 iterations after a
  burn-in of 500 ; Thinning: 5
P-IWLS acceptance rates: 0.85 for alpha; 0.64 for xi.

Null deviance:           676
Mean posterior deviance: 459

Marginal posterior inclusion probabilities and term importance:
              P(gamma=1)     pi dim
u                     NA     NA   1
lin(pregnant)      0.067  0.004   1
sm(pregnant)       0.079 -0.001   8
lin(glucose)       1.000  0.453   1 ***
sm(glucose)        0.082  0.000   9
lin(pressure)      0.101 -0.009   1
sm(pressure)       0.066  0.000   9
lin(mass)          1.000  0.238   1 ***
sm(mass)           0.959  0.064   9 ***
lin(pedigree)      0.148  0.009   1
sm(pedigree)       0.253  0.004   8   *
lin(age)           0.952  0.089   1 ***
sm(age)            0.993  0.150   8 ***
*:P(gamma=1)>.25 **:P(gamma=1)>.5 ***:P(gamma=1)>.9

R> (dev(pimaDiab[testInd, "diabetes"], pr1))

[1] 177.21
```

The selected terms are very similar, and the prediction is slightly more accurate (predictive deviance for `m0` was 181.01).

4. Simulation Studies and Application Results for `spikeSlabGAM`

4.1. Simulation studies

This section summarizes results from tests of `spikeSlabGAM` on simulated data. Section 4.1.1 investigates the adaptive shrinkage properties of the proposed prior structures. Section 4.1.2 shows that the proposed parameter expansion with multiplicative redundant parameters can improve sampling behavior for settings in which the posterior of the regression coefficients contains strong correlations. Sections 4.1.3 and 4.1.4 investigate model selection and estimation performance for models with random intercepts and smooth functions, respectively. Section 4.1.5 describes results for additive models of some complexity for both Gaussian and Poisson responses and compares the performance of our approach to the performances of other recently suggested algorithms.

We introduce some additional notation for the simulation of Gaussian data: For a given data-generating process (DGP) that generates a random design matrix $\boldsymbol{X}$ and a (fixed or random) vector of coefficients $\boldsymbol{\beta}$, let $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}$ denote the "true" linear predictor. For responses with $\boldsymbol{y} = \boldsymbol{\eta} + \varepsilon$, the difficulty level of estimating both $\boldsymbol{\beta}$, and, consequently, $\boldsymbol{\eta}$ is determined mostly by the ratio between the systematic variability that can be quantified as the observed variability of $\boldsymbol{\eta}$, i.e. the "signal", and the unsystematic variability introduced by the Gaussian error terms $\boldsymbol{\epsilon}$, the "noise". Let $\text{sd}_\eta = \sqrt{\sum_i^n(\eta_i - \bar{\eta})^2/n}$ and define the signal-to-noise ratio $\text{SNR} = n\,\text{sd}^2_\eta \,/\, \sum_i^n \varepsilon_i^2$. For a given value of SNR and realization of $\boldsymbol{\eta}$, responses $\boldsymbol{y}$ are then generated via $y_i \sim N\left(\eta_i, \text{sd}^2_\eta \,/\, \text{SNR}\right)$.

4.1.1. Adaptive shrinkage

We investigate the shrinkage properties of the proposed prior structures in a simple setting. The following describes the data-generating process:

- $n = 20, 50, 100$ observations
- $\beta = (.1, .2, .3, \ldots, 1), p = 10$
- signal-to-noise ratio SNR$= 0.5, 2$

- covariates $\boldsymbol{x}_j$ are independent, with $\boldsymbol{x}_j \sim U[-2,2]$ and enter the model scaled to have mean 0 and standard deviation .5.

- 100 replications per setting

We want to compare the shrinkage properties of the posterior means from `spikeSlabGAM` with those of the horseshoe prior (HS) as implemented in R package `monomvn` (Gramacy, 2010) and the LASSO estimator (L1) as implemented in R package `lasso2` (Lokhorst, Venables, Turlach, and Maechler, 2009). The horseshoe prior (a scale mixture of normals with a scaled half-Cauchy mixing distribution, where the scale of the mixing distribution is itself half-Cauchy distributed), has recently been shown to have excellent adaptive shrinkage properties (Carvalho et al., 2010) and we use its behavior as a reference for good adaptive shrinkage properties, while the LASSO estimators serve as a reference for a shrinkage estimator without adaptivity.

Figure 4.1 shows the median and the inter-quartile ranges of the posterior means of the estimated coefficients over the 100 replications for each combination of the different numbers of observations n and the signal-to-noise ratios SNR. We compare models with (peNMIG) and without (NMIG) the redundant multiplicative parameter expansion with $(a_\tau, b_\tau, v_0) = (5, 25, 0.00025)$ or $(5, 50, 0.005)$.

Note that the frequentist LASSO (L1, in brown) performs about the same amount of regularization in all of the settings – all six approaches overshrink the larger coefficients for $N = 20$ and $N = 50$, SNR$= 0.5$; LASSO less so than the Bayesian approaches. However, as more information from the data becomes available with increasing N and SNR, the Bayesian approaches (NMIG, peNMIG, HS) perform less regularization, since the likelihood contribution of the posterior increasingly dominates the prior contribution to the posterior. This is visible especially for the bottom right panel with $N = 100$ and SNR$= 2$.

Adaptive shrinkage in the sense of strong regularization of smaller coefficients (i.e. $\beta \leq 0.5$) and simultaneously weak shrinkage for large coefficients (i.e. $\beta \geq 0.8$) is observable only for $N = 50, 100$. For $N = 20$, posterior means for peNMIG with $(a_\tau, b_\tau) = (5, 50)$ and $v_0 = 0.005$ are closest to those returned by the horseshoe-prior model. We observe no systematic differences between the shrinkage properties of NMIG and peNMIG for $v_0 = .005$. Estimates and inclusion probabilities (see Figure 4.2) for the larger coefficients are much smaller for the NMIG model. We also note that inclusion probabilities for peNMIG seem to be somewhat less sensitive to the different hyperparameters than for NMIG. Shrinkage of the smaller coefficients is more pronounced for smaller v_0 and τ^2 (red and green symbols) without a corresponding increase in estimation bias for the larger coefficients, at least for settings with enough data (i.e. $n = 50$, SNR$= 2$ and $n = 100$). For settings with $n = 50$,

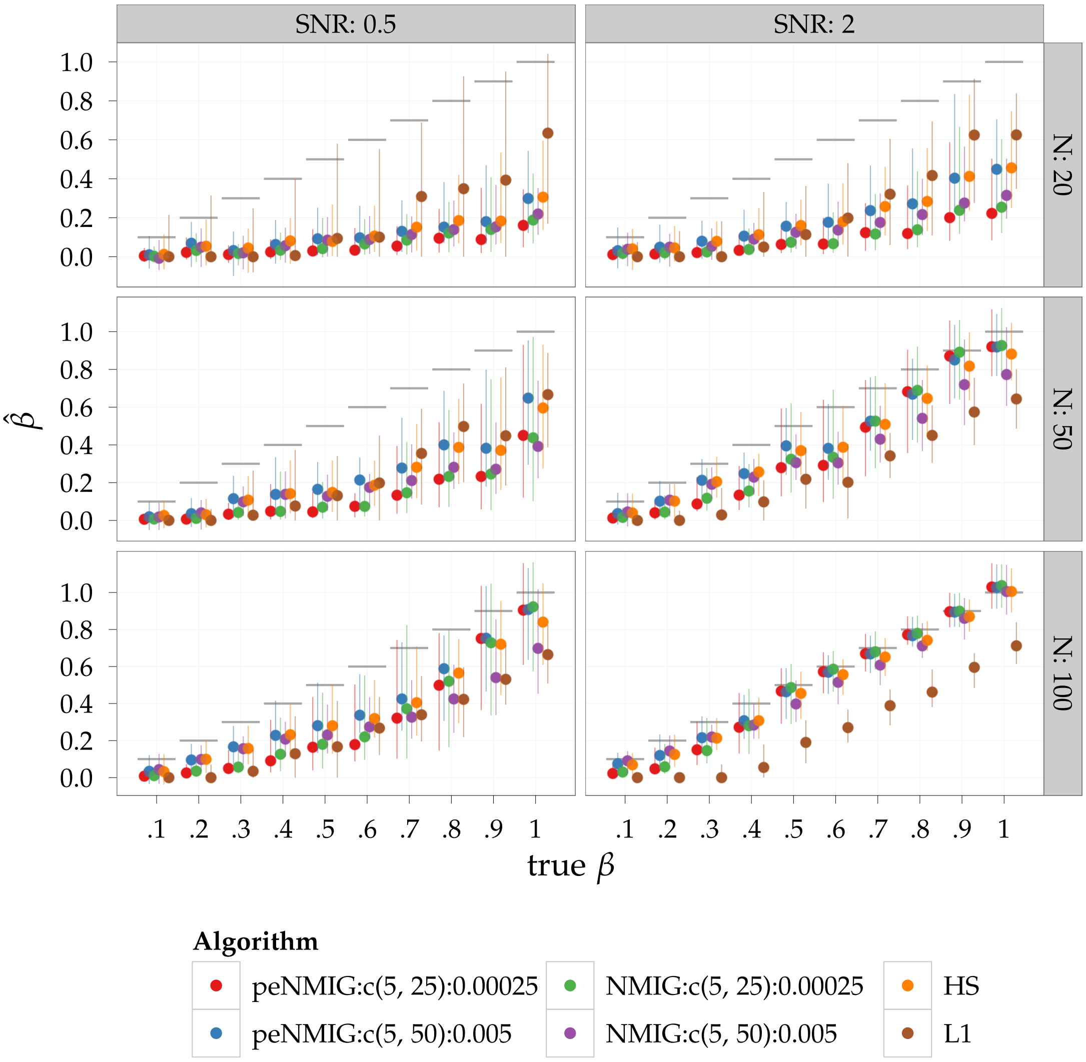


Figure 4.1.: Estimated coefficients (median & inter-quartile range) for different (pe)NMIG-prior settings, the horseshoe prior (HS) and the frequentist LASSO (L1). Fat dark gray horizontal bars show values of the true coefficients.

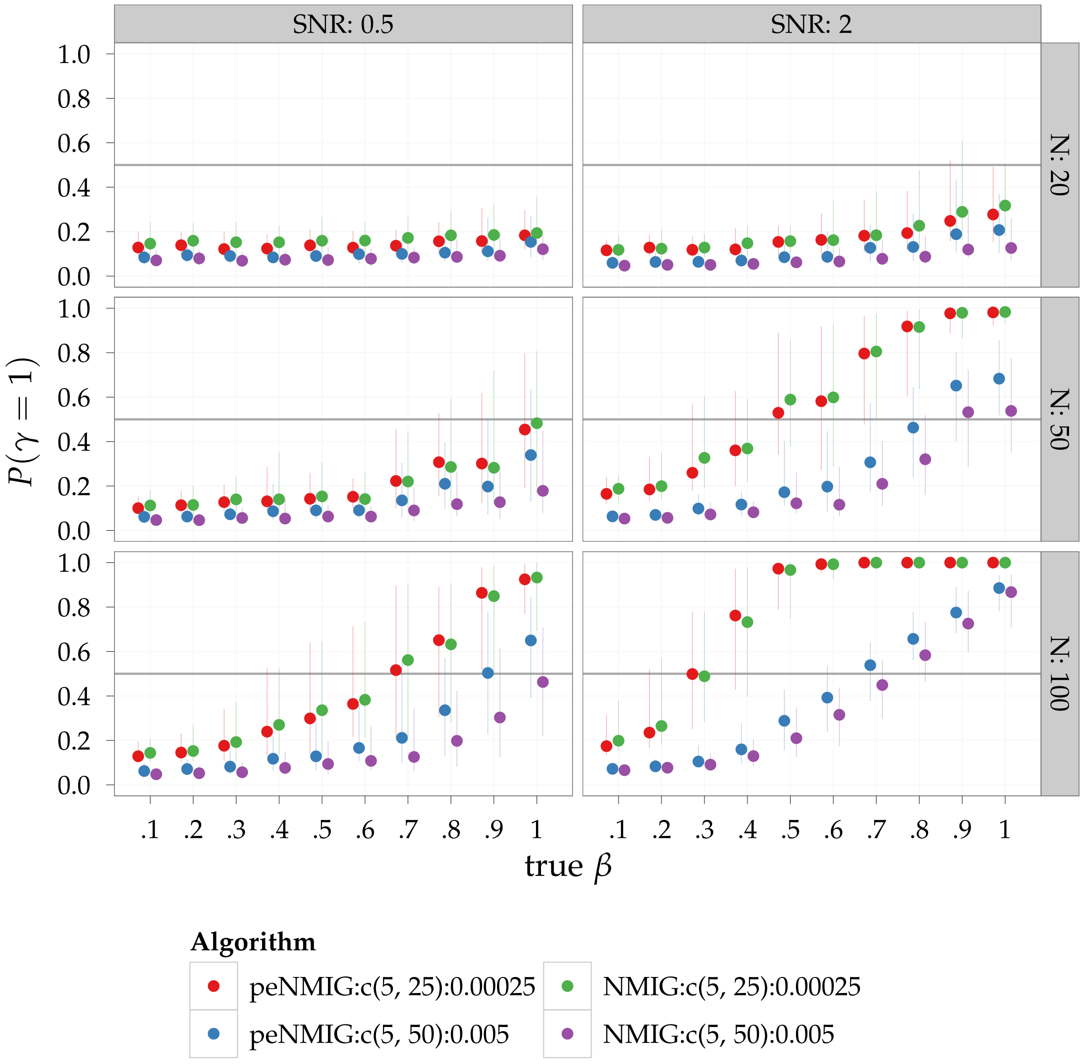


Figure 4.2.: Posterior means of $P(\gamma = 1)$ (median & inter-quartile range) for different NMIG-prior settings.

SNR$=2$ or $n=100$, larger v_0 and τ^2 NMIG models without parameter expansion (in purple) perform much worse. This is due to lower inclusion probabilities (see Figure 4.2). In general, we find that the `spikeSlabGAM` estimates are similar to the HS estimates.

Across all settings, estimation times for `spikeSlabGAM` for both NMIG and peNMIG were about one third to half of those for `monomvn`. In absolute terms, running 3000 iterations of the chains took between 0.16 and 0.36 seconds for `spikeSlabGAM` depending on n and whether parameter expansion was used, while `monomvn`'s horseshoe implementation took between 0.58 and 0.64 seconds on a modern desktop PC (Intel Core2 Quad Q9550 CPU with 2.83GHz).

Tail robustness and sparsity recovery

In order to compare the robustness of our approaches to large coefficient values relative to that of the horseshoe prior, we replicate the simulation study in Section 3.1. of Polson and Scott (2010). We simulate 100 datasets with $n=60$ observations and $p=40$ covariates. The covariates are independent standard normal variates. The true coefficient vector is 80% sparse, with the first 32 entries equal to zero (i.e. the "noise" component) and the remaining 8 drawn from a t-distribution with 3 degrees of freedom (i.e. the "signal" component). We simulate responses $\boldsymbol{y}$ with normal errors so that the signal-to-noise ratio is 2. Results are shown for prior settings $a_\tau=5, b_\tau=50, v_0=0.00025, a_w=b_w=1$ and the default settings for the horseshoe prior as implemented in `monomvn`. Figure 4.3 shows the mean square estimation errors (MSE) for posterior means of $\boldsymbol{\beta}$ separately for the noise (upper panel) and signal (lower panel) components of $\boldsymbol{\beta}$. MSE for the noise part is consistently higher for the

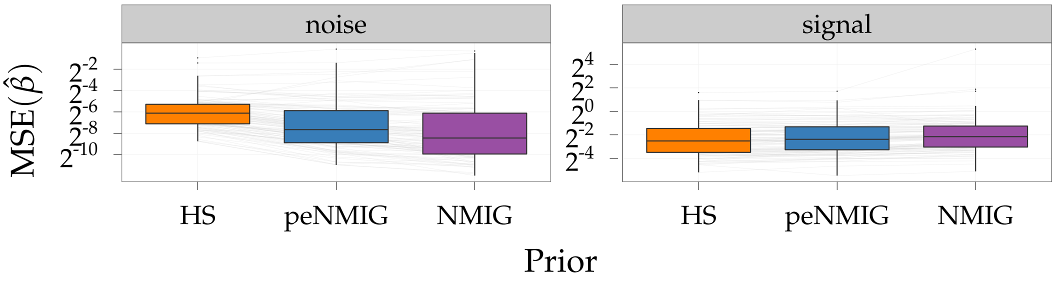


Figure 4.3.: Mean square estimation errors (MSE) for posterior means of $\boldsymbol{\beta}$. Upper panel shows MSE($\hat{\boldsymbol{\beta}}$) for the coefficients that are zero, lower panel shows MSE($\hat{\boldsymbol{\beta}}$) for the coefficients drawn from t_3. Dark grey lines connect values from the same replicates.

horseshoe estimates (average MSE-ratio is 2.8 compared to the `spikeSlabGAM`-estimates for peNMIG and 5.0 for NMIG), while the MSE for the signal part is slightly lower (average MSE-ratio: 0.94 for peNMIG and 0.83 for NMIG). These results show satisfactory tail robustness for both approaches comparable to that of the horseshoe prior and excellent sparsity recovery. As expected (see Section 2.2.4, Figures 2.4, 2.6), robustness is stronger for peNMIG than for NMIG. Sparsity recovery is very good for both of our approaches. We observed qualitatively similar results for signal-to-noise ratios 5 and .5 (not shown).

4.1.2. Sampling performance with parameter expansion

We investigate the approximate integrated autocorrelation times – defined as

$$\text{IAT}(\boldsymbol{x}) = \frac{1}{2} + \sum_{t=1}^{T} \hat{r}(t),$$

$\hat{r}(t)$ are the estimated auto correlations for lag t (Jackman, 2009)– for the regression coefficients and their estimation error in designs with strong correlations in the posterior distribution of $\boldsymbol{\beta}$. We generate random design matrices $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ so that $\Psi = \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}$ is a matrix with 1 on the diagonal and a constant ρ everywhere else, i.e. the correlations between all the OLS-estimators are equal to ρ. Specifically, $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Psi}^{-1/2}$, where $\boldsymbol{U}$ is an orthonormal matrix and $\boldsymbol{\Psi}^{-1/2}$ is the Cholesky root of $\boldsymbol{\Psi}^{-1}$. Responses $\boldsymbol{y}$ are then generated as

$$\boldsymbol{y} \sim N_n\left(\boldsymbol{\eta}, \frac{\text{sd}^2_{\eta}}{\text{SNR}}\boldsymbol{I}_n\right).$$

Regression coefficients $\boldsymbol{\beta}$ are set as an equidistant descending sequence of length 10 from 2 to .5 interspersed with zeroes, i.e. $\boldsymbol{\beta} = (2, 0, \dots, 0.\bar{6}, 0, 0.5, 0)'$ so that $p = 20$.

We use the following settings for our simulations:

- correlation of $\boldsymbol{\beta}_{OLS}$: $\rho = .9, .95$
- signal-to-noise-ratio SNR $= 1, 3$
- no of observations: $n = 50, 100$
- 100 replications for each setting

Figure 4.4 shows ratios between average integrated autocorrelation times for $\hat{\boldsymbol{\beta}}$ (top graph) and root mean square estimation error $\sqrt{||\hat{\beta} - \beta||_2^2}$ (middle graph) with and without parameter expansion for the different settings

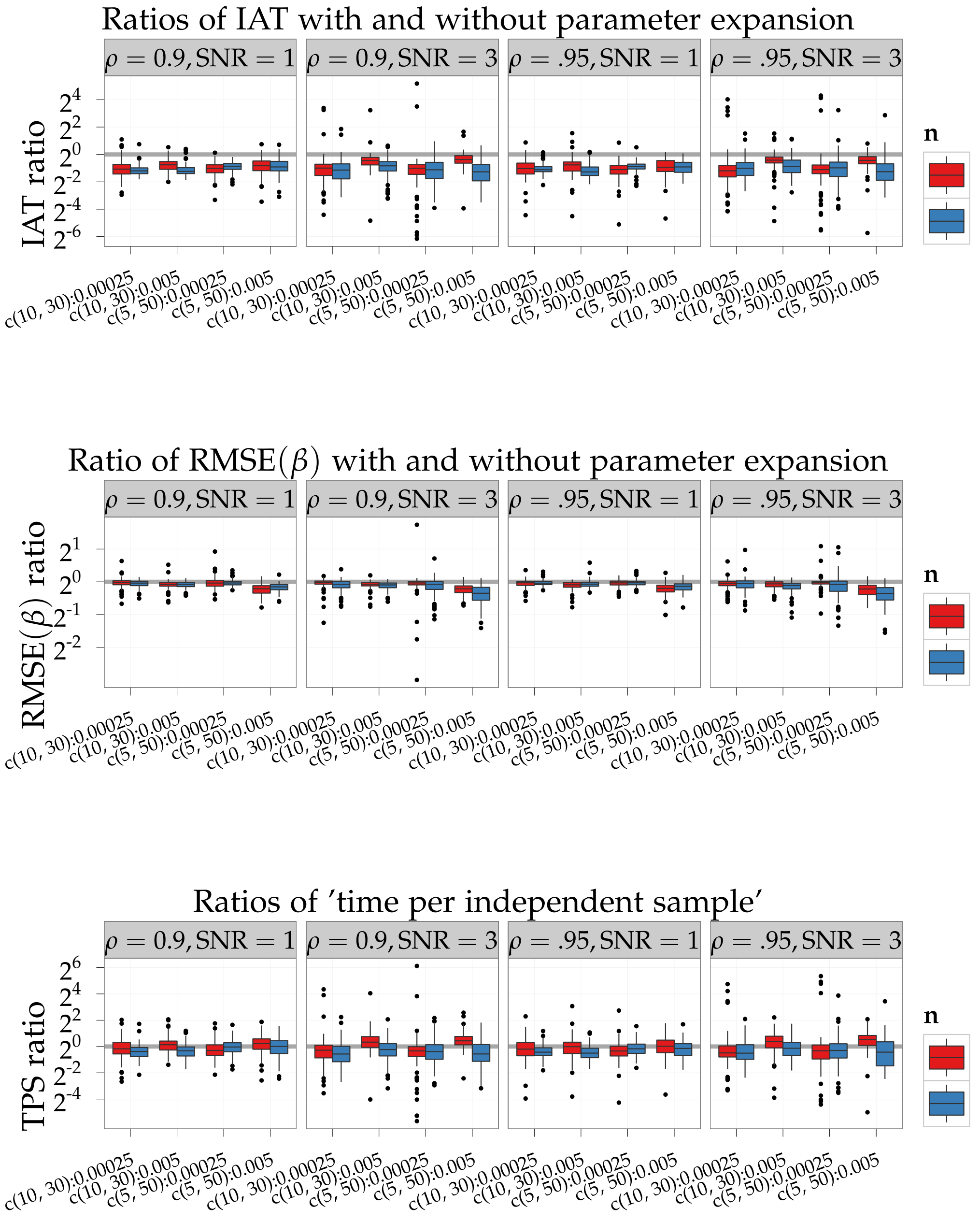


Figure 4.4.: Ratios of average integrated autocorrelation times for $\hat{\boldsymbol{\beta}}$ (upper graph), root mean square estimation error $\sqrt{\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2}$ (middle graph) and time per "independent" sample (bottom graph). Columns correspond to the settings of the data generating process (correlation and SNR). Boxplots contain the ratio between peNMIG and NMIG results for each replicate. Boxplots are grouped into the four different prior settings. Red boxplots correspond to results for $n = 100$ observations, blue for $n = 50$. Vertical axes are on binary log scale; fat gray horizontal line corresponds to a ratio of 1, i.e no change.

for the posterior correlation, the signal-to-noise ratio and hyperparameters (a_τ, b_τ) and v_0. Panels from left to right show results for correlation 0.9 with SNR 1 and SNR 3 followed by results for correlation 0.95 with SNR 1 and SNR 3. The simulation shows that the suggested parameter expansion improves mixing and reduces estimation error for all DGP settings and hyperparameter configurations, especially for higher SNR, smaller v_0, and low number of observations. Parameter expansion reduces estimated integrated autocorrelation times for $\boldsymbol{\beta}$ by a median factor of .49 for $n = 50$ and .57 for $n = 100$ and estimation error $\sqrt{\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2}$ by a median factor of .94 for $n = 50$ and .95 for $n = 100$. Because of the larger complexity of the sampler for peNMIG (see Section 3.2), the observed improvement in mixing is not large enough to translate into consistent reductions in computing time for $n = 100$: The bottom graph in Figure 4.4 shows that the time to generate a single "independent sample" (defined as the total run time of the sampler divided by the effective sample size, which is in turn the number of iterations of the chain divided by 2IAT (Jackman, 2009)) remains about the same in most settings, with median ratios of estimated time per independent sample of .80 for $n = 50$ and .98 for $n = 100$. Regression analyses of the simulation results with performance measures as dependent variables and second-degree interactions and main effects for the data-generating process (n, SNR, correlation) and the hyperparameters $((a_\tau, b_\tau), v_0)$ also show that using peNMIG increases the odds of correctly including a covariate in the model by a factor of 1.11, without a corresponding decrease in specificity. Accuracy increases by a factor of 1.04. Table 4.1 gives mean performance measures for the different settings and priors.

In summary, these results indicate that parameter expansion has the potential to improve mixing for difficult data situations dramatically, although this may not translate into relevant savings in computation time for larger data sets with many parameters due to the higher computational burden of sampling from the parameter expanded posterior. Parameter expansion also reduces estimation error and improves complexity recovery. Note that we did not investigate whether these advantages disappear if the sampler for the conventional NMIG model is allowed to run long enough to achieve a similar effective sample size as that of the parameter expanded model.

4.1.3. Random intercept models

This section summarizes simulation results on selecting and estimating random intercept coefficients for Gaussian, Poisson and binomial response. The basic data generating process for all types of response is

$$\boldsymbol{\eta} = \boldsymbol{x}\beta + \boldsymbol{Z}\boldsymbol{b}$$

		Parameter					
DGP	Prior	Expansion	Sensitivity	Specificity	IAT	$\sqrt{\text{MSE}(\hat{\boldsymbol{\beta}})}$	TPS $[ms]$
0.9:100:1	c(10, 30):0.00025	Yes	0.69	0.84	4.21	0.09	7.96
		No	0.70	0.80	8.18	0.09	8.50
	c(10, 30):0.005	Yes	0.68	0.86	1.79	0.09	3.36
		No	0.65	0.85	2.97	0.09	2.83
	c(5, 50):0.00025	Yes	0.60	0.93	4.36	0.10	8.22
		No	0.55	0.94	8.44	0.10	8.74
	c(5, 50):0.005	Yes	0.51	0.96	1.14	0.09	2.02
		No	0.37	0.98	2.18	0.11	1.85
0.9:100:3	c(10, 30):0.00025	Yes	0.93	0.95	3.40	0.04	6.31
		No	0.92	0.93	5.83	0.04	6.50
	c(10, 30):0.005	Yes	0.91	0.96	0.93	0.04	1.78
		No	0.89	0.97	1.53	0.04	1.58
	c(5, 50):0.00025	Yes	0.91	0.98	1.90	0.04	3.49
		No	0.89	0.97	4.74	0.04	5.50
	c(5, 50):0.005	Yes	0.77	1.00	0.77	0.04	1.40
		No	0.69	1.00	1.08	0.05	1.04
0.9:50:1	c(10, 30):0.00025	Yes	0.38	0.86	3.86	0.14	5.28
		No	0.34	0.85	8.62	0.15	6.76
	c(10, 30):0.005	Yes	0.38	0.87	1.68	0.14	2.29
		No	0.29	0.88	3.76	0.15	2.90
	c(5, 50):0.00025	Yes	0.19	0.97	4.40	0.16	6.04
		No	0.17	0.97	8.10	0.16	6.29
	c(5, 50):0.005	Yes	0.24	0.96	1.20	0.14	1.59
		No	0.13	0.99	2.38	0.16	1.79
0.9:50:3	c(10, 30):0.00025	Yes	0.77	0.89	4.11	0.07	5.84
		No	0.76	0.85	8.95	0.08	7.79
	c(10, 30):0.005	Yes	0.76	0.91	1.63	0.07	2.06
		No	0.72	0.90	3.20	0.08	2.74
	c(5, 50):0.00025	Yes	0.68	0.96	4.41	0.08	6.30
		No	0.59	0.97	9.08	0.09	7.67
	c(5, 50):0.005	Yes	0.60	0.98	1.14	0.08	1.55
		No	0.42	0.99	3.03	0.10	2.40
0.95:100:1	c(10, 30):0.00025	Yes	0.68	0.85	4.11	0.09	7.93
		No	0.66	0.80	8.17	0.10	8.66
	c(10, 30):0.005	Yes	0.67	0.86	1.63	0.09	2.96
		No	0.64	0.83	2.99	0.10	3.03
	c(5, 50):0.00025	Yes	0.60	0.95	3.25	0.09	6.15
		No	0.56	0.94	7.34	0.10	7.73
	c(5, 50):0.005	Yes	0.52	0.97	1.02	0.09	1.89
		No	0.39	0.97	2.30	0.11	2.24
0.95:100:3	c(10, 30):0.00025	Yes	0.95	0.93	3.04	0.04	5.69
		No	0.94	0.93	5.43	0.04	6.19
	c(10, 30):0.005	Yes	0.92	0.96	1.13	0.04	2.17
		No	0.90	0.96	1.66	0.04	1.87
	c(5, 50):0.00025	Yes	0.93	0.97	2.30	0.04	4.39
		No	0.91	0.98	4.11	0.04	4.40
	c(5, 50):0.005	Yes	0.78	1.00	0.75	0.04	1.45
		No	0.71	1.00	1.35	0.05	1.39
0.95:50:1	c(10, 30):0.00025	Yes	0.38	0.83	4.05	0.14	5.42
		No	0.35	0.83	8.34	0.15	6.94
	c(10, 30):0.005	Yes	0.37	0.84	1.67	0.14	2.24
		No	0.32	0.86	3.86	0.15	3.13
	c(5, 50):0.00025	Yes	0.18	0.96	4.63	0.16	6.23
		No	0.14	0.97	8.57	0.16	6.82
	c(5, 50):0.005	Yes	0.24	0.94	1.22	0.14	1.65
		No	0.10	0.98	2.43	0.16	1.94
0.95:50:3	c(10, 30):0.00025	Yes	0.80	0.87	4.44	0.07	5.90
		No	0.79	0.85	8.75	0.08	7.77
	c(10, 30):0.005	Yes	0.79	0.90	1.91	0.07	2.55
		No	0.75	0.88	3.26	0.08	2.67
	c(5, 50):0.00025	Yes	0.69	0.95	5.03	0.08	7.50
		No	0.59	0.96	9.93	0.09	8.57
	c(5, 50):0.005	Yes	0.62	0.98	1.25	0.07	1.72
		No	0.43	0.99	3.10	0.10	2.52

Table 4.1.: Means of sensitivity (ratio of included coefficients $\geq .5$), specificity (ratio of excluded coefficients $= 0$), integrated autocorrelation times, root mean square error for estimated coefficients and estimated times per independent sample (in milliseconds, on an AMD Opteron 270)

with an incidence matrix $\boldsymbol{Z}$ for a grouping factor and

$$\begin{aligned} x_i &\overset{\text{i.i.d.}}{\sim} U\left(0, \sqrt{12}\right),\ i = 1, \ldots, n \text{ so that } \operatorname{Var}(\boldsymbol{x}) = 1 \\ \beta &= 1 \\ \tilde{b}_g &\overset{\text{i.i.d.}}{\sim} t_\nu,\ g = 1, \ldots, \text{no. of groups}; \\ \boldsymbol{b} &= \sigma \frac{\tilde{\boldsymbol{b}} - \operatorname{mean}(\tilde{\boldsymbol{b}})}{\operatorname{sd}(\tilde{\boldsymbol{b}})} \end{aligned}$$

with all combinations of the following settings:

- $g = 10$ or 100 groups/subjects (i.e $\boldsymbol{b} \in \mathbb{R}^{10}$ or $\mathbb{R}^{100}$) for linear mixed models and $g = 10, 20, 50, 100$ groups for binomial and Poisson response
- with (on average) 5 or 20 observations for each group/subject for linear mixed models and (on average) 5 observations per group for binomial and Poisson response
- with degrees of freedom $\nu = 1$ or 20 (i.e. Cauchy or approximately Gaussian random effects)

We use scaled and centered random effects $\boldsymbol{b}$ so that the contribution of the random effects to the variability of the linear predictor is constant across replications for the same value of σ and for different values of ν. We compare misclassification rates and root mean square estimation error (RMSE) $\sqrt{\text{MSE}} = \|\hat{\boldsymbol{b}} - \boldsymbol{b}\|/g$ between various prior settings for our approach and mixed models fitted with `lme4` (Bates and Maechler, 2009) and tested with (restricted) likelihood ratio tests.

Linear mixed model

For the linear mixed model, we use

- signal-to-noise-ratio SNR = 1, 5
- random effects scale factor $\sigma = 0$, 0.0625, 0.125, 0.25, 0.5, 0.75, 1

and balanced data, in addition to the settings described above. We generate 100 data sets for each combination of settings.

Inclusion or exclusion of the random intercept term in the LMM is based on the p-value of an exact restricted likelihood ratio test (RLRT) for $H_0 : \sigma^2 = 0$ with nominal significance level $\alpha = .05$ as implemented in `RLRsim` (Scheipl, 2010a; Scheipl, Greven, and Küchenhoff, 2008). We consider the random intercept included in the `spikeSlabGAM`-models if the Rao-Blackwellized estimate

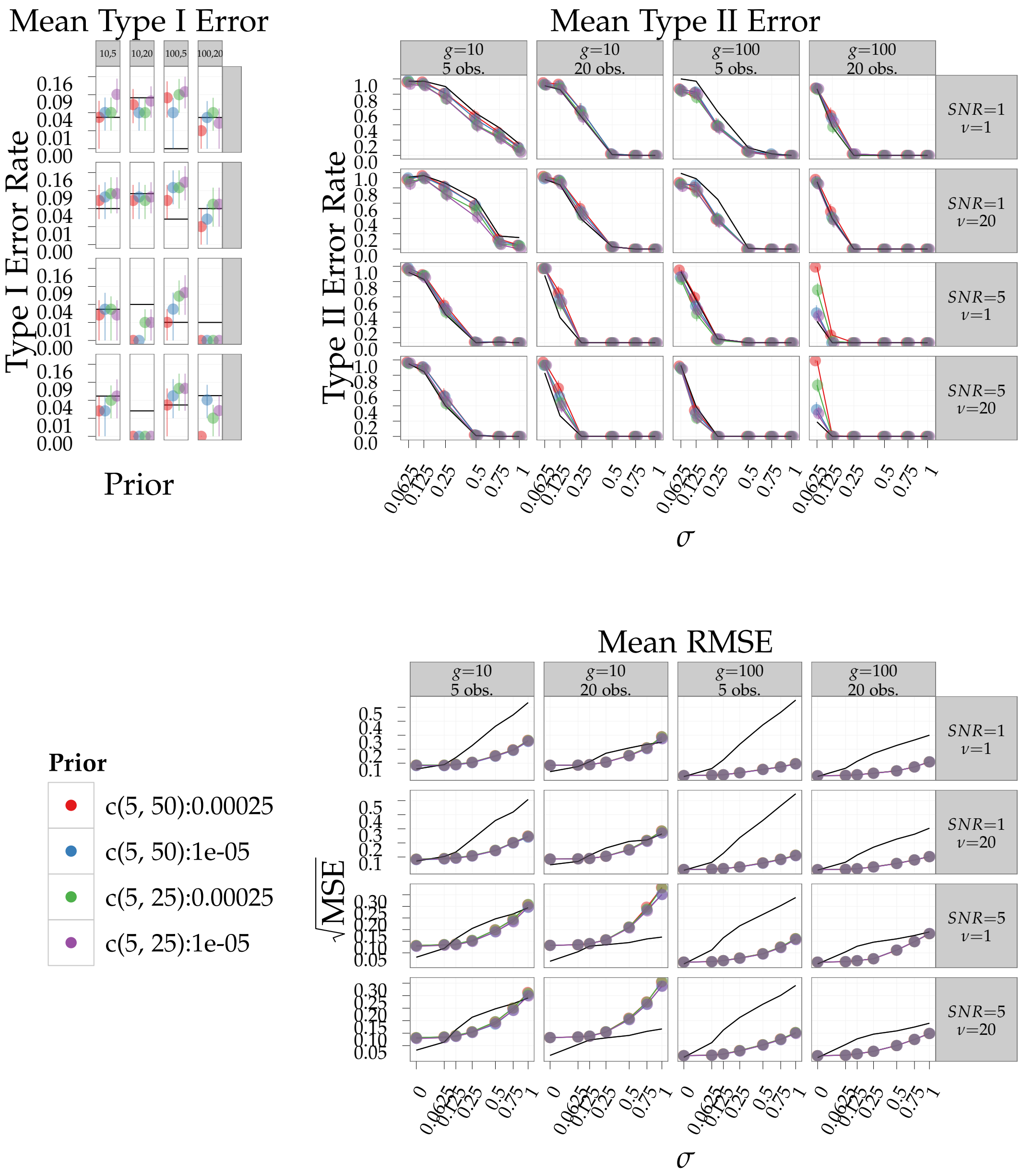


Figure 4.5.: Mean type I / type II error rates with 95% bootstrap CIs and $\sqrt{\text{MSE}}$ for linear mixed models with a random intercept. Rows correspond to the different combinations of SNR and degrees of freedom ν, top two rows are for SNR $= 1$. Columns correspond to the different combinations of number of groups/subjects and observations per group/subject, two rightmost columns are for 10 groups/subjects. Left graph gives type I error for $\sigma = 0$, right graph gives type II error rates for $\sigma > 0$. Graph on the lower right gives mean estimation error $\sqrt{\text{MSE}} = \|\hat{\boldsymbol{b}} - \boldsymbol{b}\|/g$. Solid black lines line give error rates and RMSE for the LMM (based on the p-value of a restricted LR-test with nominal $\alpha = .05$). Vertical axis for type I error is on $\sqrt{\ }$-scale. Error bars show 95% CIs for mean error rates.

of the posterior mean of $P(\gamma_b = 1)$ is greater than 0.5. Figure 4.5 shows error rates (top left: false positive or type I error for $\sigma = 0$, top right: false negative or type II error for $\sigma > 0$) and root mean square estimation errors for the random intercept model for Gaussian responses. Type I and type II error rates for the hyperparameter configurations considered here are mostly very close to those of the RLRT with nominal significance level $\alpha = .05$ (black lines) and very robust against the different hyperparameter configurations, especially for smaller sample sizes. As in the other simulations, a smaller v_0 (red, green symbols) yields less conservative models in most settings, because the threshold an effect has to cross before the associated hypervariance is sampled from the "slab" and not from the "spike" decreases. Counterintuitively, this effect seems to be largest for large data sets where one would expect the prior's influence to be smaller.

Estimation error for the frequentist LMM is markedly larger than for peNMIG for larger sample sizes when $\sigma > 0$, about the same for smaller sample sizes and remarkably stable across the different prior settings. Average estimation error for the frequentist LMM for $\sigma = 0$ is always lower because random effects can be estimated as *exactly* zero if $\hat{\sigma} = 0$ for the LMM while `spikeSlabGAM` only enforces strong shrinkage. Contrary to what we would have expected, estimation error for $\sigma = 0$ is not much lower for $v_0 = 10^{-5}$ despite the fact that it imposes stronger shrinkage than $v_0 = 0.00025$.

Mixed model with non-Gaussian response

For the generalized linear mixed models, binary responses $\boldsymbol{y}$ are generated from

$$y_i \sim B\left(n = 1, p = (1 + \exp(-\eta_i))^{-1}\right)$$

and Poisson responses are generated from

$$y_i \sim Po\left(\lambda = \exp(\eta_i)\right)$$

with

- random effects scale factor $\sigma = 0,\ 0.125, 0.25,\ 0.5,\ 0.75,\ 1$
- balanced design or unbalanced with relative group sizes drawn from a Dirichlet distribution with concentration parameter $\boldsymbol{\alpha} = (5, \ldots, 5)'$

and the other settings as described at the beginning of this section. Results are shown only for the unbalanced case with 5 observations per group. Increasing the number of observations per group and/or using balanced groups did not change the results in pilot runs (cf. results for the LMM in the preceding section) and the corresponding settings were omitted.

Inclusion or exclusion of the random intercept term in the GLMM is based on the p-value of a likelihood ratio test for $H_0 : \sigma^2 = 0$ with significance level $\alpha = 0.15$. The reference distribution for this test was determined by a parametric bootstrap for each dataset. We generate 100 data sets for each combination of settings. Figures 4.6 and 4.7 show error rates (top left: false positive or type I error for $\sigma = 0$, top right: false negative or type II error) and root mean square estimation errors for the random intercept model for Poisson and binomial responses, respectively.

For both binomial and Poisson response, type I error rates are large and increase with the number of groups. Type II error rates for binomial response remain essentially constant as the variance of the random effects increases and mostly remain above 20% even for fairly large values of the variance and regardless of the hyperparameter settings. For Poisson response, there is strong sensitivity of error rates and estimation error towards v_0, with 100% type I error for $g = 100$ and 80% type I error for $g = 50$ for $v_0 = 10^{-5}$. Type II error rates also decline much faster for $v_0 = 10^{-5}$ for $g = 10, 20$.

Estimation error for both Poisson and binomial response is mostly lower than that of the `lme4` fit, especially for larger random effect variances and larger v_0 and despite the better selection properties of smaller v_0.

We obtained similar results for balanced data and data with more than 5 observations per group.

Closer examination of the estimated inclusion probabilities reveals that the estimated inclusion probabilities for the settings with more than ten groups are usually between 40% and 70%, with no change in their distribution as the variance of the random intercept increases. The mixing of the indicator variables γ in any given chain is very poor for these large coefficient blocks: Chains that move from v_0 to 1 usually move back to v_0 immediately in the following iteration: Since the IWLS proposal does not yield sufficiently large steps to move the coefficient values to their more heavily shrunk value in one iteration, the indicator usually changes back to 1 immediately. Moves from 1 to v_0 occur rarely for more than 10 groups.

The simulation results for LMM and GLMM suggest that the model selection behavior for random effects is similar to that of the (restricted) likelihood ratio test for a broad variety of settings in the Gaussian case, but breaks down for non- Gaussian responses in the case of random effects with many levels. Estimation of the random effects is much better than that produced by the conventional ridge-type shrinkage of the frequentist mixed model with Gaussian random effects for almost all settings we considered.

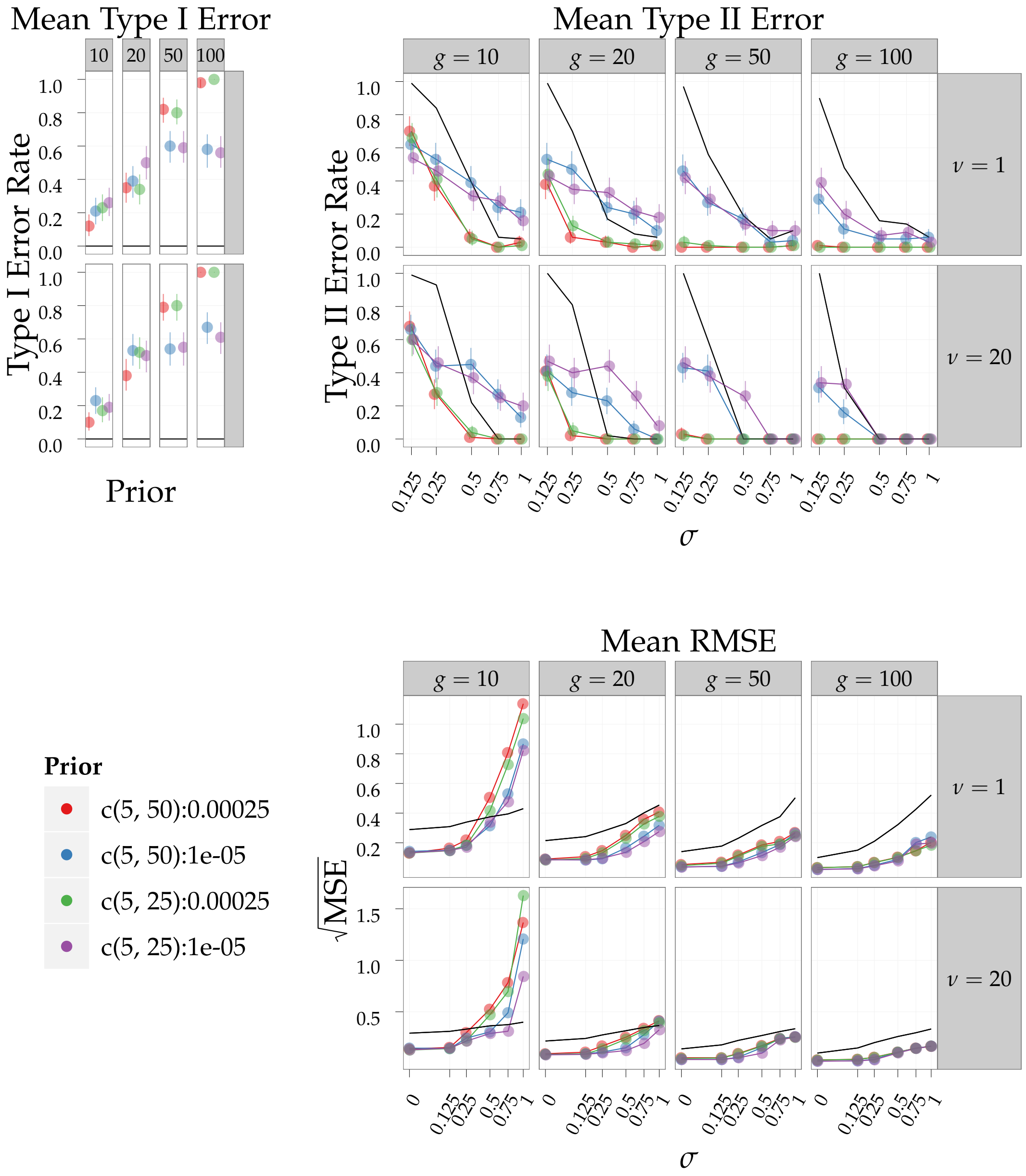


Figure 4.6.: Mean type I / type II error rates and $\sqrt{\text{MSE}}$ for mixed models with a random intercept and Poisson response. Rows correspond to the different degrees of freedom ν. Columns correspond to the different numbers of groups g. Left graph gives type I error for $\sigma = 0$, right graph gives type II error rates for $\sigma > 0$. Graph on the lower right gives mean estimation error $\sqrt{\text{MSE}} = \sqrt{\|\hat{\boldsymbol{b}} - \boldsymbol{b}\|^2}$. Solid black lines line give error rates and RMSE for the GLMM (based on the p-value of a bootstrap LR test with $\alpha = .15$). Vertical axis for type I error is on $\sqrt{\ }$-scale. Error bars show 95% CIs for mean error rates.

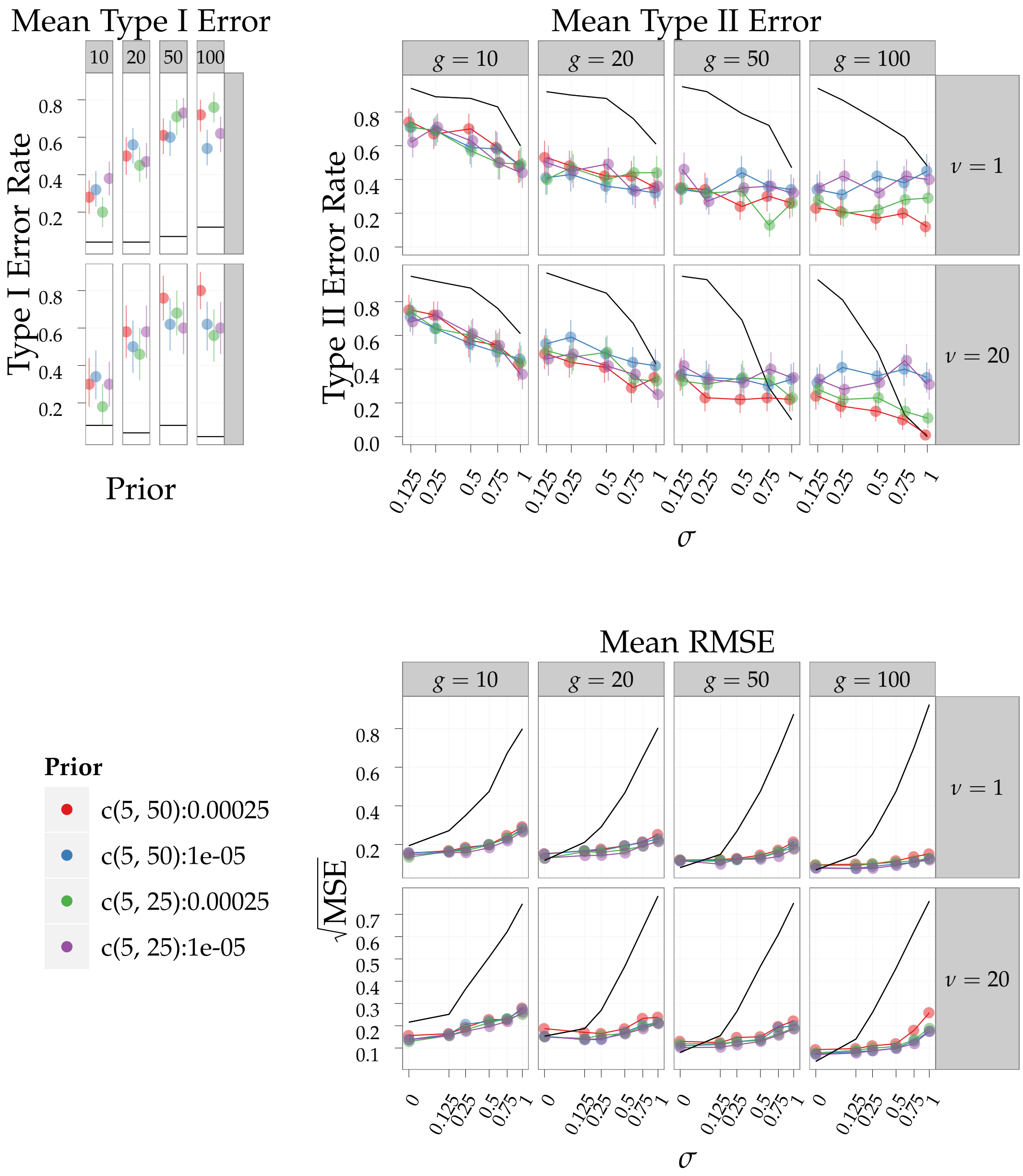


Figure 4.7.: Mean type I / type II error rates and $\sqrt{\text{MSE}}$ for mixed models with a random intercept and binary response. Rows correspond to the degrees of freedom ν. Columns correspond to the different number of groups g. Left graph gives type I error for $\sigma = 0$, right graph gives type II error rates for $\sigma > 0$. Graph on the lower right gives mean estimation error $\sqrt{\text{MSE}} = \sqrt{\|\hat{\boldsymbol{b}} - \boldsymbol{b}\|^2}$. Solid black lines line give error rates and RMSE for the GLMM (based on the p-value of a bootstrap LR test with $\alpha = .15$). Vertical axis for type I error is on $\sqrt{\ }$-scale, vertical axis for RMSE is on $\log_2$-scale. Error bars show 95% CIs for mean error rates.

4.1.4. Univariate smoothing

We investigate the properties of the peNMIG prior in terms of function selection for both randomly generated and fixed functions for Gaussian and binary binomial responses.

We compare inclusion probabilities and misclassification rates for peNMIG with various hyperparameter configurations to boosting with separate base learners for the linear and smooth parts of the function with `mboost` and to additive models (AM) in mixed model representation fitted with `amer` (Scheipl, 2010c). Inclusion or exclusion of a smooth term for the latter is based on the p-value of an exact finite sample restricted likelihood ratio test (RLRT) for $H_0 : \sigma^2 = 0$ with $\alpha = .05$ as implemented in `RLRsim` (Scheipl, 2010a; Scheipl et al., 2008) in the Gaussian case. A parametric bootstrap LRT is used for binomial responses. Ten-fold cross validation on the training data is employed to determine the optimal stopping iteration for `mboost` and a baselearner is included in the model if it is selected in at least half of the cross-validation runs up to the stopping iteration. Smooth terms are included in the `spikeSlabGAM`-models if the Rao-Blackwellized posterior mean of $P(\gamma = 1)$ is greater than 0.5 (cf. Section 3.2.3).

Randomly generated functions

We investigate the properties of our approach first on data from a very basic data-generating process for a simple spline model:

- $\boldsymbol{\eta} = \boldsymbol{x} + \boldsymbol{Z}(\boldsymbol{x})\boldsymbol{b}$; $\boldsymbol{Z}(\boldsymbol{x})$ is the penalized part of a B-spline basis for covariate $\boldsymbol{x}$ with a difference penalty of order 2.
- $\boldsymbol{b} \sim \sigma N(\boldsymbol{\mu}, \boldsymbol{I}_d)$, $\boldsymbol{\mu}$ is drawn from $\{-1, 1\}^d$.

We use the following settings for the simulation:

- number of observations: $n = 50, 500$
- signal-to-noise-ratio SNR $= 0.5, 5$
- dimension of spline basis: $d_s = 5, 20$
- degree of nonlinearity: $\sigma^2 = 0, 0.125, 0.25, 0.375, 0.5$
- 100 replications

For $\sigma^2 = 0$, the function to be estimated is linear, so the correct model does not include a smooth term. Results for this data generating process are shown in Figure 4.9. Figure 4.8 shows 10 realizations of simulated functions $\boldsymbol{x} + \boldsymbol{Z}(\boldsymbol{x})\boldsymbol{b}$ for the various settings.

Fixed functions

We also investigate the properties of our approach with a data-generating process (DGP) based on non-random functions:

- $\eta = \boldsymbol{x} + \sigma f(x)$
- $f(x) = \begin{cases} (2x - 1.5)^2/3 & \text{(quadratic)} \\ (\pi \sin(2\pi x))/11 & \text{(sinus)} \\ (\phi((x-0.2)/0.12) - \phi((x-0.7)/0.055)) & \text{(bumpy)} \end{cases}$
  $\phi(\cdot)$ is the standard normal density.

We use the following settings for the simulation:

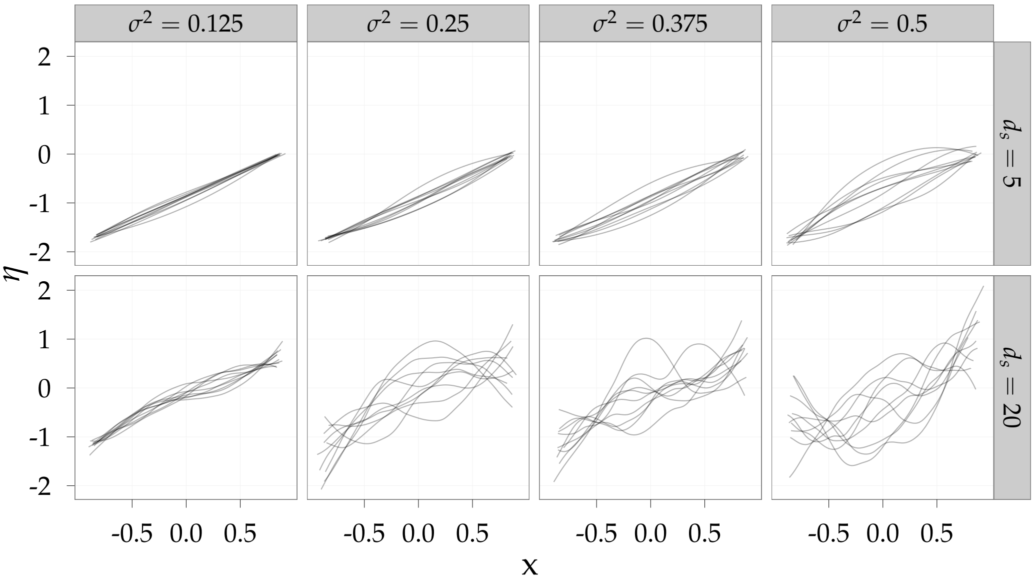
- number of observations: $n = 50, 500$ for Gaussian responses and $n = 100, 1000$ for binary responses.
- signal-to-noise-ratio SNR $= 0.5, 5$ for Gaussian responses
- for binary responses we scale the linear predictor so that the range of $P(y = 1|\boldsymbol{\eta})$ for each data set is restricted to $[0 + r, 1 - r]$ with $r = 0.05$ corresponding to a high SNR and $r = 0.2$ corresponding to a low SNR.
- degree of nonlinearity: $s = 0, 0.25, 0.5, 0.75, 1$ for Gaussian responses and $s = 0, 0.1, 0.25, 0.5, 0.75, 1, 1.5$ for binary responses
- 100 replications

For $\sigma = 0$, the function to be estimated is a simple line, so the correct model is one without a smooth term. Figure 4.8 shows the shape of the 3 functions for varying d. We use 10 basis functions to estimate the effects.

Analysis for Gaussian response

Figures 4.9 and 4.10 show type I and type II error rates along with square root of the mean square error $\|\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}\|^2/n$ for the various priors, additive models fit with `amer` (solid black lines) and tested with `exactRLRT` and component-wise boosting fit with `mboost` (dashed black lines) for Gaussian responses. Selection via component-wise boosting is extremely anti-conservative, with type I error rate between 60% and 95% and type II error rates below 20% across all settings, and comparatively large prediction error especially for strong nonlinearity and/or larger samples and SNR.

Type II error rates for `spikeSlabGAM` are heavily influenced by the prior settings, while type I error rates are very stable. Since smaller values of v_0 imply stronger regularization if the hypervariance is sampled from the "spike", the



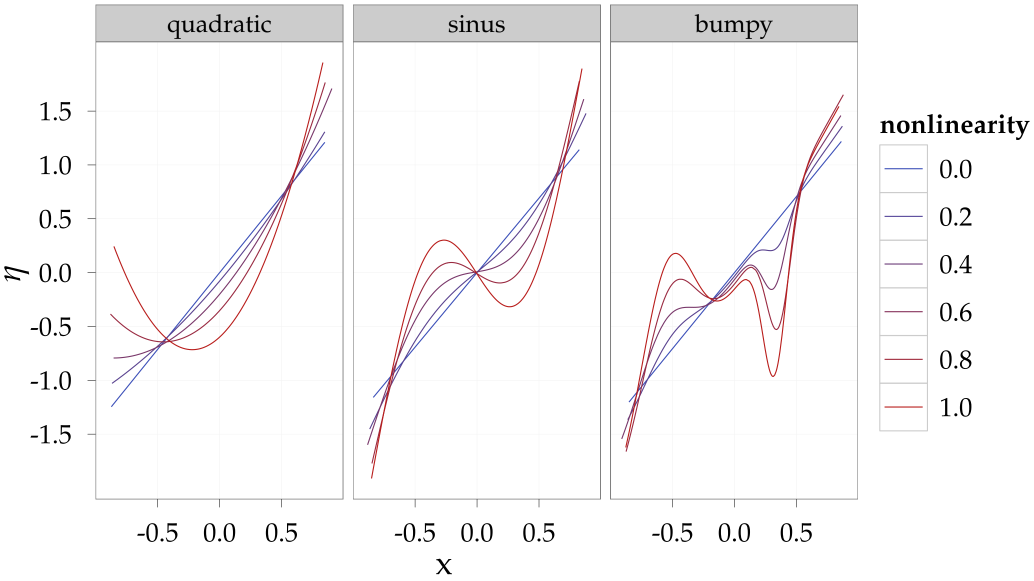


Figure 4.8.: True linear predictor for univariate smoothing simulations. Upper graph displays randomly generated functions: Upper row for 5 basis functions, lower row for 20 basis functions. Columns correspond to the different settings of $\sigma^2 > 0$. Bottom graph displays true linear predictors for the fixed functions: Columns correspond to the 3 different functions, line color indicates value of nonlinearity parameter s.

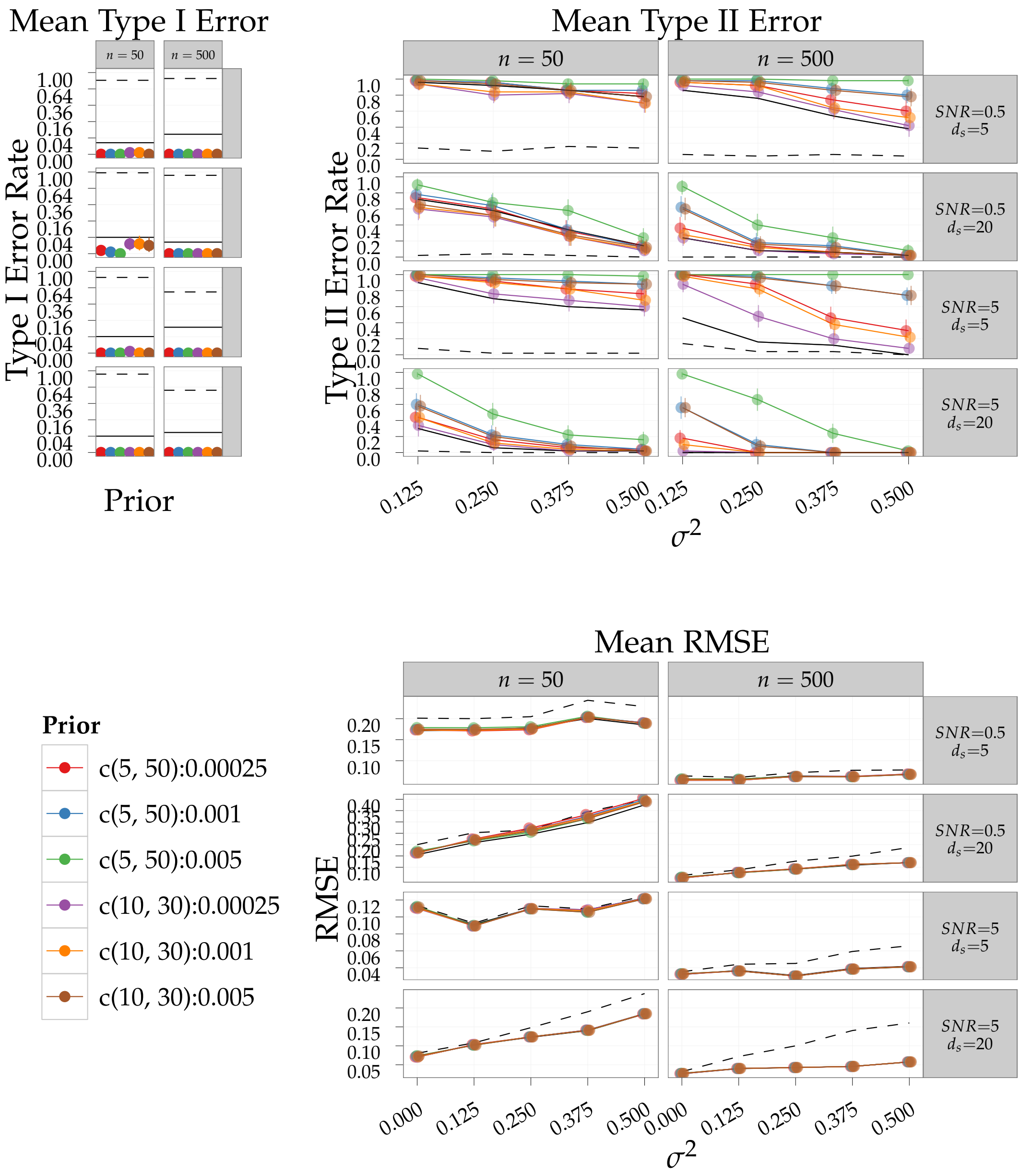


Figure 4.9.: Mean type I / type II error rates and $\sqrt{\text{MSE}}$ for randomly generated functions.
Left graph gives type I error for $\sigma^2 = 0$, right graph gives type II error rates for $\sigma^2 > 0$. Left column in each graph for $n = 50$, right column for $n = 500$. Upper two rows for SNR $= .5$ with $d_s = 5, 20$, lower two for SNR $= 5$. Graph on the lower right gives mean prediction $\sqrt{\text{MSE}}$. Solid black lines line gives error rates for the GAM (based on the p-value of a restricted LR-test with $\alpha = .05$), dashed black line for `mboost`. Vertical axis for type I error is on $\sqrt{\ }$-scale. Error bars show 95% CIs for mean error rates.

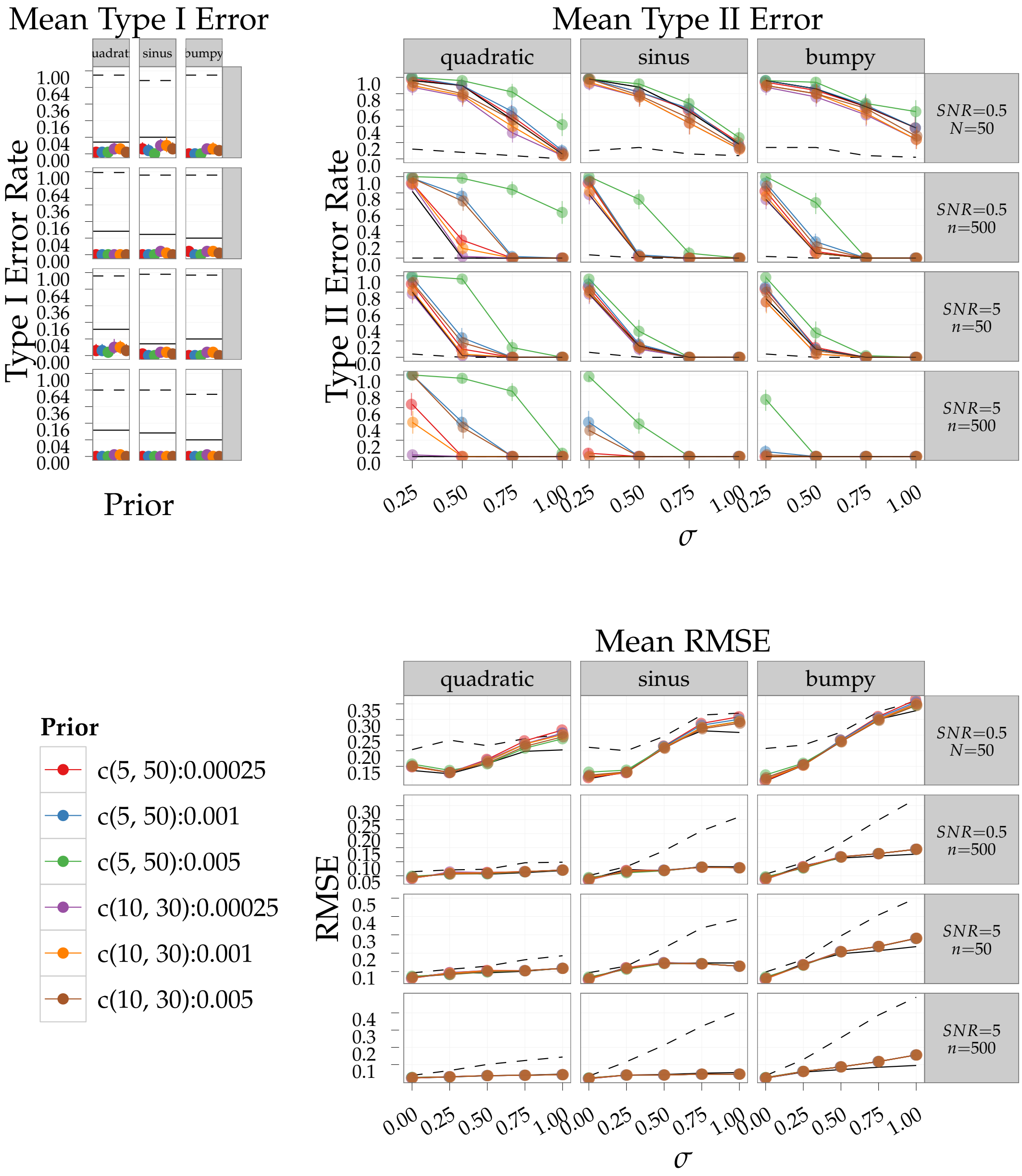


Figure 4.10.: Mean type I / type II error rates and $\sqrt{\text{MSE}}$ for fixed functions and Gaussian response. Columns correspond to the three different functions in the upper graphs. Left graph gives type I error for $\sigma = 0$, right graph gives type II error rates for $\sigma > 0$. Top two rows for SNR $= .5$ with $n = 50, 500$, bottom rows for SNR $= 5$. Graph on the lower right gives mean prediction $\sqrt{\text{MSE}}$. Solid black lines line gives error rates for the GAM (based on the p-value of a restricted LR-test with $\alpha = .05$), dashed black line for `mboost`. Vertical axis for type I error is on $\sqrt{}$-scale. Error bars show 95% CIs for mean error rates.

odds of sampling from the "spike" are smaller and thus the smaller values of v_0 have more "power" and are quicker to include smooth terms in the model (i.e. sample from the "slab") – the smaller v_0, the smaller is the threshold an effect has to cross in order to be included in the model.

Compared to function selection based on the RLRT with nominal $\alpha = .05$ – note that model selection via AIC corresponds to an RLRT with $\alpha = .05$ in this context (Greven, 2007, p. 104) – our approach is more conservative, i.e., has lower mean type I error rates across all of the considered settings and priors.

Correspondingly, type II error rates are mostly higher than those for the RLRT, especially for $(a_\tau, b_\tau) = (5, 50)$ and/or $v_0 = 0.005$. Nevertheless, the prior with $(a_\tau, b_\tau) = (10, 30)$ and $v_0 = 0.00025$ dominates the RLRT in terms of misclassification for some settings with low SNR and small samples and achieves very similar type II error rates to that of the RLRT across all settings. In general, type II error rates decrease about as fast as those of the RLRT, but on a higher absolute level. This reflects the fact that the model selection implemented in `spikeSlabGAM` is designed to select "relevant" terms and not "significant" terms. The threshold of relevance depends on (a_τ, b_τ) and v_0. In that sense, the generally very high exclusion rates for the randomly generated functions with $d_s = 5$ may be sensible behavior if the goal is to build a parsimonious model.

The graphs for the $\sqrt{\text{MSE}}$ on the lower right of Figures 4.9 and 4.10 show that even much larger type II error rates do not translate into larger estimation errors. For both randomly generated and fixed functions, the model averaging implicit in our procedure recovers the true predictor as good as the frequentist AM in this context and seems to perform much better than component-wise boosting, especially as nonlinearity increases. Across all settings, estimation errors are much more robust against the different prior settings than model selection.

Analysis for binary response

Figure 4.11 shows type I and type II error rates along with square root of the mean square error $\|\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}\|^2/n$ for binary responses. Results for additive models fit with `amer` and tested with a parametric bootstrap LRT are shown as solid black lines and component-wise boosting results with `mboost` are shown as dashed black lines. As for Gaussian responses, selection via component-wise boosting is extremely anti-conservative, with type I error rate above 90% for all settings and type II error rates below 20% across all settings, and comparatively large prediction error especially for larger samples and higher SNR (i.e., $r = 0.05$).

Type II error rates for `spikeSlabGAM` for binary response are influenced less

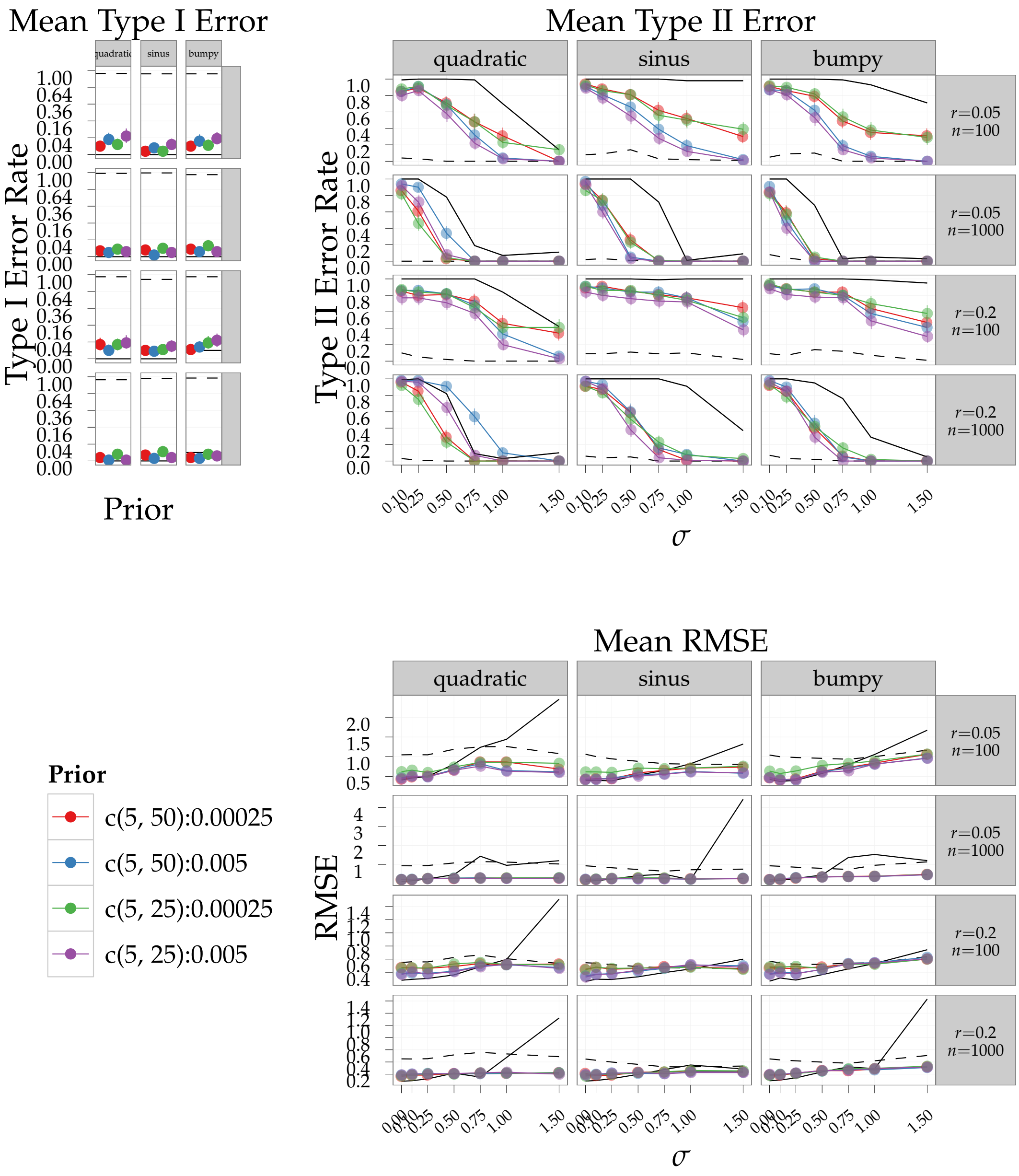


Figure 4.11.: Mean type I / type II error rates and $\sqrt{\text{MSE}}$ for fixed functions and binary binomial response. Columns correspond to the three different functions in the upper graphs. Left graph gives type I error for $\sigma = 0$, right graph gives type II error rates for $\sigma > 0$. Top two rows for $r = .05$ (high "SNR") with $n = 100, 1000$, bottom rows for $r = 2$ (low "SNR"). Graph on the lower right gives mean prediction $\sqrt{\text{MSE}}$. Solid black lines line gives error rates for the GAM (based on the p-value of a parametric bootstrap LR-test with $\alpha = .05$), dashed black line for `mboost`. Vertical axis for type I error is on $\sqrt{}$-scale. Error bars show 95% CIs for mean error rates.

strongly by the prior settings than for Gaussian response. Type I error rates are very stable and remain below 5%. Unlike for Gaussian responses, we do not observe a consistent pattern that would indicate larger "power" for prior configurations with smaller v_0. Compared to function selection based on a bootstrap LRT with nominal $\alpha = .05$, our approach is less conservative, i.e., has higher mean type I error rates and (much) lower type II error rates across most of the considered settings and priors. Note that even in the setting in the bottom right corner ("bumpy function" with 1000 observations and small range) where `spikeSlabGAM` achieves mostly lower type I error rates the type II error rates are much lower than that of the LRT.

The graphs for $\sqrt{\text{MSE}}$ on the lower right of Figure 4.11 show that even much lower type II error rates do not translate into consistently lower estimation errors. Nevertheless, the model averaging implicit in our procedure recovers the true predictor consistently better than boosting in this context, about as good as the frequentist AM for weaker nonlinearity and much better than the frequentist AM for strong nonlinearity.

4.1.5. Generalized additive models

In the following Sections 4.1.5 and 4.1.5, we compare the performance of peNMIG in (generalized) additive models to that of component-wise boosting (Hothorn, Bühlmann, Kneib, Schmid, and Hofner, 2010) in terms of predictive MSE and complexity recovery. As a reference, we also fit a conventional GAM (as implemented in `mgcv` (Wood, 2008)) based on the "true" formula (i.e. a model without any of the "noise" terms), which we subsequently call the "oracle"-model. For Gaussian responses only, we also compare our results to those from ACOSSO (Storlie, Bondell, Reich, and Zhang, 2011). ACOSSO is not able to fit non-Gaussian responses.

We supply separate base learners for the linear and smooth parts of covariate influence for the component-wise boosting in order to compare complexity recovery between boosting and our approach. We use 10-fold cross validation on the training data to determine the optimal stopping iteration for `mboost` and count a baselearner as included in the model if it is selected in at least half of the cross-validation runs up to the stopping iteration. BIC is used to determine the tuning parameter for ACOSSO. We were unable to compare our approach to Reich et al. (2009), which is implemented for Gaussian responses, since the available R implementation is impractically slow.

For both Gaussian responses (Section 4.1.5) and Poisson responses (Section 4.1.5), the data generating process has the following structure:

- We define 4 functions
  - $f_1(x) = x$,

- $f_2(x) = x + \frac{(2x-2)^2}{5.5}$,
- $f_3(x) = -x + \pi \sin(\pi x)$,
- $f_4(x) = 0.5x + 15\phi(2(x - .2)) - \phi(x + 0.4)$, where $\phi()$ is the standard normal density function,

which enter into the linear predictor. Note that all of them have (at least) a linear component.

- We define 2 scenarios:
  - a "low sparsity" scenario: Generate 16 covariates, 12 of which have non-zero influence. The true linear predictor is

$$\begin{aligned}\eta = &f_1(x_1) + f_2(x_2) + f_3(x_3) + f_4(x_4) + \\ &+ 1.5(f_1(x_5) + f_2(x_6) + f_3(x_7) + f_4(x_8)) + \\ &+ 2(f_1(x_9) + f_2(x_{10}) + f_3(x_{11}) + f_4(x_{12})).\end{aligned}$$

  - a "high sparsity" scenario: Generate 20 covariates, only 4 of which have non-zero influence and $\eta = f_1(x_1) + f_2(x_2) + f_3(x_3) + f_4(x_4)$.
- The covariates are either
  - $\overset{\text{i.i.d.}}{\sim} U[-2, 2]$ or
  - from an AR(1) process with correlation $\rho = 0.7$.
- We simulate 50 replications for each combination of the various settings.

We compare 9 different prior specifications:

- $(a_\tau, b_\tau) = (5, 25), (10, 30), (5, 50)$
- $v_0 = 0.00025, 0.005, 0.01$

Predictive MSE is evaluated on test data sets with 5000 observations. Complexity recovery, i.e. how well the different approaches select covariates with true influence on the response and remove covariates without true influence on the response is measured in terms of accuracy, defined as the number of correctly classified model terms (true positives and true negatives) divided by the total number of terms in the model. For example, the full model in the "low sparsity" scenario has 32 potential terms under selection (linear terms and basis expansions/smooth terms for each of the 16 covariates), only 21 of which are truly non-zero (the linear terms for the first 12 covariates plus the 9 basis expansions of the covariates not associated with the linear function $f_1()$). Accuracy in this scenario would then be determined as the sum of the correctly included model terms plus the correctly excluded model terms, divided by 32.

Gaussian response

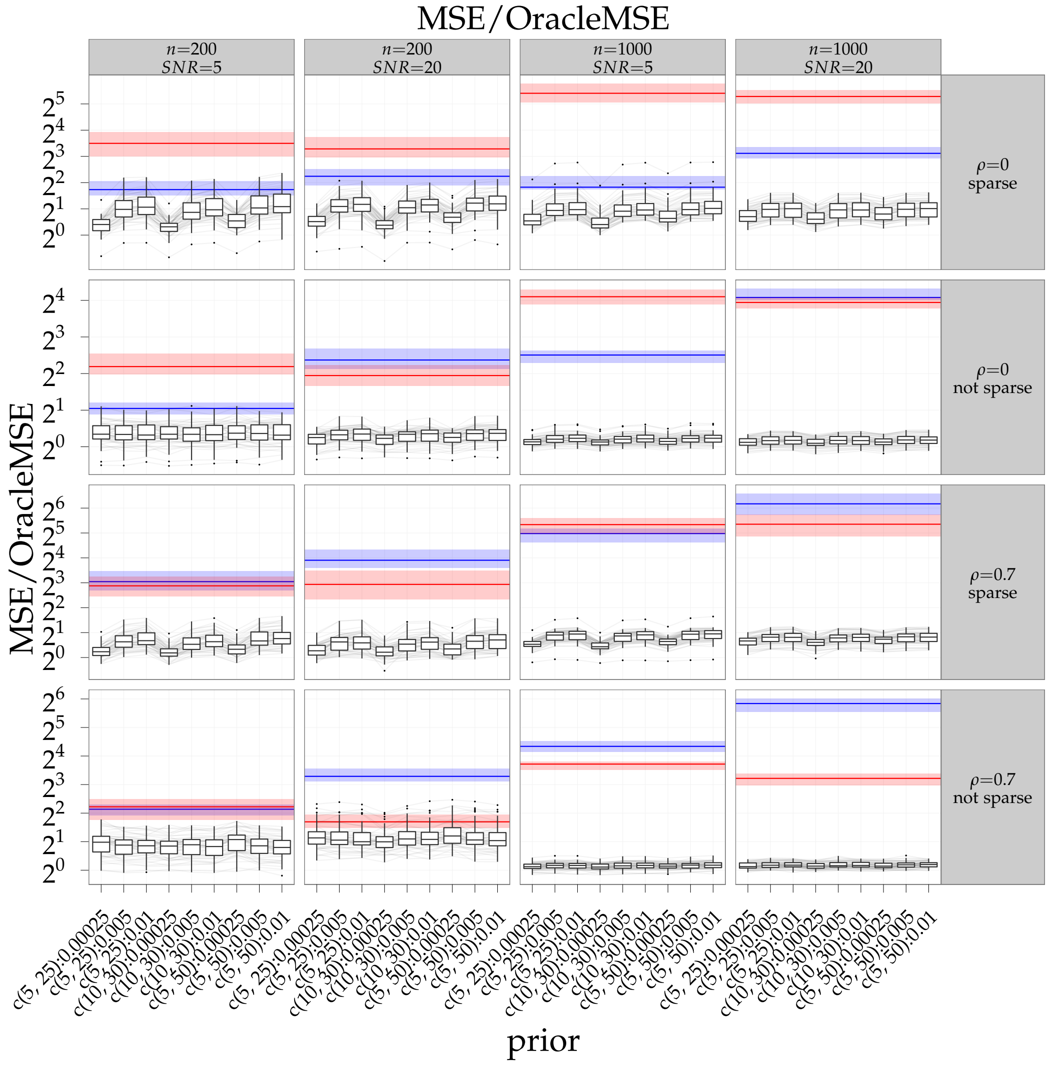


Figure 4.12.: Prediction MSE divided by oracle MSE for Gaussian response. White boxplots show results for the different prior settings, blue and red symbols show results for `mboost` and ACOSSO, respectively: Shaded region gives IQR, line represents median. Dark grey lines connect results for the same replication. Columns from left to right: 200 obs. with SNR=5, 20; 1000 obs. with SNR=5, 20. Rows from top to bottom: uncorrelated obs. with sparse and unsparse predictor, correlated obs. with sparse and unsparse predictor. Vertical axis is on binary log scale.

In addition to the basic structure of the data generating process described at the beginning of this section, the data generating process for the Gaussian

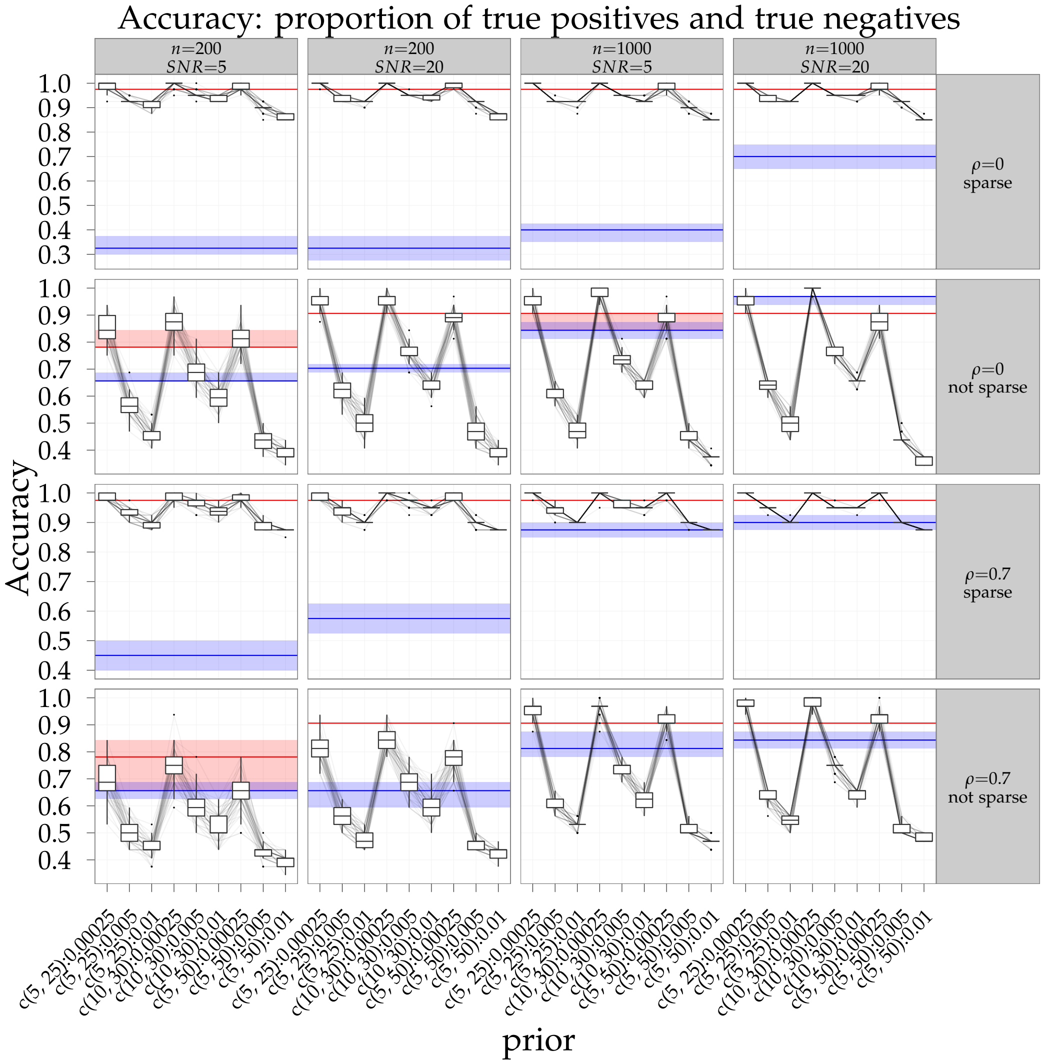


Figure 4.13.: Complexity recovery for Gaussian response: proportion of correctly included and excluded model terms. White boxplots show results for the different prior settings, blue and red symbols show results for `mboost` and ACOSSO, respectively: Shaded region gives IQR, line represents median. Dark grey lines connect results for the same replication. Columns from left to right: 200 obs. with SNR=5, 20; 1000 obs. with SNR=5, 20. Rows from top to bottom: uncorrelated obs. with sparse and unsparse predictor, correlated obs. with sparse and unsparse predictor.

responses has the following properties:

- signal-to-noise-ratio SNR $= 5, 20$
- number of observations: $n = 200, 1000$

Figure 4.12 shows the mean squared prediction error divided by the one achieved by the "oracle"-model, a conventional GAM without any of the noise variables. Predictive performance is very robust against the different prior settings especially for the settings with low sparsity. Different prior settings also behave similarly within replications, as shown by the mostly parallel grey lines. Predictions are more precise than those of both boosting and ACOSSO, and this improvement in performance relative to the "true" model is especially marked for $n = 1000$ (two rightmost columns). With the exception of the first scenario, the median relative prediction MSE is < 2 everywhere, while both boosting and ACOSSO have a median relative prediction MSE above 4 in most scenarios that goes up to above 32 and 64 for ACOSSO and boosting, respectively, in the "large sample, correlated covariates" cases. In the "large sample, low sparsity" scenarios (two leftmost columns in rows two and four), the performance of our approach comes very close that of the oracle model – the relative prediction MSEs are close to one.

Figure 4.13 shows the proportion of correctly included and excluded terms (linear terms and basis expansions) in the estimated model. Except for $v_0 = 0.00025$, accuracy is consistently lower than for ACOSSO. However, a direct comparison with ACOSSO is not entirely appropriate because ACOSSO does not differentiate between smooth and linear terms, while `mboost` and our approach do. Therefore ACOSSO solves a less difficult problem. Estimated inclusion probabilities are very sensitive to v_0 and comparatively robust against (a_τ, b_τ). Across all settings, $v_0 = 0.00025$ delivers the most precise complexity recovery, with sensitivities consistently above 0.7. The accuracy of peNMIG is better than `mboost` for the sparse settings (1st and 3rd rows) because the specificity of our approach is $> .97$ across settings, regardless of the prior (!), while `mboost` mostly achieves only very low specificity, but fairly high sensitivity.

Effect of centering the design

All the results in Sections 4.1 and 4.2 are based on design matrices for the penalized parts of smooth effects that are orthogonalized against the designs functions in their nullspace, as described on page 36. Figure 4.14 shows the ratios of the mean square prediction error for settings of the simulation study in Section 4.1.5 if this orthogonalization is omitted: With the sole exception of the sparse setting with correlated covariates for 1000 observations and $SNR = 5$, the median ratio is > 1, i.e. using the orthogonalized designs yields lower prediction errors in more than half of the replications across almost all of

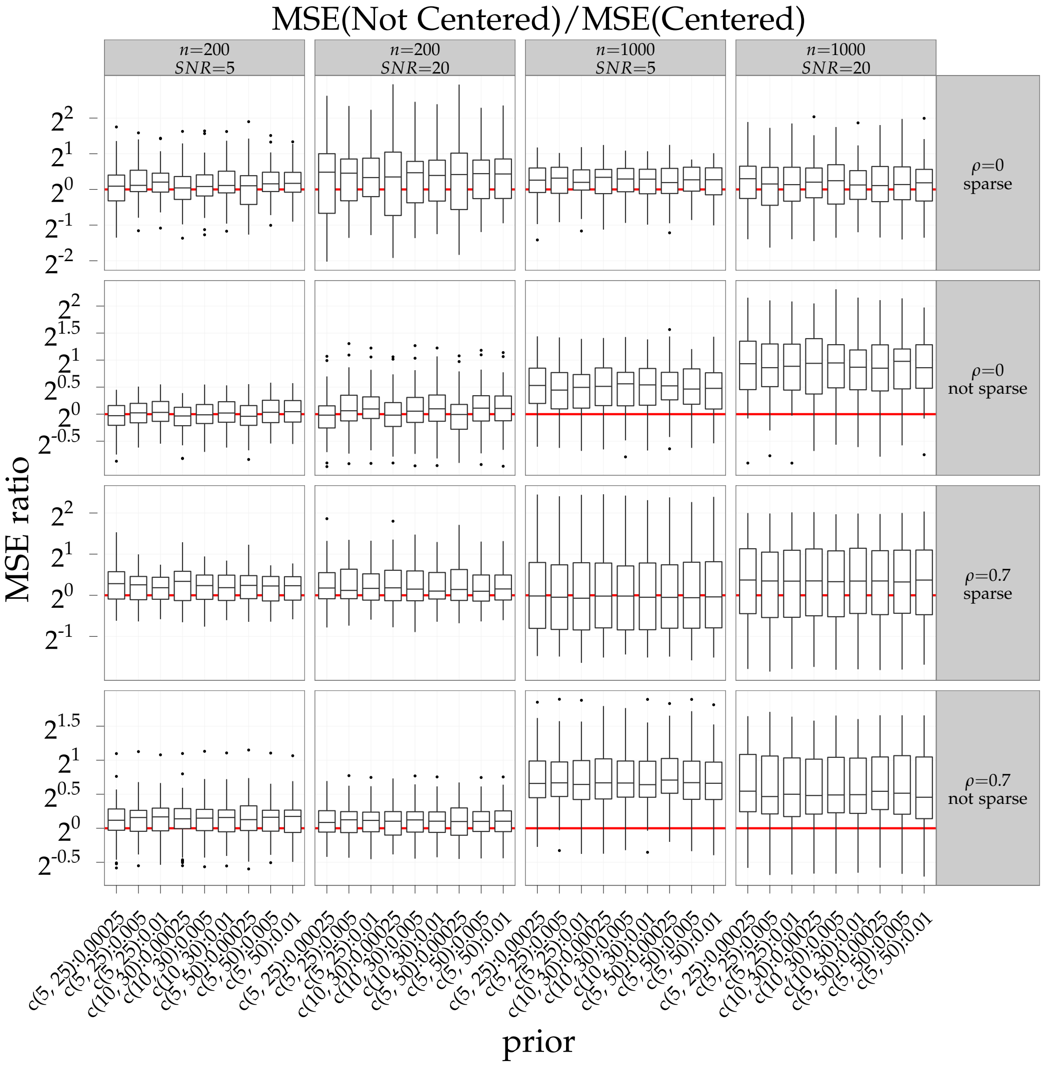


Figure 4.14.: Ratios of prediction MSE between uncentered and centered designs, i.e. MSE without orthogonalization divided by MSE with orthogonalization. White boxplots show results for the different prior settings. Red horizontal line marks a ratio one. Columns from left to right: 500 obs., 2000 obs. Rows from top to bottom: uncorrelated obs. with sparse and unsparse predictor, correlated obs. with sparse and unsparse predictor.

the settings and priors, and fairly large median gains in prediction accuracy occur for settings with larger sample sizes and low sparsity (median MSE ratios between 1.36 and 1.97 in the two leftmost columns for the second and fourth rows).

Poisson response

In addition to the basic structure of the data generating process described at the beginning of this section, the data generating process for the Poisson responses has the following properties:

- number of observations: $n = 500, 2000$
- responses are generated with overdispersion: $y_i \sim Po\left(s_i \exp(\eta_i)\right);\ s_i \sim U[0.66, 1.5]$

We did not use $v_0 = 0.01$ for this experiment because of its inferior performance in terms of complexity recovery in the Gaussian case.

Figure 4.15 shows the mean squared prediction error (on the scale of the linear predictor) divided by the one achieved by the "oracle"-GAM that includes only the relevant covariates and no noise terms. Predictive performance is very robust against the different prior settings. Different prior settings also behave similarly within replications, as shown by the mostly parallel grey lines. Predictions are more precise than those of `mboost`, especially for smaller data sets (left column) and correlated responses (two bottom rows). For the "low sparsity, correlated covariates" setting (bottom row), the performance of our approach comes fairly close to that of the "oracle"-GAM, with relative prediction errors mostly between 1 and 1.5, and occasionally even improving on the oracle model for $n = 500$.

Figure 4.16 shows the proportion of correctly included and excluded terms (linear terms and basis expansions) in the estimated models. Estimated inclusion probabilities are sensitive to v_0 and comparatively robust against (a_τ, b_τ). The smaller value for v_0 tends to perform better in the unsparse settings (second and fourth rows) since it forces more terms into the model (resulting in higher sensitivity and lower specificity) and vice versa for the sparse setting and the larger v_0. Complexity recovery is (much) better across the different settings and priors for our approach than for boosting. The constant accuracy for `mboost` in the low sparsity scenario with uncorrelated responses (second row) is due to its very low specificity: It includes practically all model terms all the time.

The simulations for generalized additive models show that the proposed peNMIG-Model is very competitive in terms of estimation accuracy and confirms that estimation results are robust against different hyperparameter configurations even in fairly complex models. Model selection is more sensi-

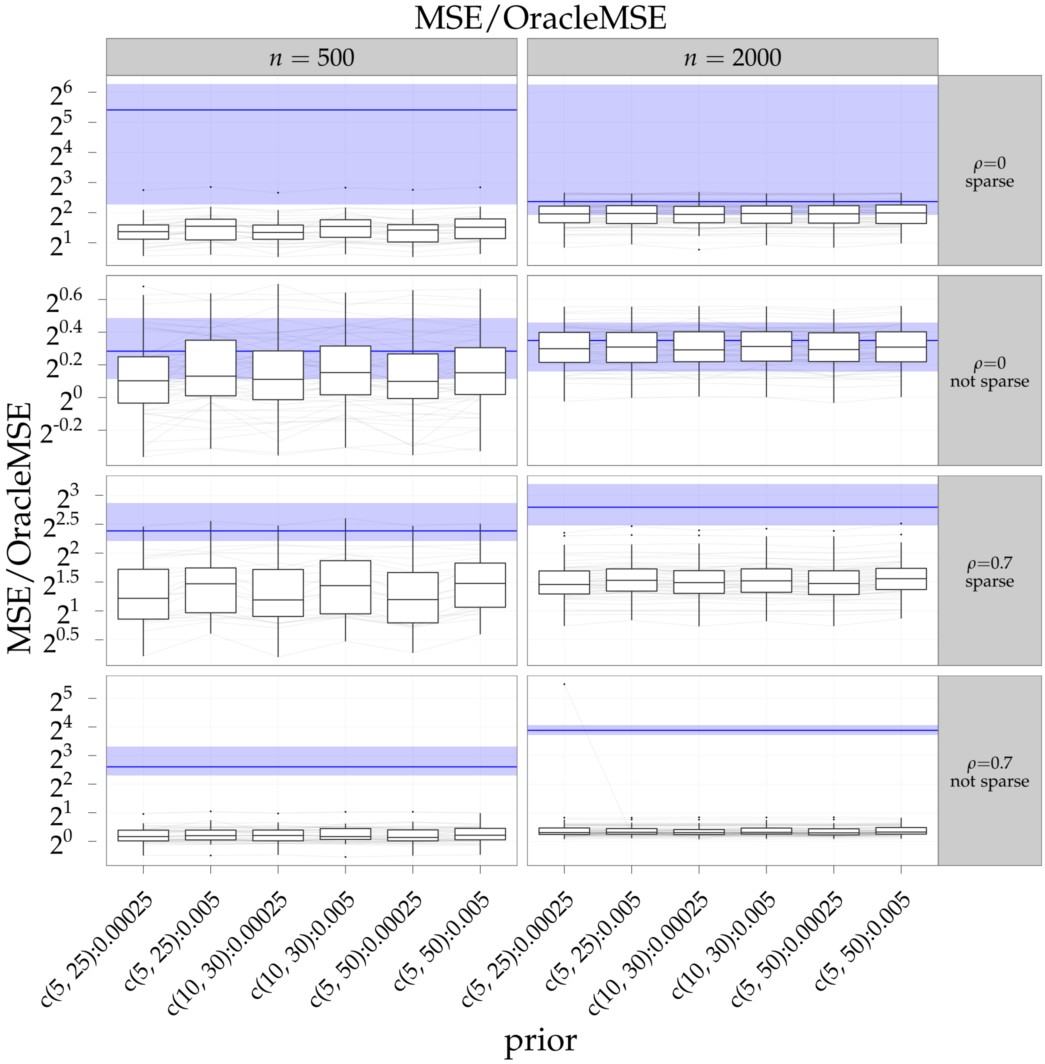


Figure 4.15.: Prediction MSE divided by oracle MSE (on the scale of the linear predictor). White boxplots show results for the different prior settings. Blue symbols show results for `mboost`: shaded region gives IQR, line represents median. Dark grey lines connect results for the same replication. Columns from left to right: 500 obs., 2000 obs. Rows from top to bottom: uncorrelated obs. with sparse and unsparse predictor, correlated obs. with sparse and unsparse predictor. Vertical axis is on binary log scale.

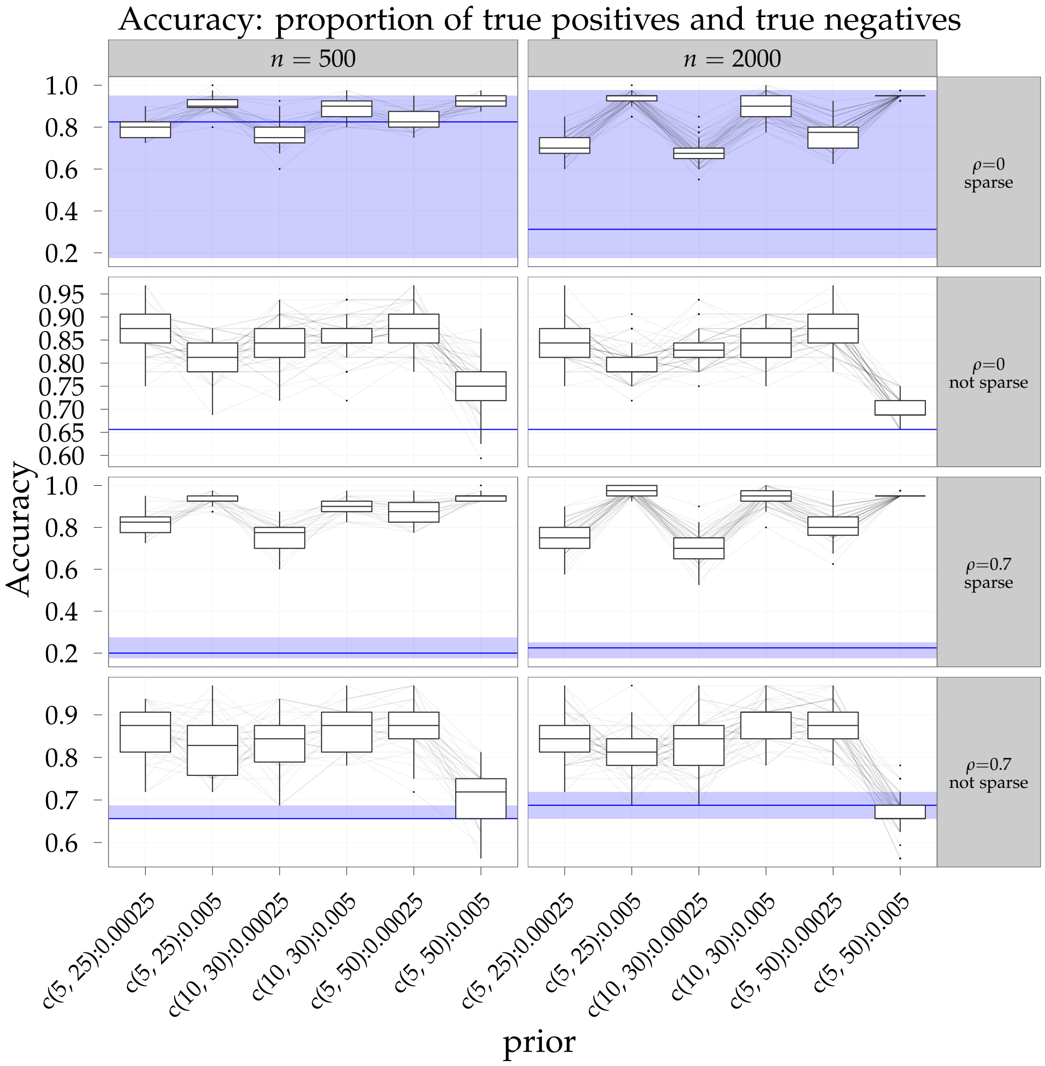


Figure 4.16.: Complexity recovery for Poisson response: proportion of correctly included and excluded model terms. White boxplots show results for the different prior settings. Blue symbols show results for `mboost`: shaded region gives IQR, line represents median. Dark grey lines connect results for the same replication. Columns from left to right: 500 obs., 2000 obs. Rows from top to bottom: uncorrelated obs. with sparse and unsparse predictor, correlated obs. with sparse and unsparse predictor.

tive towards hyperparameter configurations, especially v_0. For smaller v_0, spikeSlabGAM seems to be able to distinguish between important and irrelevant terms fairly reliably.

We are not aware of any other SSVS implementations for function selection in additive models with non-Gaussian responses available for benchmarking, but the performance of peNMIG as implemented in spikeSlabGAM seems to be very competitive to that of component-wise boosting as implemented in mboost. Results for an earlier, more rudimentary implementation of the peNMIG model on identical data generating processes are published in (Scheipl, 2010b).

4.2. Binary classification: UCI data benchmarks

We fit additive models with a logit link to a collection of 21 data sets for binary classification from the UCI Machine Learning Repository (Asuncion and Newman, 2007), as previously analyzed in Eugster, Hothorn, and Leisch (2008) and Meyer, Leisch, and Hornik (2003). Figure 4.17 gives an overview of the datasets we use and their characteristics. The vertical axis gives the number of covariates and different factor levels, the horizontal axis gives the number of (complete) observations. Most of the datasets contain a mixture of continuous and factor variables. We do not consider any interactions, only linear and smooth main effects. We evaluate prediction performance based on the deviance values for a 20-fold cross validation on each dataset. Predictive deviance $\bar{D}$ is defined as twice the average negative log likelihood

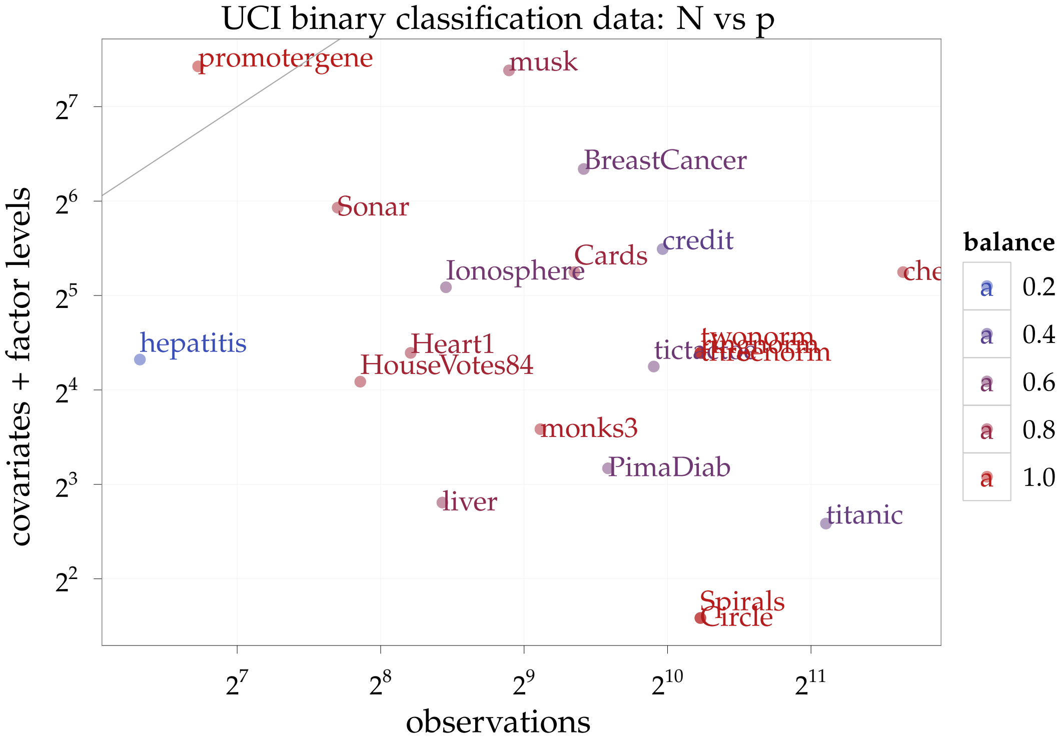


Figure 4.17.: Characteristics of UCI data sets: number of observations versus number of features.
"Balance" is the ratio between the number of observations in the larger class and the number of observations in the smaller class, i.e. it is 1 if the data set is balanced. `promotergene` is the only dataset we consider that has more parameters than observations before accounting for spline basis expansions.

$\bar{D} = -2/n_P \sum_{i=1}^{n_P} L(y_{P,i}, \hat{\eta}_{P,i})$ in the test sample where $\boldsymbol{y}_P$ and $\hat{\boldsymbol{\eta}}_P$ are the out-of-sample responses and estimated linear predictors for the test sample. The size of the test sample is n_P. AUC is defined as the area under the receiver-operator-characteristic (ROC) curve and can be expressed as

$$\text{AUC}(\boldsymbol{y}, \boldsymbol{\eta}) = \left(|\{j : y_j = 0\}||\{i : y_i = 1\}|\right)^{-1} \sum_{\{j:y_j=0\}} \sum_{\{i:y_i=1\}} I(\eta_i - \eta_j > 0),$$

i.e. the proportion of pairs of cases ($y = 1$) and non-cases ($y = 0$) for which the predictor η is larger for the case than for the non-case.

As for the experiments with simulated data, we use component-wise boosting with separate base learners for the linear and smooth parts of covariate influence and compare prediction performance of the boosting models to our approach. We additionally compare the precision of the estimated probabilities to those for a simpler version of `mboost` with no separation into linear and smooth baselearners. Results for `mboost` are based on a stopping iteration determined via the empirical risk on 10 cross-validation folds of each training data set.

Besides the accuracy of the predictions, we are interested in the parsimony of the estimated models. To measure this, we can count the number of model terms (baselearners for boosting) included in the model. As usual, we count a model term as "included" if its marginal posterior inclusion probability is greater than 0.5. For boosting, the relevant quantities are the selection frequencies of the baselearners. We count a baselearner as included in the model if it was selected earlier than iteration m_{stop} in more than half of bootstrap samples used to determine the stopping iteration m_{stop}. Since this definition of term inclusion for `mboost` is somewhat arbitrary, we also compare the proportion of unequivocally excluded model terms: A baselearner is considered to be unequivocally excluded if it was not selected at all up to the stopping iteration. For `spikeSlabGAM`, we consider a term to be unequivocally excluded if its marginal posterior inclusion probability is smaller than 0.05.

Results are based on ten parallel MCMC chains with a burn-in of 1000 iterations and a sampling phase of 4000 iterations of which we save every fifth.

The data are preprocessed fairly brutally in an automated fashion in order to preempt possible numerical problems: All covariates with less than 6 unique values are coded as factor variables. All numeric covariates are scaled to the unit interval $[0, 1]$ first, followed by taking the logarithm of the covariate values (plus an offset of 0.1) if skewness is greater than 2 or taking the logarithm of 1.1 minus the covariate value if skewness is below -2. All numeric covariates (transformed or not) are then standardized to have mean 0 and standard deviation 1. All incomplete observations are removed.

We evaluate our approach for two model building scenarios: For the first one, we perform an automated preselection procedure to generate model formulas based on the following heuristic, which roughly follows the ideas in Harrell (2001):

1. Determine the "available degrees of freedom" for the smooth terms by dividing the number of observations in the smaller one of the two classes by 3 and subtracting the sum of the number of levels of all factor variables in the data.
2. - if the available degrees of freedom are larger than 4 times the number of numeric covariates, assign a spline expansion with 10 basis functions to each numerical covariate. You're done.
   - if not go to next step
3. - split all numerical covariates by quintile
   - perform χ^2-tests of association of the resulting 5-level factors with the response
   - sort numerical covariates by decreasing strength of association (as measured by the p-value of the χ^2-test)
4. starting with the covariate with the strongest marginal association with the response, assign spline expansions with 5 basis functions to the numerical covariates and subtract 5 "available degrees of freedom" until no more degrees of freedom are left
5. if any numerical covariates remain after all available degrees of freedom are spent, they enter the model as simple linear terms.

This approach results in model specifications that are below the maximum complexity for datasets `credit`, `Cards`, `Heart1`, `Ionosphere`, `hepatitis`, `Sonar` and `musk`. Models for these datasets include only linear terms for some of the covariates.

In the second approach, we assign a spline expansion with 5 basis functions to all numerical covariates regardless of the number of predictors and observations, leading to a more difficult estimation and selection problem in data sets with large p and small n. Results for this second approach are discussed in Section 4.2.2.

4.2.1. Models with preselection of smooth terms

We show results for combinations of $(a_\tau, b_\tau) = (10, 50)$ or $(5, 25)$ and $v_0 = 0.005$ or 0.00025. We use a uniform prior $w \sim \text{Beta}(1, 1)$.

Figure 4.18 shows the achieved predictive performance for the first model building strategy for the 21 datasets. Outliers with large deviances for the

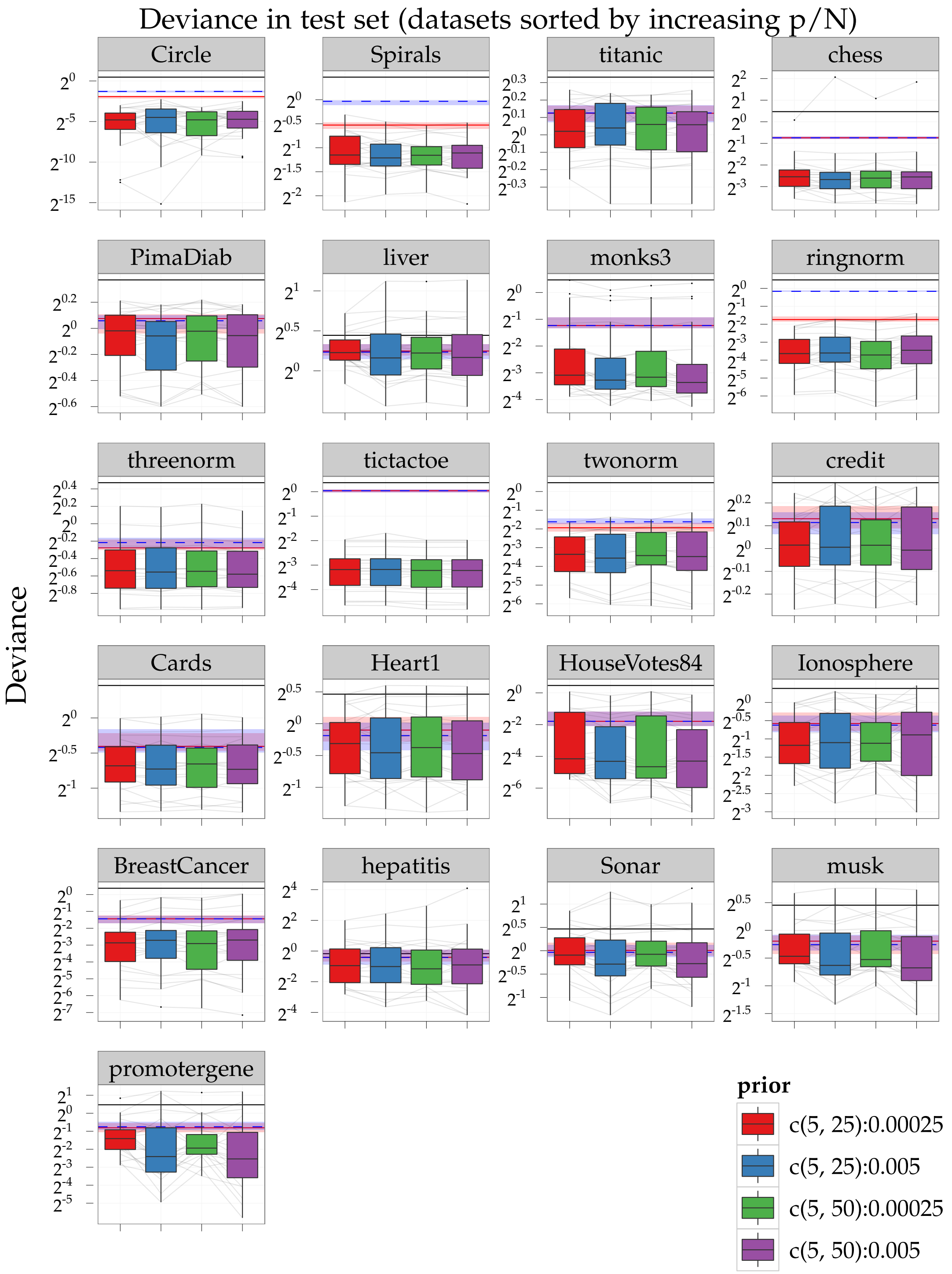


Figure 4.18.: UCI data with preselection: Predictive deviances for 20-fold CV. Boxplots show results for the different prior settings. Blue and red symbols show results for `mboost` with and without separate linear and smooth baselearners: Shaded regions give IQR, line represents median. Dark grey lines connect results for the same fold. Horizontal black line gives average predictive deviance for the null model.

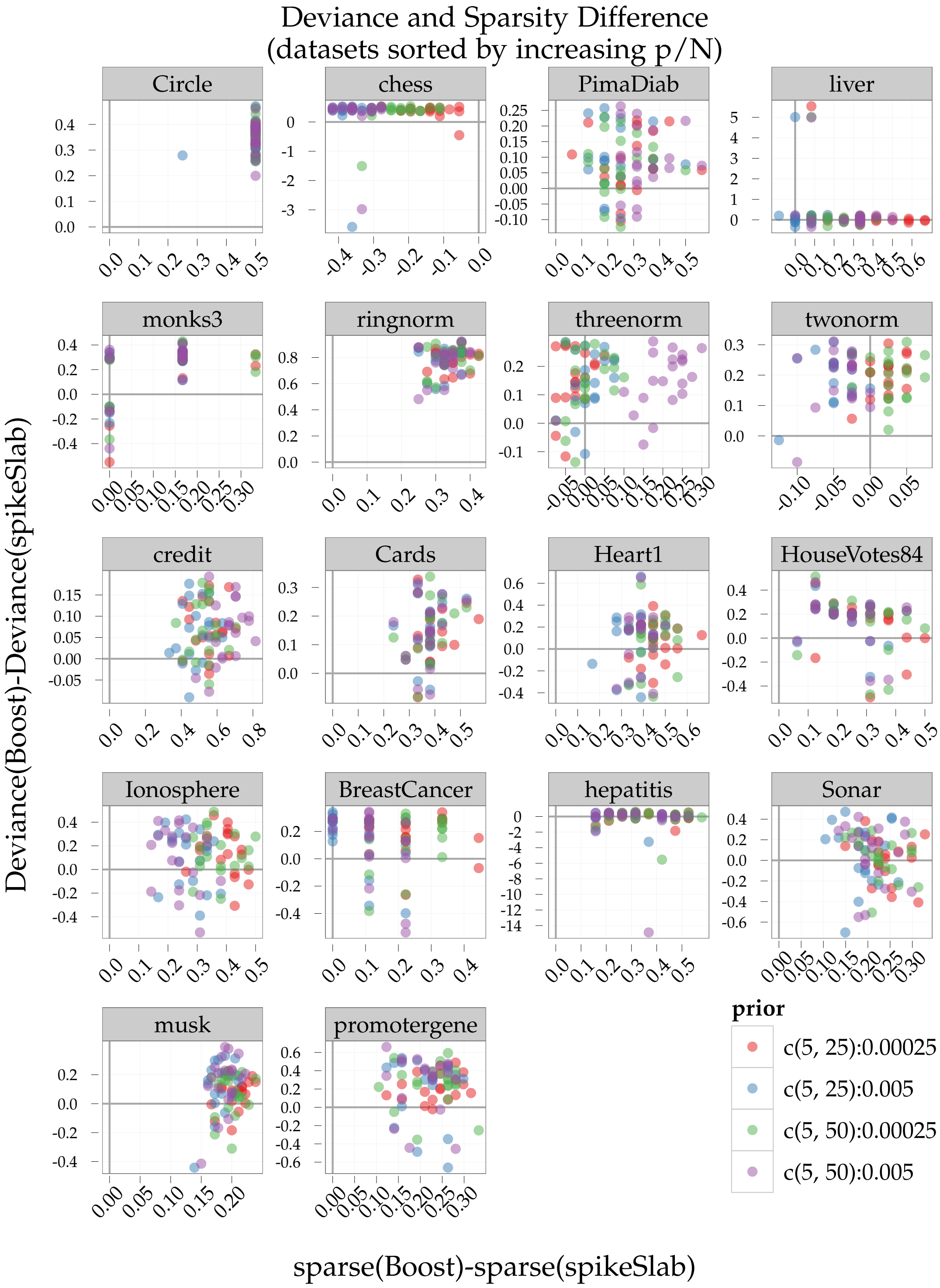


Figure 4.19.: UCI data with preselection: Difference in proportion of included model terms versus differences in predictive deviances. Positive values denote smaller deviances/models for our approach compared to `mboost`. Results for `Spirals`, `titanic` and `tictactoe` not shown because there were no differences in sparsity.

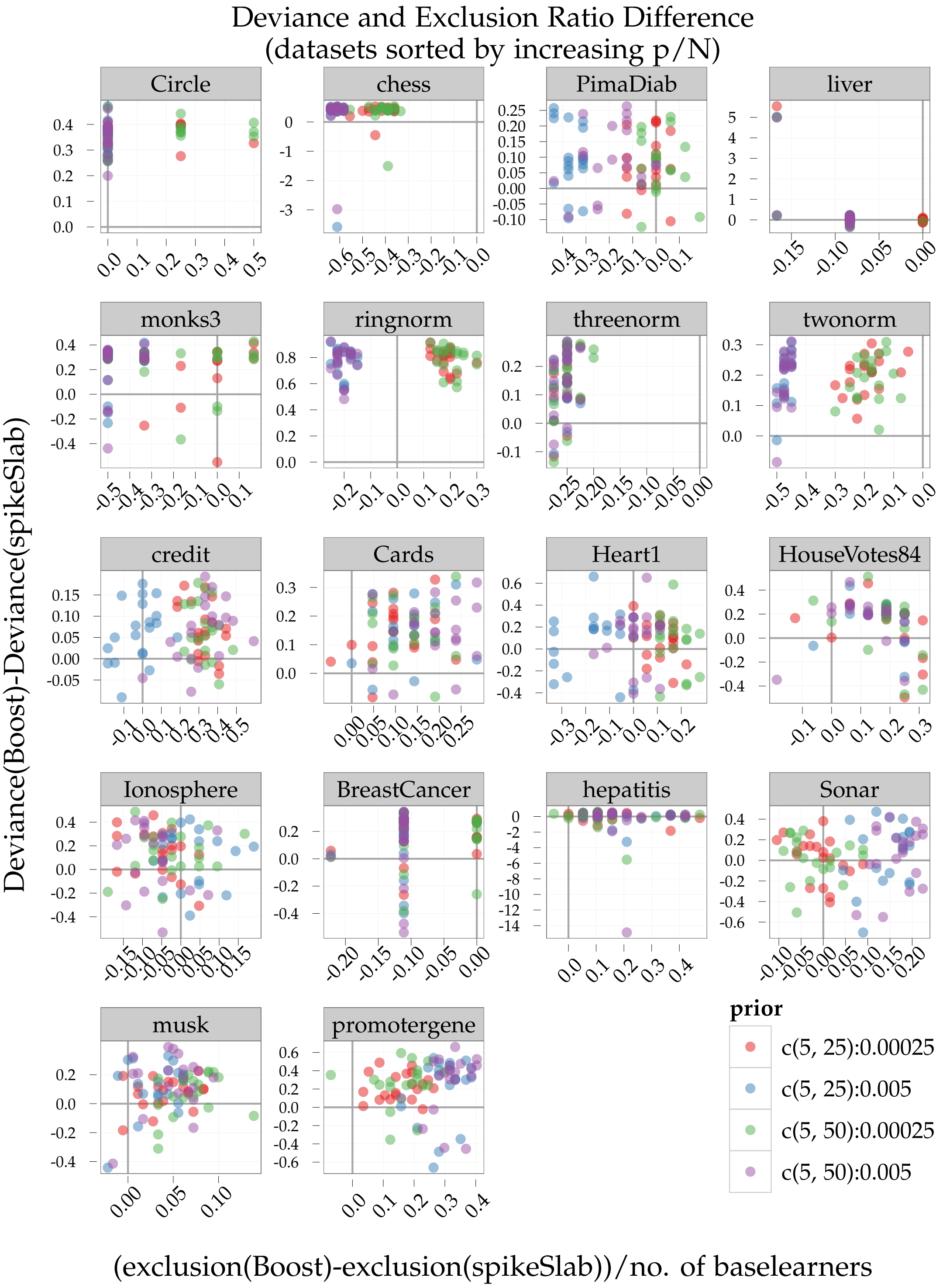


Figure 4.20.: UCI data with preselection: Difference in proportion of excluded model terms versus differences in predictive deviances. Positive values denote smaller deviances/ more excluded terms for our approach compared to `mboost`. Results for `Spirals`, `titanic` and `tictactoe` not shown because there were no differences in sparsity.

chess, monks3 and hepatitis data sets are due to the sampler getting stuck for some of the parallel chains in specific folds. By rerunning the analysis with different starting values or random seeds or by removing the samples of the offending chains from the posterior estimates, these could have presumably been avoided. We include them unchanged to provide a more realistic picture of the reliability of our approach. Practitioners should always check traceplots and convergence diagnostics for MCMC-based methods.

Note that the performance of our approach is more variable than that of mboost, but has lower median predictive deviances in all of the datasets for all four prior specifications. Predictive performance is very robust against different hyperparameter settings, even for large p/N where the influence of the hyperparameters on the posterior is stronger. Table 4.2 gives the median deviances and AUCs (area under the ROC-curve) for the different datasets and priors.

To investigate the parsimony of the fitted models, i.e., whether equivalent or better prediction can be achieved by simpler models, we plot the differences in predictive deviances versus the difference in the proportion of potential model terms included in the models for each cross-validation fold in Figure 4.19. Results for datasets Spirals, titanic and tictactoe are omitted because there were no differences in sparsity. Positive values on the vertical axis indicate smaller deviance for our approach, and positive values on the horizontal axis indicate a sparser fit for our approach. Figure 4.19 shows that our approach achieves its relatively more precise predictions with fewer retained terms on the large majority of the benchmarked data sets. The only exceptions are datasets chess, where the increased precision is achieved at the cost of less sparse models, and, to a much lesser extent twonorm and threenorm. There does not seem to be a correlation between model size and accuracy of the predictions and neither absolute performance nor performance relative to boosting seem to be tied to any of the easily observable characteristics of the data sets (i.e. p, N, p/N or balancedness of the response).

Figure 4.20 uses the difference in proportion of unequivocally excluded model terms as the metric for the sparsity of the estimated models. As before, the sparsity of the estimated models does not seem to affect the precision of the predictions as measured by the predictive deviance. For most of the less heavily parameterized datasets, we see the expected pattern of fewer deselected terms for smaller v_0 (in blue and green), but this pattern is reversed for two of the three most heavily parameterized datasets (Sonar, promotergene). For this metric of model parsimony, the comparison with mboost is slightly less favorable – in four of the 21 datasets, it consistently removes more terms from the model than our approach. spikeSlabGAM fits better models with fewer terms for the heavily overparameterized promotergene and hepatitis data sets.

In summary, no clear picture emerges for the differences between the four prior specifications: As expected, a smaller v_0 (green and blue) tends to yield models with more included terms, cf. datasets chess, twonorm, Ionosphere, musk, but there are counterexamples as well, e.g. threenorm, and for most datasets there does not seem to be a pattern at all. Both predictive deviance and sparsity results are much more sensitive towards v_0 than towards (a_τ, b_τ).

Table 4.3 gives the average run times for mboost. Computation times were recorded on a server with ten AMD Opteron 6174 processors with 2.2 GHz, i.e., both the ten chains for spikeSlabGAM and the ten cross-validation folds to determine the stopping iteration for the mboost-fits were run in parallel. Note that spikeSlabGAM is actually faster for many of the smaller problems, but its computation time increases much faster than that for mboost as the number of terms and observations increases due to the additional steps needed for the parameter expansion.

4.2.2. Models without preselected smooth terms

We use the second model-building strategy and repeat the analysis without restricting the number of smooth terms for data sets credit, Cards, Heart1, Ionosphere, hepatitis, Sonar and musk in order to evaluate the performance on more high dimensional and overparameterized problems. For all other data sets, the model without preselection would have been the same as the one with preselection. We use slightly different priors for this model building strategy to investigate the possibility of enforcing sparsity via the specification of an informative prior on w in high-dimensional models with $n \lessapprox q$ such as the ones considered here. We also use NMIG instead of peNMIG, that is, we omit the parameter expansion for model terms with $d = 1$ (i.e. linear terms and binary factors) to reduce the posterior's dimensionality. Reported results are for combinations of $v_0 = 0.005, 0.00025$ and $(a_w, b_w) = (1, 1), (20, 40)$ with $(a_\tau, b_\tau) = (5, 25)$.

Figures 4.21 and 4.22 show deviance values for the test data and differences in deviances and sparsity between our approach and componentwise boosting with mboost. Figures 4.23 shows the differences in the proportion of unequivocally excluded terms versus differences in deviance. Compared to the results for the models with preselection, prediction performance for our approach as measured by the median predictive deviance improves in five of the seven datasets (the exceptions are credit and Ionosphere) when smooth terms of all numeric covariates are allowed to enter the model. This indicates that the procedure avoids overfitting even for heavily overparameterized models. The gains are usually fairly small. Figures 4.22 and 4.23 reinforce the conclusion from the previous section: our approach achieves its

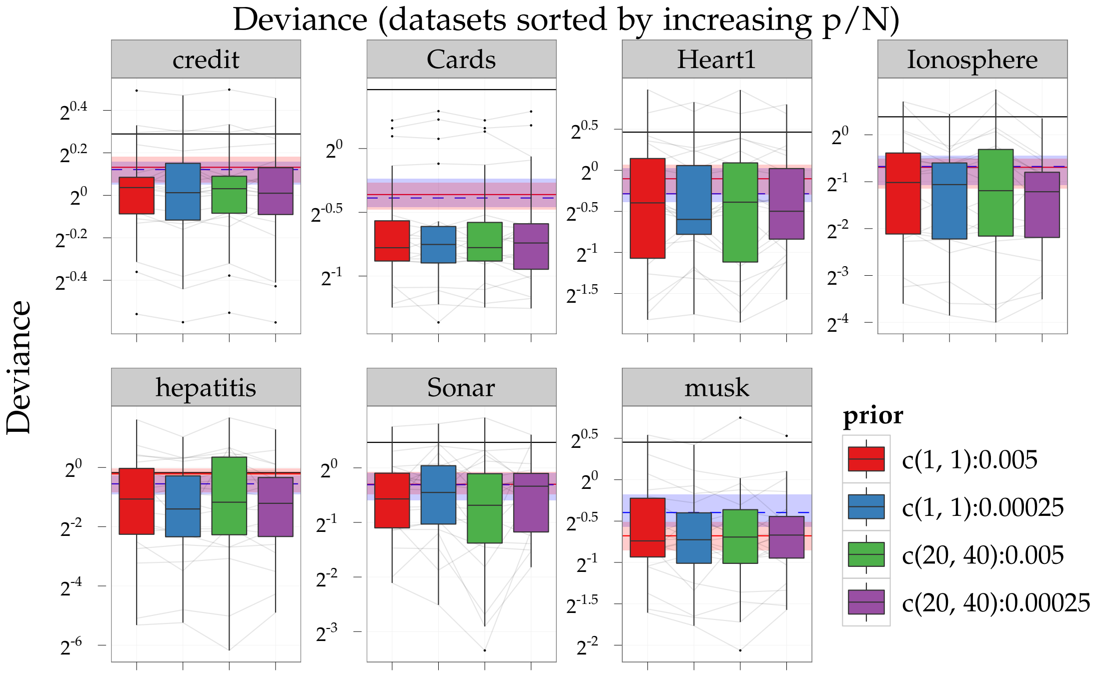


Figure 4.21.: UCI data results without preselection: Predictive deviances for 20-fold crossvalidation. Boxplots show results for the different prior settings. Blue and red symbols show results for `mboost` with and without separate linear and smooth base-learners: Shaded regions give IQR, line represents median. Dark grey lines connect results for the same fold. Horizontal black line gives average predictive deviance for the null model.

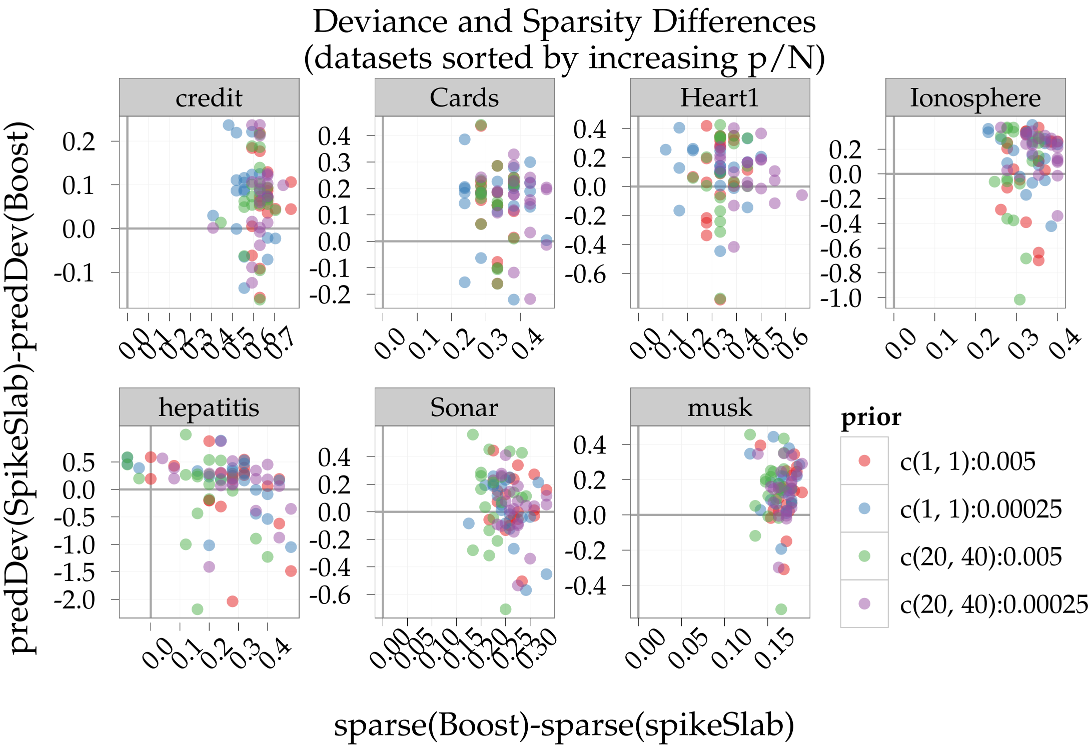


Figure 4.22.: UCI data results without preselection: Difference in proportion of included model terms versus difference in predictive deviance. Points in topright quadrant denote folds and prior settings in which our approach achieved smaller deviances with fewer included model terms. Points in lower 2 quadrants denote folds/priors in which our approach resulted in larger deviances than the corresponding `mboost`-fits.

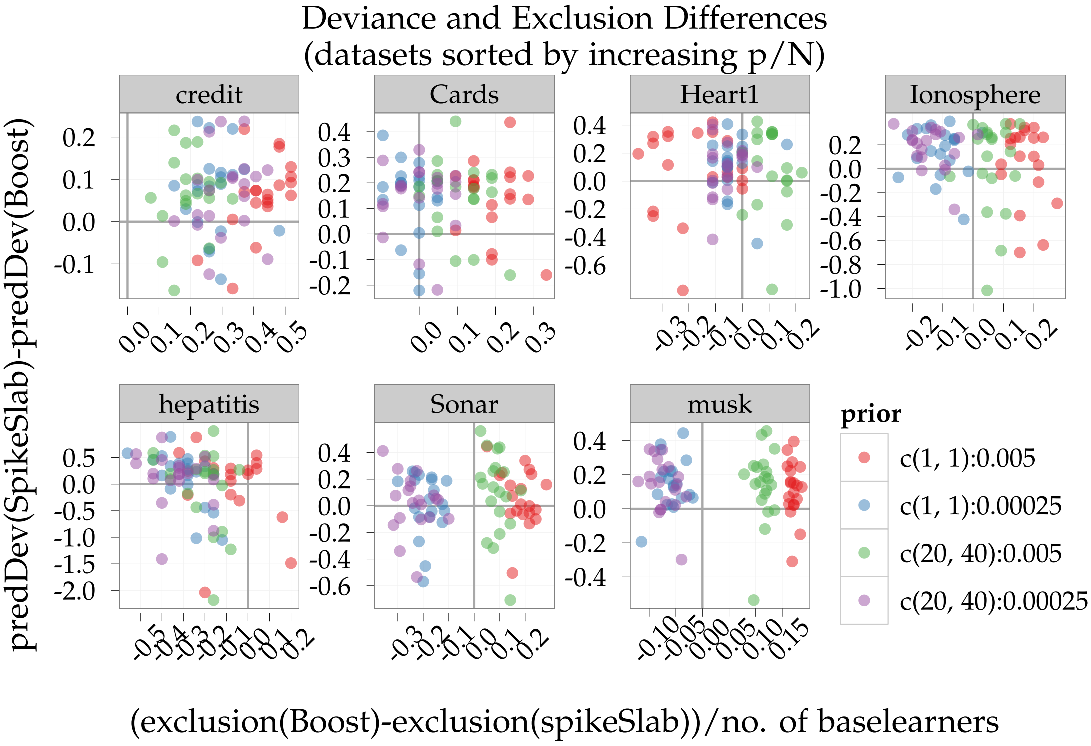


Figure 4.23.: UCI data results without preselection: Difference in proportion of unequivocally excluded model terms versus difference in predictive deviance. Points in topright quadrant denote folds and prior settings in which our approach achieved smaller deviances with sparser models. Points in lower 2 quadrants denote folds/priors in which our approach resulted in larger deviances than the corresponding `mboost`-fits.

relatively more precise predictions with smaller models on the large majority of the benchmarked data sets and cross validation folds. Table 4.4 gives the median deviances and AUCs (area under the ROC-curve) for the different datasets and priors.

Using an informative prior for w to enforce model sparsity has no appreciable effect and does not influence prediction quality in either direction.

More generally, the performance of peNMIG on the binary classification datasets we used is very competitive to componentwise boosting and neither relative nor absolute performance suffer in very high-dimensional problems with many smooth terms (cf. results for `musk` with $n = 476$ and 332 potential model terms, of which 166 are smooth terms). Results for an earlier, more rudimentary implementation of the peNMIG model on identical data are published in (Scheipl, 2010b).

Dataset	(a_τ, b_τ)	v_0	median($\bar{D}$)			median(AUC)		
			mboost, simple	mboost	spikeSlabGAM	mboost, simple	mboost	spikeSlabGAM
Circle	(5, 25)	0.00025	0.26	0.41	0.04	1.00	1.00	1.00
	(5, 25)	0.005			0.04			1.00
	(5, 50)	0.00025			0.04			1.00
	(5, 50)	0.005			0.04			1.00
Spirals	(5, 25)	0.00025	0.69	0.98	0.45	0.96	0.94	0.97
	(5, 25)	0.005			0.43			0.97
	(5, 50)	0.00025			0.45			0.97
	(5, 50)	0.005			0.46			0.97
titanic	(5, 25)	0.00025	1.09	1.09	1.01	0.68	0.68	0.67
	(5, 25)	0.005			1.03			0.66
	(5, 50)	0.00025			1.04			0.68
	(5, 50)	0.005			1.04			0.68
PimaDiab	(5, 25)	0.00025	1.05	1.04	0.99	0.85	0.84	0.84
	(5, 25)	0.005			0.96			0.85
	(5, 50)	0.00025			0.98			0.84
	(5, 50)	0.005			0.96			0.85
chess	(5, 25)	0.00025	0.60	0.60	0.17	0.99	0.99	1.00
	(5, 25)	0.005			0.16			1.00
	(5, 50)	0.00025			0.16			1.00
	(5, 50)	0.005			0.17			1.00
liver	(5, 25)	0.00025	1.19	1.18	1.17	0.81	0.79	0.74
	(5, 25)	0.005			1.12			0.79
	(5, 50)	0.00025			1.17			0.75
	(5, 50)	0.005			1.12			0.78
monks3	(5, 25)	0.00025	0.42	0.42	0.12	1.00	1.00	1.00
	(5, 25)	0.005			0.10			1.00
	(5, 50)	0.00025			0.11			1.00
	(5, 50)	0.005			0.10			1.00
ringnorm	(5, 25)	0.00025	0.30	0.89	0.08	1.00	0.98	1.00
	(5, 25)	0.005			0.08			1.00
	(5, 50)	0.00025			0.08			1.00
	(5, 50)	0.005			0.09			1.00
threenorm	(5, 25)	0.00025	0.83	0.86	0.69	0.93	0.93	0.93
	(5, 25)	0.005			0.68			0.93
	(5, 50)	0.00025			0.68			0.94
	(5, 50)	0.005			0.67			0.94
tictactoe	(5, 25)	0.00025	1.03	1.03	0.11	0.90	0.90	1.00
	(5, 25)	0.005			0.11			1.00
	(5, 50)	0.00025			0.11			1.00
	(5, 50)	0.005			0.11			1.00
twonorm	(5, 25)	0.00025	0.26	0.33	0.10	1.00	1.00	1.00
	(5, 25)	0.005			0.09			1.00
	(5, 50)	0.00025			0.09			1.00
	(5, 50)	0.005			0.09			1.00
credit	(5, 25)	0.00025	1.09	1.08	1.01	0.79	0.79	0.79
	(5, 25)	0.005			1.00			0.79
	(5, 50)	0.00025			1.01			0.78
	(5, 50)	0.005			1.00			0.79
Cards	(5, 25)	0.00025	0.76	0.75	0.62	0.94	0.94	0.94
	(5, 25)	0.005			0.60			0.94
	(5, 50)	0.00025			0.63			0.94
	(5, 50)	0.005			0.60			0.94
Heart1	(5, 25)	0.00025	0.93	0.88	0.81	0.93	0.94	0.90
	(5, 25)	0.005			0.73			0.93
	(5, 50)	0.00025			0.77			0.88
	(5, 50)	0.005			0.72			0.92
HouseVotes84	(5, 25)	0.00025	0.29	0.29	0.06	1.00	1.00	1.00
	(5, 25)	0.005			0.05			1.00
	(5, 50)	0.00025			0.04			1.00
	(5, 50)	0.005			0.05			1.00
Ionosphere	(5, 25)	0.00025	0.67	0.65	0.44	0.98	0.97	0.98
	(5, 25)	0.005			0.47			0.98
	(5, 50)	0.00025			0.46			0.98
	(5, 50)	0.005			0.54			0.97
BreastCancer	(5, 25)	0.00025	0.37	0.37	0.14	1.00	1.00	1.00
	(5, 25)	0.005			0.15			1.00
	(5, 50)	0.00025			0.13			1.00
	(5, 50)	0.005			0.15			1.00
hepatitis	(5, 25)	0.00025	0.76	0.75	0.53	1.00	1.00	1.00
	(5, 25)	0.005			0.50			1.00
	(5, 50)	0.00025			0.48			1.00
	(5, 50)	0.005			0.54			1.00
Sonar	(5, 25)	0.00025	1.00	0.98	0.94	0.89	0.92	0.88
	(5, 25)	0.005			0.82			0.90
	(5, 50)	0.00025			0.95			0.88
	(5, 50)	0.005			0.83			0.93
musk	(5, 25)	0.00025	0.87	0.84	0.72	0.92	0.94	0.92
	(5, 25)	0.005			0.65			0.94
	(5, 50)	0.00025			0.69			0.91
	(5, 50)	0.005			0.63			0.94
promotergene	(5, 25)	0.00025	0.57	0.59	0.38	1.00	1.00	1.00
	(5, 25)	0.005			0.19			1.00
	(5, 50)	0.00025			0.26			1.00
	(5, 50)	0.005			0.17			1.00

Table 4.2.: Median deviances and AUCs for test samples of UCI data (models with preselection).

Dataset	n	no. of terms	mean run time [sec]	
			spikeSlabGAM	mboost
Circle	1200	4	11.72	19.15
Spirals	1200	4	12.26	17.44
titanic	2201	3	17.06	11.28
PimaDiab	768	16	33.02	51.98
chess	3196	36	198.67	69.62
liver	345	12	12.23	13.00
monks3	554	6	5.64	8.15
ringnorm	1200	40	245.75	137.76
threenorm	1200	40	250.25	131.99
tictactoe	958	9	17.06	12.01
twonorm	1200	40	239.53	132.07
credit	1000	27	62.35	44.36
Cards	653	21	38.51	44.92
Heart1	296	18	12.92	17.44
HouseVotes84	232	16	5.58	13.99
Ionosphere	351	42	26.86	33.75
BreastCancer	683	9	52.89	12.24
hepatitis	80	19	2.85	14.69
Sonar	208	67	28.00	49.02
musk	476	180	295.99	166.66
promotergene	106	57	23.91	42.41

Table 4.3.: Average run times in seconds for spikeSlabGAM and mboost

Dataset	(a_w, b_w)	v_0	median($\bar{D}$)			median(AUC)		
			mboost, simple	mboost	spikeSlabGAM	mboost, simple	mboost	spikeSlabGAM
credit	(1, 1)	0.005	1.10	1.09	1.03	0.78	0.79	0.78
	(1, 1)	0.00025			1.01			0.77
	(20, 40)	0.005			1.02			0.79
	(20, 40)	0.00025			1.01			0.78
Cards	(1, 1)	0.005	0.78	0.76	0.58	0.93	0.94	0.94
	(1, 1)	0.00025			0.59			0.95
	(20, 40)	0.005			0.58			0.94
	(20, 40)	0.00025			0.60			0.94
Heart1	(1, 1)	0.005	0.93	0.82	0.76	0.92	0.94	0.91
	(1, 1)	0.00025			0.66			0.92
	(20, 40)	0.005			0.76			0.92
	(20, 40)	0.00025			0.71			0.91
Ionosphere	(1, 1)	0.005	0.62	0.63	0.50	0.99	0.98	0.98
	(1, 1)	0.00025			0.49			0.99
	(20, 40)	0.005			0.44			0.97
	(20, 40)	0.00025			0.43			0.99
hepatitis	(1, 1)	0.005	0.86	0.68	0.48	1.00	1.00	1.00
	(1, 1)	0.00025			0.38			1.00
	(20, 40)	0.005			0.45			1.00
	(20, 40)	0.00025			0.43			1.00
Sonar	(1, 1)	0.005	0.81	0.81	0.68	0.96	0.95	0.96
	(1, 1)	0.00025			0.73			0.92
	(20, 40)	0.005			0.62			0.96
	(20, 40)	0.00025			0.79			0.92
musk	(1, 1)	0.005	0.63	0.76	0.60	0.98	0.96	0.95
	(1, 1)	0.00025			0.60			0.96
	(20, 40)	0.005			0.62			0.94
	(20, 40)	0.00025			0.63			0.96

Table 4.4.: Median deviances and AUCs in test samples for UCI data. (Models without preselection)

Dataset	n	no. of terms	mean run time [sec]	
			spikeSlabGAM	mboost
credit	1000	27	55.51	58.93
Cards	653	21	34.18	60.75
Heart1	296	18	10.04	20.53
Ionosphere	351	65	57.86	73.00
hepatitis	80	25	3.59	24.22
Sonar	208	120	83.80	127.85
musk	476	332	1490.15	426.45

Table 4.5.: Average run times in seconds for spikeSlabGAM and mboost for UCI data (models without preselection)

4.3. Case study: hymenoptera venom allergy

4.3.1. Data

We reanalyze data on bee and wasp venom allergy from a large observational multicenter study previously analyzed in Ruëff et al. (2009). The data consists of 962 patients from 14 European study centers with established bee or wasp venom allergy who suffered an allergic reaction after being stung. The binary outcome of interest is whether patients suffered a severe, life-threatening reaction, defined as anaphylactic shock, loss of consciousness, or cardiopulmonary arrest. A severe reaction was observed for 206 of the 962 patients (21.4%). Data were collected on the concentration of tryptase, a potential biomarker, patients' sex and age, whether the culprit insect was a bee or wasp, on the intake of three types of cardiovascular medication (β-blockers, ACE inhibitors and anti-hypertensive drugs), whether the patient had had at least one minor systemic reaction to a sting prior to the index sting and the CAP-class (a measure of antibody load) of the patient with regard to the venom of the culprit insect, with levels $0, 1, 2, 4, 5+$.

4.3.2. Analysis

An analysis of this data has to take into account possible study center effects, possible non-linear effects of both age and the (logarithm of) blood serum tryptase concentrations and the possibility of differing effect structures for bee and wasp stings. Our aim is twofold again: We want to (1) estimate a model that allows assessment of the influence of each covariate on the susceptibility for a severe reaction, accounting for possibly nonlinear effects and interaction effects and (2) use this setting to evaluate the stability of the selection and estimation of increasingly complex models on real data as well as investigate the consequences of less-than-optimal sampler convergence we observed.

Full data analysis

We fit a peNMIG model with all main effects and all second order interactions except those with study center, with smooth functions for both age and tryptase and a random intercept for the study center. In total, this model has 267 coefficients in 66 model terms: 13 main effects including the global intercept, separate linear and non-linear terms for age and tryptase and a random intercept for study center, 21 interactions between the seven factor variables, 28 terms for the linear and smooth interactions for age and tryptase with each of the seven factors, and four terms for the interaction effect of age and tryptase (one linear-linear interaction, two varying coefficient terms, one smooth interaction surface) Results are based on samples from 20 chains with 40000 iterations each after 1000 burn-in, with every 20^{th} saved. Running a single chain of this length on a modern desktop computer (i.e., Intel Q9550 2.83GHz) takes about 45 minutes, so that the entire fit takes about 4 hours on a quad-core CPU.

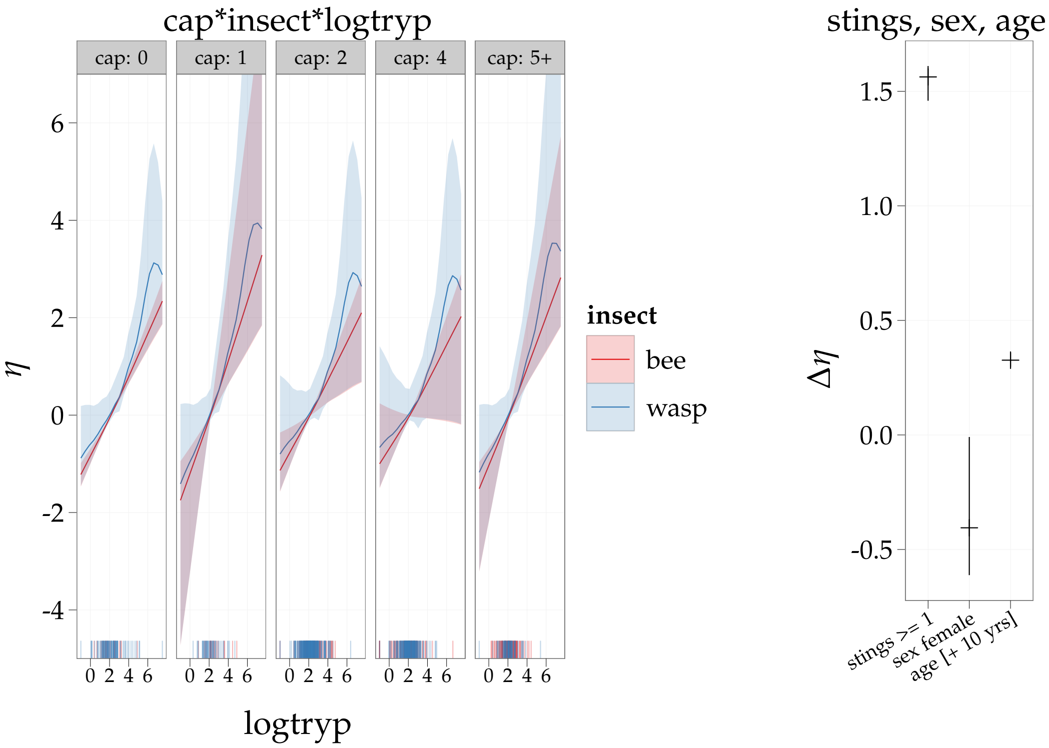


Figure 4.24.: Posterior means of effects with (pointwise) 80% credible intervals. Only effects for terms with marginal inclusion probability $> .1$ are shown. Left graph shows the joint effect of CAP class, tryptase and culprit insect. Right graph shows the relative effects of previous stings (compared to none before the index sting), female gender (compared to male) and a 10 year increase of age.

Figure 4.24 shows the estimated effects of the terms with $P(\gamma = 1) > .1$ that are listed in Table 4.6. Since the inclusion probabilities indicate interlocking interactions of CAP class, tryptase and culprit insect, the panels in the left graph in the figure show the joint effects of these 3 variables. Each panel shows the effect estimate of tryptase plasma concentration for bee patients (red) and wasp patients (blue) for the given CAP class. The rug plot at the bottom indicates the locations of the data. The large uncertainty precludes a detailed interpretation of this 3-way interaction, but in general, the risk is higher for wasp patients: the main effect of culprit insect yields an odds ratio of 1.16 (80%CI: 1-2.43) and the risk increase for wasp patients seems to be

Term	$P(\gamma = 1)$
culprit insect	0.16
stings	1.00
sex	0.70
age, linear	1.00
tryptase, log-linear	1.00
study center	0.71
insect:tryptase, smooth	0.46
cap:tryptase, log-linear	0.14

Table 4.6.: Posterior means of marginal inclusion probabilities $P(\gamma = 1)$ (only given for terms with $P(\gamma = 1) > .1$).

smaller for lower and larger for higher tryptase concentrations. The effect of the CAP class is negligible.

The graph on the right in Figure 4.24 shows the relative effects of previous stings (compared to none before the index sting), female gender (compared to male) and an increase in the patient's age by 10 years. Estimated random effects for the study centers are not shown, the associated posterior mean odds ratios range between 0.44 and 2.13.

Lack of convergence for γ

For this fairly complicated model, we experience some difficulties with the convergence of the MCMC sampler: We observe poor mixing for some of the entries in γ, with chains getting stuck in basins of attraction around posterior modes for long periods of time. This leads to posterior inclusion probabilities for single chains often ending up either close to zero or close to one for some of the terms. Running a large number of parallel chains from random starting configurations seems to remedy this problem. To investigate this issue, we perform a large MCMC experiment with 800 chains, each with 10000 iterations after 100 burn-in, for the model described above. Figure 4.25 shows the average inclusion probabilities for the 16 terms with the highest between-chain variability of $P(\gamma = 1)$ for 20 fits with 40 chains each. Grey lines connect posterior means based on an increasing number of chains for each fit. The black horizontal line shows the mean over all 800 chains, which we presume to be a good estimate of the "true" marginal posterior inclusion probability. Convergence of the posterior means is slow for these terms, but discrimination between important, intermediate and negligible effects seems to be reliable based on as few as 10 to 20 chains. While we would not be comfortable in claiming that 10 or 20 parallel chains are enough to completely explore this very high-dimensional model space and yield a reliable estimate

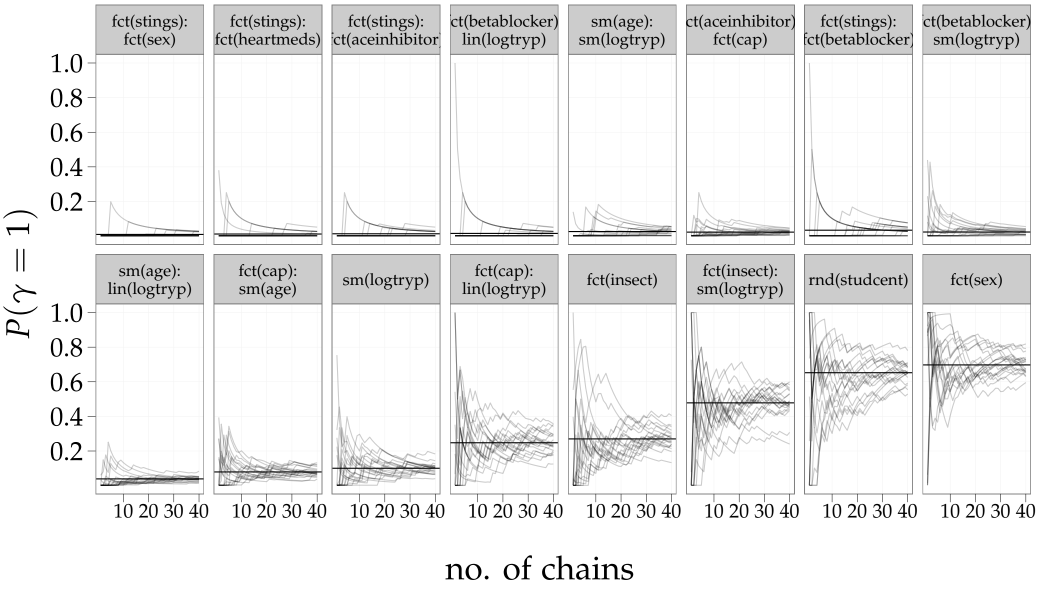


Figure 4.25.: Average inclusion probabilities for those terms with convergence issues for 20 fits with 40 chains each. Grey lines connect posterior means over an increasing number of chains for each fit. Black horizontal line shows the mean over all 800 chains.

of posterior model probabilities, i.e., the joint distribution of $\boldsymbol{\gamma}$, the marginal inclusion probabilities $P(\gamma_j = 1)$, $j = 1, \ldots, p$ of the various terms seem to be estimated well enough to distinguish between important, intermediate and negligible effects, which is usually all that is required in practice.

Predictive performance comparison

We subsample the data 20 times to construct independent training data sets with 866 subjects each and test data sets with the remaining 96 patients to evaluate the precision of the resulting predictions and compare predictive performance to that of equivalent component-wise boosting models fitted with `mboost` and an unregularized GAMM-fit with all main effects estimated with `gamm4` (Wood, 2010b). Results for our approach are based on 8 parallel chains each running for 10000 iterations after 500 iterations of burn-in, with every 10^{th} iteration saved. Component-wise boosting results are based on a stopping parameter determined by a 25-fold bootstrap of the training data, with a maximal iteration number of 500. We compare 3 model specification of increasing complexity: a simple model with main effects only, a model with main effects and all interactions between culprit insect and the other covariates, and the complex model with all main effects and second order interactions presented in the previous section. We were unable to fit the latter model with `mboost`, and the model including the insect interactions could not be fitted by `mboost` for 4 of the training data sets. We report results for the 16 sets remaining. Figure 4.26 shows the area under the ROC curve (AUC) achieved by the different model specifications. For this data set, the models with higher maximal complexity show slight decreases in predictive accuracy, but still perform better than an unregularized generalized additive mixed model (GAMM) on the far right.

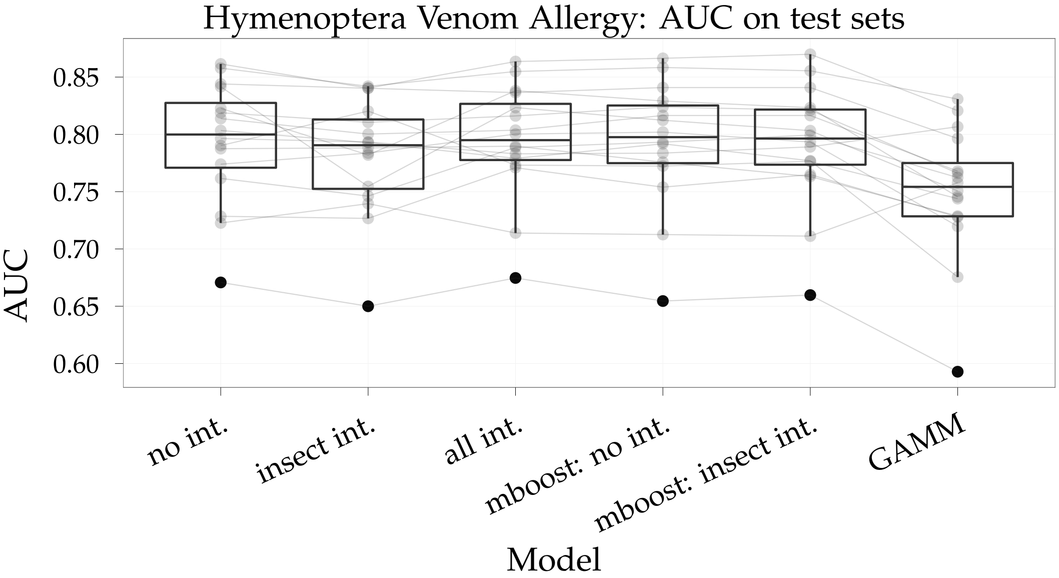


Figure 4.26.: Area under the ROC curve for 20 test sets from the hymenoptera venom allergy data set, higher is better. Grey lines connect results from identical folds.

Despite the fairly low number of parallel chains and comparatively short chain lengths, the stability of the marginal term inclusion probabilities across subsamples is fairly good, indicating that the term selection is robust to small changes in the data and that even as few as 8 chains may be enough to reach fairly reliable rough estimates of term importance in this difficult setting. All model specifications identified the same subset of important main effects (i.e., number of previous stings before the index sting, sex, linear effects of age and the log of tryptase and the random effect for study center). Figure 4.27 shows the posterior means of inclusion probabilities $P(\gamma = 1)$ across 16 subsampled training data sets for each of the 3 model specifications (from left to right: no interactions, all culprit insect two-way interactions, all two-way interactions).

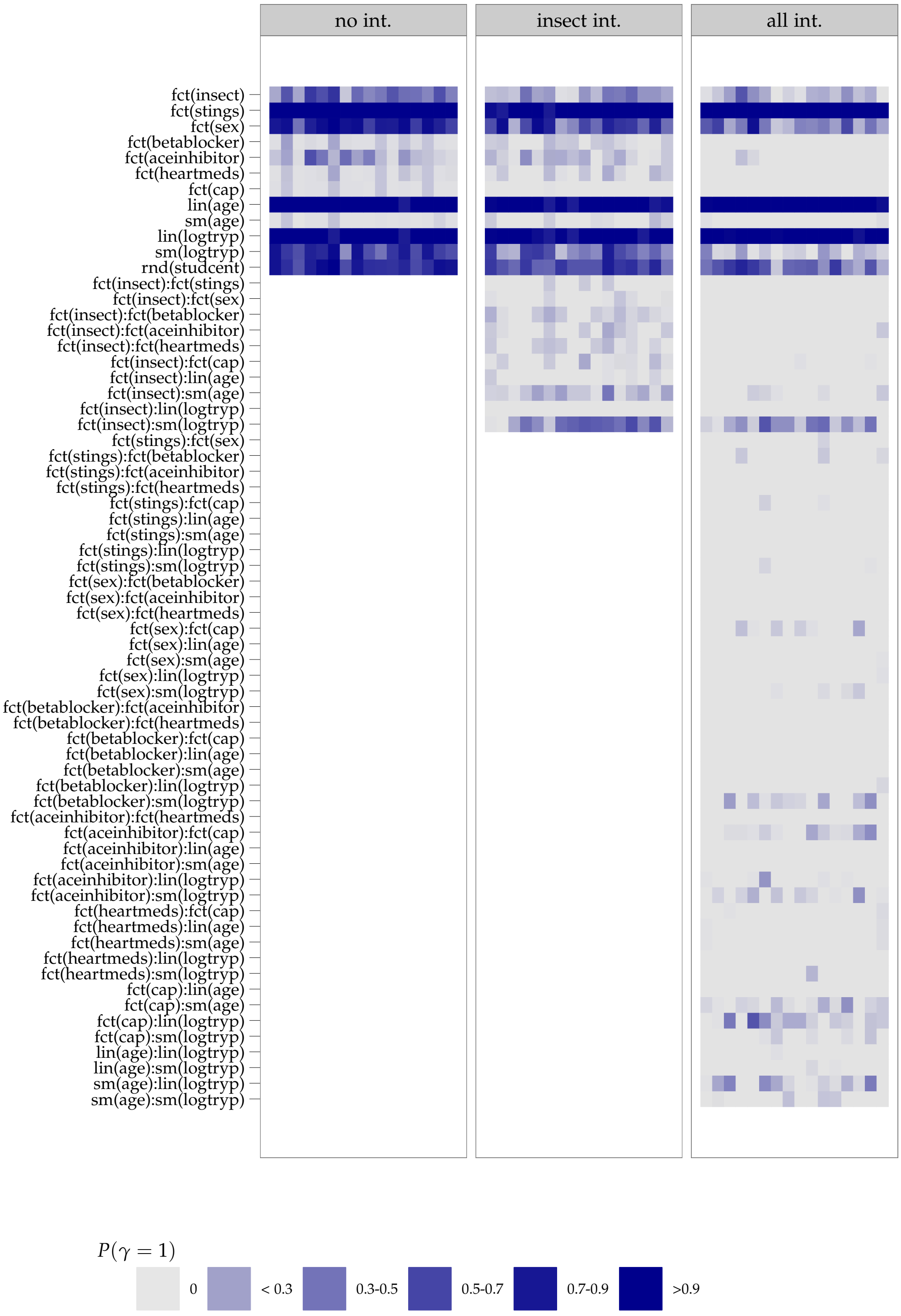


Figure 4.27.: Posterior means of inclusion probabilities $P(\gamma = 1)$ across 16 subsampled training data sets for the 3 model specifications.

4.4. Case study: Survival of surgery patients with severe sepsis

4.4.1. Data

We use data on the survival of 462 patients with severe sepsis collected in the intensive care unit of the Department of Surgery, Campus Großhadern, LMU Munich, Germany between March 1, 1993, and February 28, 2005. Hofner, Kneib, Hartl, and Küchenhoff (2010) have previously analyzed this data set. The follow-up period was 90 days after the beginning of intensive care, with one drop-out after 66 days and 179 patients surviving the observation period.

4.4.2. Model

We use a piecewise exponential model (PEM) (Laird and Olivier, 1981) to model the hazard rate $\lambda(t, \boldsymbol{x})$ of the underlying disease process, i.e. for fixed time intervals defined by cutpoints $\boldsymbol{\kappa} = (\kappa_0 = 0, \kappa_1, \ldots, \kappa_I = t_{\max})$, where $t_{\max}$ is the maximal follow-up time, the hazard rate for subject i at time t, $\kappa_{j-1} < t \leq \kappa_j$ in the j^{th} interval is given by $\lambda(t, \boldsymbol{v}_i) = \lambda_j \exp(\boldsymbol{\eta}_i)$, where λ_j is the baseline hazard rate in the respective interval and $\boldsymbol{\eta}_i$ is the predictor for subject i. The interval borders $\boldsymbol{\kappa} = (0, 5, 15, 25, \ldots, 85, 90)$ were chosen based on the shape of a nonparametric estimate of the marginal hazard rate from R package `muhaz` (Gentleman, 2010). The likelihood for this model is equivalent to that of a Poisson model with (1) one observation for each interval for each subject, yielding 2826 pseudo-observations in total, (2) offsets $o_{ij} = \max(0, \min(\kappa_j - \kappa_{j-1}, t_i - \kappa_{j-1}))$, where t_i is the observed time under risk for subject i and (3) responses y_{ij} equal to the event indicators δ_{ij}, with $\delta_{ij} = 0$ if subject i survived interval j and $\delta_{ij} = 1$ if not (Friedman, 1982).

4.4.3. Analysis

Our aim is twofold: We want to (1) estimate a model that allows assessment of the influence of each covariate on the prognosis of patients, accounting for possibly time-varying and/or nonlinear effects and (2) use this setting to evaluate the stability of the selection and estimation of increasingly complex models on real data.

Full data results

We perform term selection for a maximal model which includes the (linear and non-linear) effects of all 20 covariates as well as their time-varying ef-

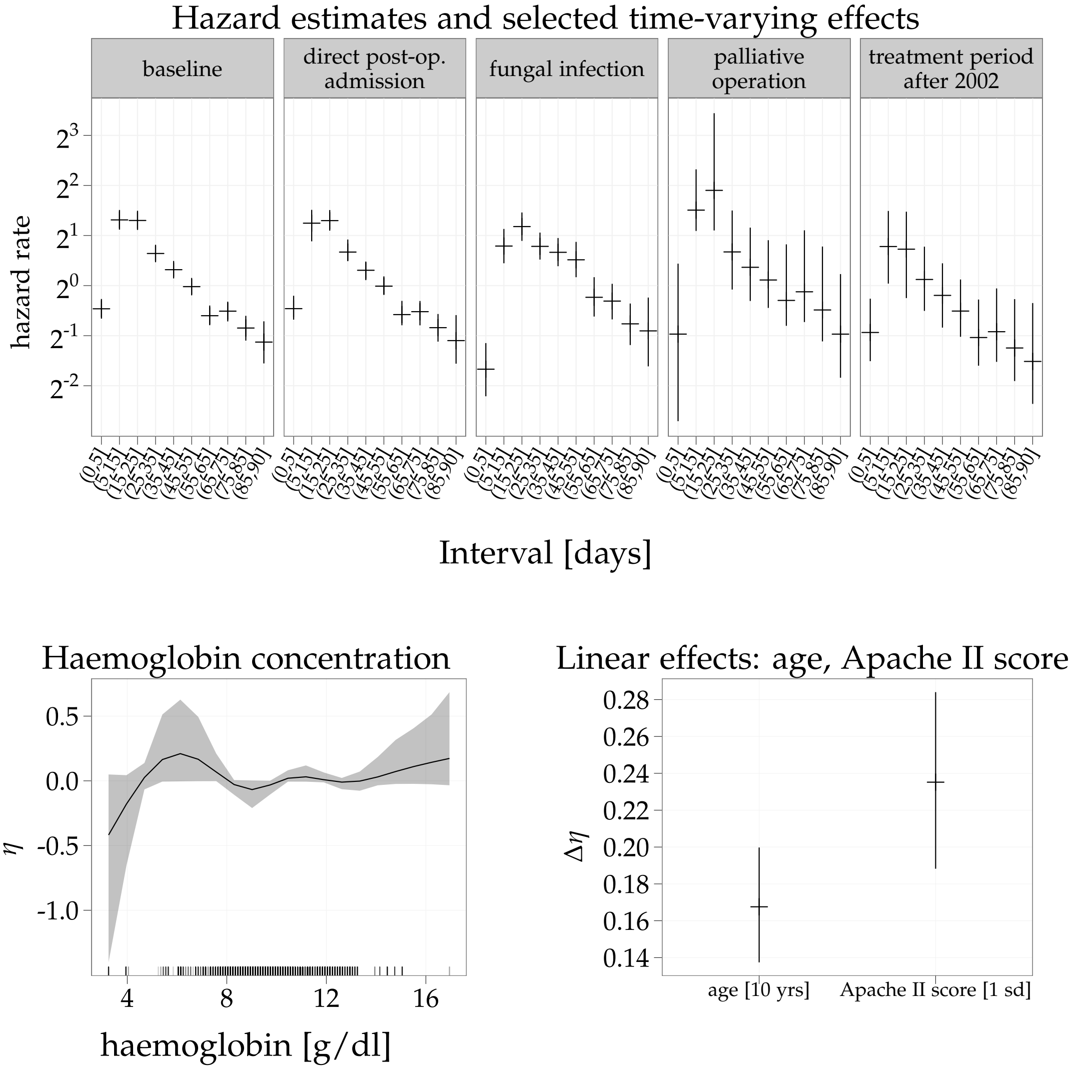


Figure 4.28.: Posterior means of effects with (pointwise) 80% credible intervals. Top: Baseline hazard rate and baseline hazard rate plus the time-varying and time-constant effects for direct postoperative admission, presence of a fungal infection, palliative operation and beginning of treatment after 2002. Bottom: smooth effect of haemoglobin concentration and linear effects of age (10 year increase) and Apache II score, a measure for disease severity (increase of score by 1 standard deviation).

Term	$P(\gamma = 1)$
MRF(Interval)	1.00
palliative operation	0.19
treatment period	0.71
Age, linear	0.99
Apache II score, linear	1.00
Haemoglobin concentration, smooth	0.38
MRF(Interval):direct postoperative admission	0.28
MRF(Interval):fungal infection	1.00
MRF(Interval):palliative operation	0.38
MRF(Interval):treatment period	0.13

Table 4.7.: Posterior means of marginal inclusion probabilities $P(\gamma = 1)$ (only given for terms with $P(\gamma = 1) > .1$).

fects, i.e. 48 potential model terms with 262 coefficients in total. Hyperparameters were set to the default values determined in the simulation studies, i.e. $a_\tau = 5, b_\tau = 25, v_0 = 0.00025$ and $a_w = b_w = 1$. Estimates are based on 8 parallel chains running for 20000 iterations each after a burn-in of 500 iterations, with every 10^{th} iteration saved. We use a first order random walk prior on the interval-specific log baseline hazard rates $\log(\lambda_j)$ in order to regularize the baseline hazard's roughness, i.e. we use an intrinsic GMRF prior for the piecewise constant log baseline hazard.

The estimated marginal inclusion probabilities indicate a fairly sparse model, with posterior marginal inclusion probabilities greater than 0.10 for only 10 terms, as shown in Table 4.7. The estimated effects for this subset of terms are visualized in Figure 4.28. To verify the suitability of the model, we perform a posterior predictive check and generate 100 replicates of survival times from the posterior predictive. Figure 4.29 indicates that the fit is satisfactory, although there seems to be a tendency to overestimate survival rates until about day 70.

Predictive performance comparison

We subsample the data 20 times to construct independent training data sets with 415 patients each and test data sets with the remaining 47 patients to evaluate the precision of the resulting predictions and compare predictive performance to that of equivalent component-wise boosting models fitted with `mboost`. Results for our approach are based on 8 parallel chains each running for 5000 iterations after 500 iterations of burn-in, with every 5^{th} iteration saved. Component-wise boosting results are based on a stopping parameter determined by a 25-fold bootstrap of the training data, with a maximal itera-

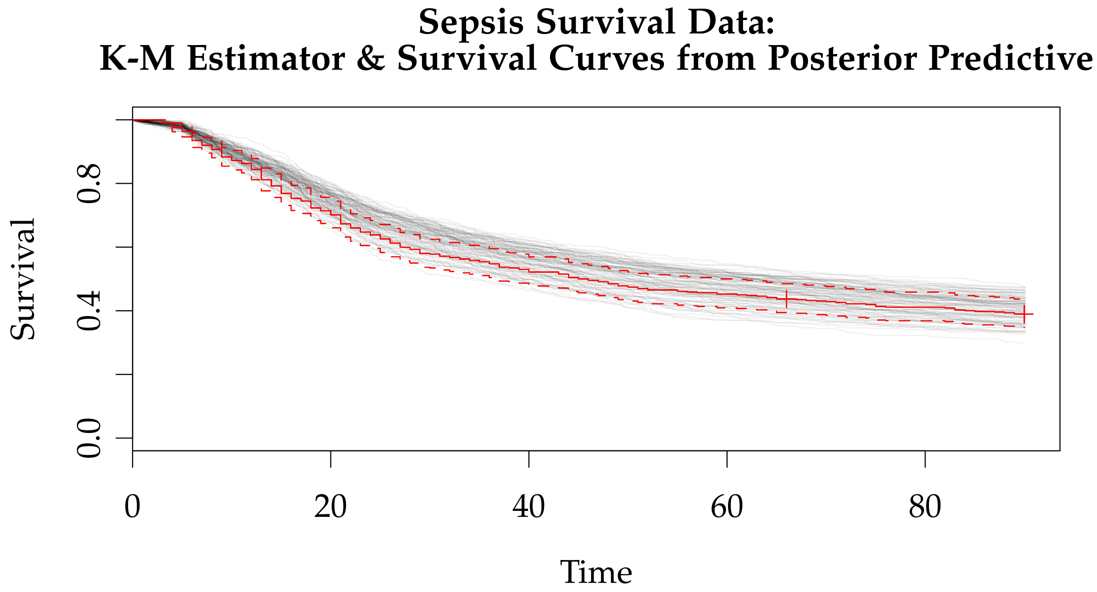


Figure 4.29.: Kaplan-Meier estimate of the survival curve for observed data in red. Grey overlays are survival curves for 100 replicates of survival time vectors generated from the posterior predictive distribution.

tion number of 1500.

The previous analysis by Hofner et al. (2010) has used expert knowledge to define a set of six covariates forced into the model (indicators for presence of malignant primary disease, palliative operation and beginning of treatment after 2002, as well as sex, age and Apache II score). We compare results for four model specifications of increasing complexity that suggest themselves: a model with only the main effects of the pre-selected covariate set, a model with main effects and time-varying effects for the pre-selected covariate set, a model with main effects for all 20 covariates and the model with main effects and time-varying effects for all 20 covariates which was applied to the whole data set (see above). As in the previous section, main effects for numerical covariates such as age were split into linear and non-linear parts. Figure 4.30 shows the predictive deviances achieved by the different model specifications. Predictive deviance is defined as $-2\sum_i^{N_t}\sum_j^{J(i)} \delta_{ij}(\log(\hat{\lambda}_j) + \hat{\eta}_{ij}) - o_{ij}\hat{\lambda}_j \exp(\hat{\eta}_{ij})$, where $i = 1, \dots, N_t$ indicates the subjects in the test set and $j = 1, \dots, J(i)$ indicates the intervals in which individual i was under risk, $\hat{\lambda}_j$ and $\hat{\eta}_{ij}$ are the respective posterior predictive means. For this data set, models with higher maximal complexity seem to offer no relevant improvement in terms of prediction accuracy compared to the simplest model based only on the pre-selected covariate set without time-varying effects. Most of the models yield

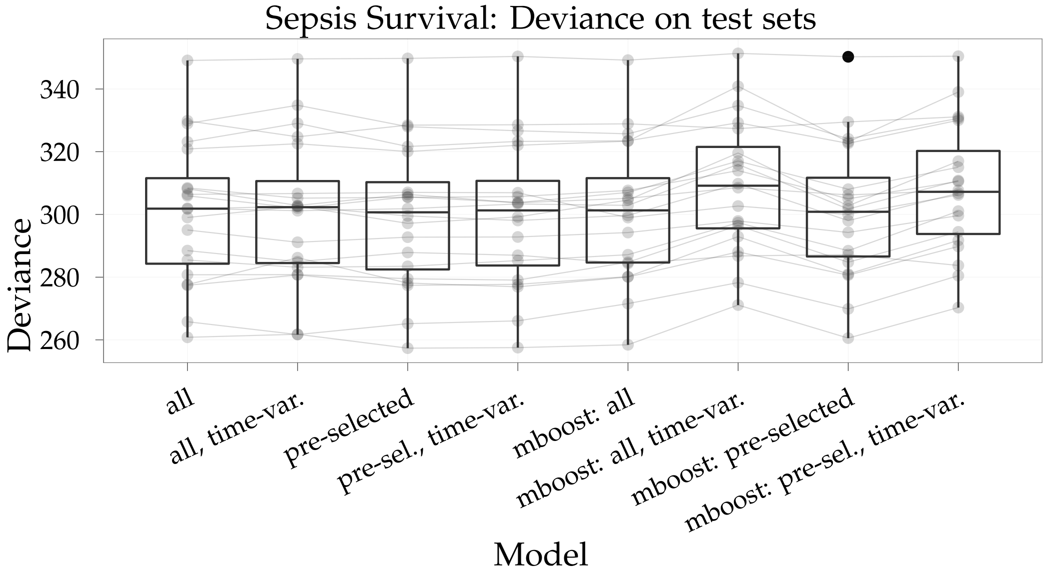


Figure 4.30.: Predictive deviances for 20 subsampling test sets for the sepsis survival data (lower is better). Grey lines connect results from identical folds.

essentially equivalent predictions. However, it is reassuring to see that the predictive performance of our approach is not degraded at all by the specification of vastly over-complex models in a setting where the underlying structure seems to be fairly simple. In contrast, prediction accuracy for component-wise boosting decreases markedly for the models including time-varying effects in this setting.

The stability of the marginal term inclusion probabilities across subsamples is fairly good, indicating that the term selection is robust to small changes in the data. All model specifications identified essentially the same subset of important effects from the set of pre-selected covariates (i.e., indicators for palliative operation and beginning of treatment after 2002 and linear effects of age and Apache II score), and also the same time-varying effects (i.e., time varying effects for palliative operation and beginning of treatment after 2002). Figure 4.31 shows the posterior means of inclusion probabilities $P(\gamma = 1)$ across 20 subsampled training data sets for each of the 4 model specifications (from left to right: pre-selected covariates only, all covariates, pre-selected covariates with time-varying effects, all covariates with time-varying effects).

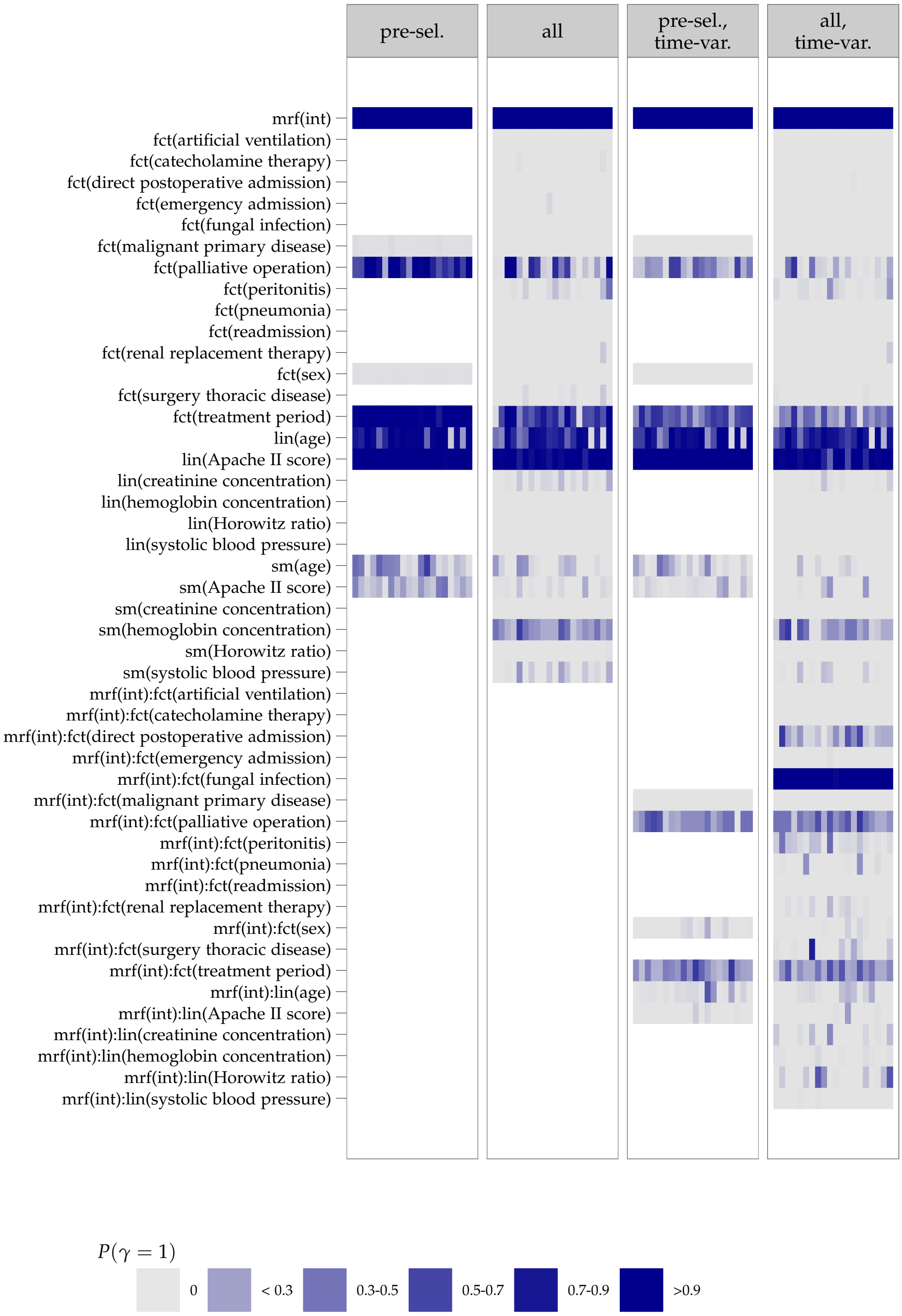


Figure 4.31.: Posterior means of inclusion probabilities $P(\gamma = 1)$ across 20 subsampled training data sets for the 4 model specifications.

5. Discussion

5.1. Results

Part I was focussed on the shrinkage and model selection properties of the peNMIG prior structure in structured additive regression models. By introducing a non-identifiable multiplicative parameter expansion combined with a spike-and-slab prior on the level of a common scaling factor, we are able to select or deselect coefficient batches (i.e., coefficients for a spline basis or random intercepts associated with a grouping factor) simultaneously in order to guide model choice for generalized additive mixed models.

Extensive simulation studies and application examples showed that the performance of the proposed approach is at least competitive and frequently superior to recently proposed adaptive shrinkage priors and frequentist approaches that address estimation and selection of model terms simultaneously such as component-wise boosting or adaptive COSSO, cf. Sections 4.1.5, 4.2. Our approach also additionally yields estimates of the (marginal) inclusion probabilities for each term.

Estimation performance was very robust against different hyperparameter configurations in all the settings we considered. Variable selection and model choice were more sensitive to varying hyperparameters, but we are confident that the collected simulations and application examples provide a solid foundation for the choice of appropriate values for applied problems. For example, results for the selection and estimation of random intercepts and smooth univariate functions indicate prior settings that yield desirable long-run operating characteristics similar to those of the RLRT (cf. Sections 4.1.3, 4.1.4) for Gaussian responses. Our simulation studies also revealed that selection of model terms with large coefficient vectors, such as random effects, remains an unsolved problem for non-Gaussian response – our results indicate that selection is extremely liberal in this context, especially so for binary response (cf. Figures 4.6, 4.7).

Our approach is implemented in the R-package `spikeSlabGAM`. The conjugacy structure of the proposed prior hierarchy allows for fast and very stable fully Bayesian inference based on MCMC sampling. The implementation is able to make use of increasingly common multi-core processors for parallel sampling of multiple chains, which aid in judging convergence and exploring the viable parameter space more quickly when starting from dispersed starting values.

In its current state, `spikeSlabGAM` allows fitting geoadditive mixed models for Gaussian, Binomial and Poisson responses, with predictors that can contain any number and combination of uni- or multivariate smooth terms, GMRFs, random intercepts, factors and simple linear terms. `spikeSlabGAM` also implements a fairly general framework to construct and estimate interaction effects of penalized and/or unpenalized terms via tensor products of the null and range spaces of the respective main effects and lower order interactions for even more flexible model specification. The package uses the established R formula syntax, which allows complex models to be specified in a very concise fashion. We hope that the familiar syntax, combined with powerful and user-friendly visualizations of the fitted models, results in a low barrier to entry for application of this new tool in practice.

5.2. Outlook

We see many worthwhile avenues of further research to pursue on the basis of what is missing from the present work.

1. The present work lacks a systematic investigation of the selection and estimation of interaction effects, an important topic in practice especially for large hierarchical models with many potential level-specific effects. Presumably, estimation of inclusion probabilities for these types of effects with large coefficient vectors will suffer from similar defects as it does for random effects for generalized responses. We feel that this is where the most important challenge for the success of the proposed method lies. Further research into more suitable MCMC methods for better mixing in these scenarios is called for.

2. There is a large literature on Bayesian model averaging (BMA) (Hoeting, Madigan, Raftery, and Volinsky, 1999) for linear and generalized linear models, with very few applications to (and no general implementations for) semiparametric models such as the ones considered here. Since there are some fairly general optimality results for the long-run performance of BMA (Raftery and Zheng, 2003) and a fairly comprehensive implementation for GLMs exists (R-package `BMA`, Raftery, Hoeting, Volinsky, Painter, and Yeung (2011)), it would be interesting to compare the performance of BMA with our approach for the subclass of models for which BMA is implemented.

3. As far as the implementation is concerned, we see a lot of potential for generalizing the approach to an even broader class of potential model terms as well as response distributions:
   The case study for the sepsis survival data in Section 4.4 shows that an extension for piecewise exponential time-to-event models via data augmentation is easily done within the constraints of the existing implementation. In a similar fashion, data augmentation along the lines of Begg and Gray (1984) could be used to fit polychotomous responses with `spikeSlabGAM`. An extension of the methodology for accelerated failure time models may also be considered a "low-hanging fruit": Since some error distributions for the logarithm of the times-to-event like the logistic distribution can be parameterized as scale mixtures of normals and censored observations can be imputed in an additional imputation step, so that the rest of the algorithm could remain unchanged. Similar extensions are conceivable for other error distributions such as penalized Gaussian mixtures as in Komárek, Lesaffre, and Hilton (2005) or any other error distributions that can be framed as Gaussian (scale) mixtures.

4. On the predictor side, the package would benefit from added capabilities for "always included" semiparametric terms as well as allowing interaction effects to be fit in the absence of the corresponding main effects. In a similar vein, adding the option to sample inclusion indicators under hierarchical constraints, i.e., never including an interaction if the associated main effects are excluded from the model could help in finding parsimonious models that are easy to interpret. A more challenging extension could tackle the issue of covariance selection, i.e., model choice for the covariance structures of correlated random effects. It may be possible to incorporate recent approaches (Cai and Dunson, 2006; Frühwirth-Schnatter and Tüchler, 2008) for covariance selection based on a Cholesky decomposition of the random effect covariance or correlation matrix into our framework.

5. Uncovering and dealing with concurvity is especially pertinent for the high-dimensional additive models that we consider to be the main area of application for our approach. While the orthogonalization of interaction effects with regard to their parent main effects (cf. Section 3.1.4) may alleviate some concurvity problems, established approaches for the diagnosis of concurvity (cf. Gu, 2002, Ch. 3.6) are presently missing in `spikeSlabGAM` and possible remedies such as using only partial effects of nuisance variables (i.e., effects that are projected into the complement of the column space of the variables of interest along the lines of Hodges and Reich (2010) and Hughes and Haran (2011)) need to be investigated

and implemented in order to further improve the stability and interpretability of results.

Overall, we are confident that the methodological developments described in this work and implemented in `spikeSlabGAM` provide a suitable stepping stone for further refinement and generalization. Much work remains to be done before a truly general and computationally feasible framework for reliable term selection in structured additive regression is in place.

Part II.

Bayesian Model Choice for Locally Adaptive Splines

6. Locally Adaptive Bayesian P-Splines with a Normal-Exponential-Gamma Prior

In many regression applications, the assumption of a linear dependence of the response on predictor variables is inappropriate. One appealing solution to the problem of modeling smooth functions of an unknown shape, that is, fitting models of the form

$$\boldsymbol{y} = f(\boldsymbol{x}) + \varepsilon; \quad \varepsilon \sim \mathcal{N}_n(0, \sigma_\varepsilon^2 \boldsymbol{I}),$$

where $f(\cdot)$ is a smooth function of a covariate x, is P-spline smoothing (Eilers and Marx, 1996). The idea behind this approach is conceptually simple: The unknown function is approximated by a piecewise polynomial function subject to some differentiability constraints at the interval boundaries. The resulting function can be represented as a linear combination of B-spline basis functions, i.e. basis functions with local support. The number of basis functions must be large enough to allow for sufficient flexibility in the shape of the estimated function. However, due to the high dimension of the basis, an unregularized fit would result in a very variable estimate. In order to avoid this overfitting problem, the basis coefficients are penalized to enforce smoothness of the resulting fit. Let $\boldsymbol{X}$ denote the matrix of the J basis functions, evaluated at $\boldsymbol{x}$. The objective function for the P-spline fit is then the penalized least squares criterion

$$\frac{1}{\sigma_\varepsilon^2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \frac{1}{\tau^2}\|\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}\|^2 \to \min_{\boldsymbol{\beta}},$$

where $\boldsymbol{\Delta}^{(d)}$ of dimension $(J-d) \times J$ is the d^{th}-degree difference operator matrix and τ^2 is the smoothing parameter controlling the amount of penalization. In effect, this form of penalization penalizes deviations of the fitted curve from a $(d-1)$-degree polynomial (Eilers and Marx, 1996) since the d^{th} order derivative of B-splines essentially depends on d^{th} order differences. From a Bayesian perspective, d^{th} order differences correspond to a Gaussian random walk prior of order d for the vector $\boldsymbol{\beta}$ (Lang and Brezger, 2004).

For functions with locally varying complexity (e.g. oscillations with varying frequency and/or amplitude, or functions with discontinuities), a global penalty with constant smoothing parameter over the range of $\boldsymbol{x}$ is inappropriate, as it would lead to overfitting in the smooth parts of the function and underfitting in the more wiggly or discontinuous parts of the function. This problem can be tackled by introducing a penalty that varies spatially in order to reflect the spatial heterogeneity of the function. Previous suggestions for locally adaptive smoothing include Bayesian and frequentist approaches that allow for (smoothly varying) spatial heterogeneity by fitting a smooth penalty function $\tau(x)$ represented as a second P-spline (Ruppert and Carroll, 2000; Baladandayuthapani, Mallick, and Carroll, 2005; Krivobokova, Crainiceanu, and Kauermann, 2008), or reweighting the individual penalty terms so that $\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}\right)_i \sim \mathcal{N}(0, \frac{\tau^2}{\delta_i})$, with $\delta_i \sim \Gamma(\frac{\nu}{2}, \frac{\nu}{2})$ leading to a marginally t-distributed random walk prior (e.g. Lang and Brezger, 2004), as well as knot-selection based approaches (Denison, Mallick, and Smith, 1998; Biller, 2000; Dimatteo, Genovese, and Kass, 2001). Jullion and Lambert (2007) investigate robust specifications of the Bayesian P-spline prior based on hyperpriors on the roughness penalty, discrete mixture priors for the smoothing variance, and adaptive priors with locally varying smoothing parameter. Crainiceanu, Ruppert, Carroll, Joshi, and Goodner (2007) extend the approach developed by Ruppert and Carroll (2000) to multivariate additive models with heteroscedastic errors.

The main idea of our fully Bayesian approach is to replace the homoscedastic Gaussian random walk prior for $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ with a heteroscedastic heavy-tailed random walk prior. Unlike Lang and Brezger (2004), we assume piecewise constant variances, and, unlike the solutions based on the original idea by Ruppert and Carroll (2000), we make no smoothness assumptions about the shape of the resulting variance function. The prior we use is a scale mixture of normals introduced by Griffin and Brown (2007) where the variance of the normal follows an exponential distribution with a gamma-distributed rate resulting in a Normal-Exponential-Gamma (NEG) prior for the differences of P-spline coefficients. This mixture distribution is strongly peaked in the origin and has heavy tails leading to advantageous adaptivity properties.

We propose a hierarchy of estimation schemes based on Markov chain Monte Carlo simulation (MCMC) techniques that introduce increasing flexibility in estimating the variance step function. Starting with fixed number and locations of the changepoints, we then introduce a more flexible alternative in which the locations of the changepoints are estimated as well. In a final step, the number of steps is included as a further unknown parameter, leading to a reversible jump MCMC algorithm. All the NEG-based algorithms are implemented in R (R Development Core Team, 2010), the code is available from the first author's website (`http://www.statistik.lmu.de/~scheipl/`).

Results from an extensive simulation study show that the NEG-based approaches can deal equally well with both smoothly varying local complexity and functions with discontinuities and usually converge quite fast due to the excellent mixing properties of the proposed samplers. The reversible jump algorithm is almost fully automatic in the sense that results are robust against changes in the two hyperparameters supplied by the user. The applicability of the proposed approach is demonstrated in two applications: the estimation of fractionation curves in quality control of cDNA microarray experiments and the estimation of neuronal firing rates.

The rest of this part is structured as follows: Chapter 7 describes the hierarchy of our model and discusses the three implementations of our approach as well as an extension to non-Gaussian responses. Results of a fairly extensive simulation study are summarized in Chapter 8, followed by exemplary applications to real data in Chapter 9.

7. Models and algorithms

Conventional Bayesian P-spline smoothing (Lang and Brezger, 2004; Jullion and Lambert, 2007) is based on a homoscedastic Gaussian prior for the d^{th} differences $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ of the J P-spline coefficients $\boldsymbol{\beta}$: $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta} \sim N(\mathbf{0}, \tau^2 \boldsymbol{I}_{J-d})$. This corresponds to a ridge-type regularization of the fitted function, leading to a proportional shrinkage of the unregularized random walk. An improved prior distribution, however, should be designed to allow for high penalization in areas with low variability and, vice versa, low penalization in areas with high curvature or discontinuities. Translated to the form of the prior distribution this means that a prior with a peak at zero on the one hand but heavy tails on the other hand should be considered. Such types of priors have received considerable attention in recent years in applications on variable selection and regularization in high-dimensional regression models (e. g. Griffin and Brown, 2005; Park and Casella, 2008). One particularly promising candidate is the Normal-Exponential-Gamma (NEG) prior by Griffin and Brown (2007) that combines the desired properties with computational convenience in a hierarchical Bayesian updating scheme.

In this framework, the prior $p_\tau(\tau^2|z)$ for τ^2 is assumed to follow an exponential distribution with rate z. This rate, in turn, is assigned a $\Gamma(a_z, b_z)$-prior, where $\Gamma(a, b)$ denotes a gamma distribution with expectation a/b and variance a/b^2. The resulting scale mixture NEG-prior for $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ has the desired properties: Its mass is concentrated around zero, with a finite spike in the origin, leading to the desired regularization properties, and yet has heavy tails which allow for the possibility of large jumps in the random walk and therefore sudden jumps or curvature changes of the fitted function. Following Griffin and Brown (2007), we set $a_z = 0.5$, since a sufficiently flexible family of distributions is obtained by letting b_z vary all by itself. The prior for b_z is a discrete uniform distribution on a $\log_{10}$-regular grid with 550 values between 10^{-3} and 10^5.

Still, assuming a homogeneous random walk with a single variance parameter for the differences of coefficients is obviously problematic for functions with locally varying complexity. To further increase adaptivity, we replace the conventional homoscedastic prior for the d^{th} differences $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ of the J P-spline coefficients $\boldsymbol{\beta}$ with a heteroscedastic prior. Specifically, we replace the sequence of identical variances for the random walk increments in $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ with a piecewise constant variance list consisting of B different values, i.e. $\tau_b^2,\ b = 1, \ldots, B$. To characterize the piecewise constant variance step function, we can either consider the changepoints of the step function (in terms of the indices of the elements of $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$) or the lengths of the constant pieces. Let $\boldsymbol{s} = (s_1, \ldots, s_{B-1})$ denote the vector of indices of interior changepoints and set $s_0 = 1$ and $s_B = J - d$. From the changepoints we can derive the lengths of the intervals by applying the first order difference operator to the vector $(1, \boldsymbol{s}, J - d)'$, i.e. $\boldsymbol{l} = (l_1, \ldots, l_B) = \boldsymbol{\Delta}^{(1)}(1, \boldsymbol{s}, J - d)'$ and vice versa $s_b = l_1 + \ldots + l_b$. The variance for the Gaussian random walk on $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ at indices $i \in \{s_{b-1}, \ldots, s_b - 1\}$ is then given by τ_b^2. By designating a random walk prior with piecewise constant variances, we reduce the numbers of parameters to be sampled. Furthermore, this allows us to take into account local information about the variability of the function to be fitted and thereby increases robustness of the fitted function to outliers compared to using individual variances $\tau_j^2, j = 1, \ldots, J$ for the random walk increments.

Figure 7.1 gives the directed acyclic graph (DAG) for the basic hierarchy of the proposed model specification for Gaussian responses. The corresponding posterior $p(\boldsymbol{y}, \sigma_\varepsilon^2, \boldsymbol{\beta}, \boldsymbol{\tau}^2, \boldsymbol{z}, b_z|\boldsymbol{x})$ is given by

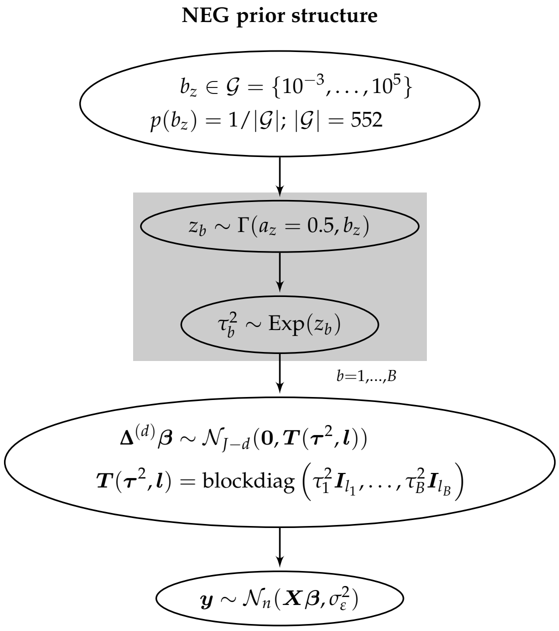

NEG prior structure
$b_z \in \mathcal{G} = \{10^{-3}, \ldots, 10^5\}$
$p(b_z) = 1/|\mathcal{G}|;\ |\mathcal{G}| = 552$
$z_b \sim \Gamma(a_z = 0.5, b_z)$
$\tau_b^2 \sim \text{Exp}(z_b)$
b=1,...,B
$\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta} \sim \mathcal{N}_{J-d}(\mathbf{0}, \boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}))$
$\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}) = \text{blockdiag}\left(\tau_1^2 \boldsymbol{I}_{l_1}, \ldots, \tau_B^2 \boldsymbol{I}_{l_B}\right)$
$\boldsymbol{y} \sim \mathcal{N}_n(\boldsymbol{X}\boldsymbol{\beta}, \sigma_\varepsilon^2)$

Figure 7.1.: Directed acyclic graph of the NEG prior structure.

$$p(\boldsymbol{y}, \sigma_\varepsilon^2, \boldsymbol{\beta}, \boldsymbol{\tau}^2, \boldsymbol{z}, b_z|\boldsymbol{x}) = \\ p_y(\boldsymbol{y}|\boldsymbol{X}\boldsymbol{\beta}, \sigma_\varepsilon^2) p_\beta(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}|\boldsymbol{\tau}^2) p_{\tau^2}(\boldsymbol{\tau}^2|\boldsymbol{z}) p_z(\boldsymbol{z}|a_z, b_z) p_{b_z}(b_z).$$

The rate b_z is sampled with a Metropolis-Hastings-Step. The remaining parameters $\boldsymbol{z}, \boldsymbol{\tau}^2, \boldsymbol{\beta}$ and σ_ε^2 are updated from their full conditionals via Gibbs-

Sampling. We use a weakly informative inverse gamma prior, $\mathrm{IG}(10^{-5}, 10^{-5})$, for the error variance σ_ε^2. Note that it may be useful in applications to standardize the response vector to zero mean and unit variance to meet the requirements of the default prior choices.

For non-Gaussian responses from a univariate exponential family, we model the conditional expectation of the response as $\boldsymbol{\mu} = \mathrm{E}(\boldsymbol{y}|\cdot) = h(\boldsymbol{X}\boldsymbol{\beta})$ for a given response function $h(\cdot)$. This implies that, conditional on the parameters, y is distributed with the density of an exponential family

$$f(y|\cdot) \sim \exp\left(\frac{y\,\theta(\mu) - b(\theta(\mu))}{\phi} + c(y, \phi)\right),$$

where $\theta(\cdot)$ and $b(\cdot)$ are determined by the choice of link function and distribution of y. Compared to the implementation for normal responses, only the last step of the hierarchy, the sampling of the spline coefficients $\boldsymbol{\beta}$, has to be adjusted and the error variance σ_ε^2 is removed from the model. We use a variant of the well-known IWLS proposal scheme (Gamerman, 1997), the penalized IWLS proposal scheme based on an approximation of the current posterior mode described in detail in Brezger and Lang (2006, Sampling scheme 1, section 3.1.1), to update $\boldsymbol{\beta}$ in a single block. This method is a Metropolis-Hastings type update which uses a Gaussian approximation to the full conditional of $\boldsymbol{\beta}$ as its proposal distribution. The approximating Gaussian is obtained by performing a single Fisher scoring step per iteration. Section 7.4 contains a detailed description of the algorithm.

Griffin and Brown (2007) show that the maximum a posteriori (MAP) estimate for the regression coefficients $\boldsymbol{\beta}$ based on the NEG hierarchy fulfills the so-called oracle property in linear models since the derivative of the scale-mixture prior tends to zero for increasing $|\boldsymbol{\beta}|$. Despite this appealing theoretical property of the MAP estimate, we use a full MCMC approach instead of, say, an EM-type algorithm, due to the importance of reliable variability measures for function estimation and because an implementation based on a full MCMC approach will facilitate inclusion into the general structured additive regression context e.g. as part of a Bayesian backfitting algorithm for (G)AMs. It is also reasonable to assume that posterior means based on a prior with the oracle property also benefit from this fact and our simulation results (Section 8) confirm this intuition: Using this hierarchy, we obtain a strong shrinkage of $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ where differences are small, increasing the smoothness of the fitted curve, while simultaneously allowing faithful modeling of jumps or sudden curvature changes.

In the following we describe and compare three approaches with increasing flexibility for the variance function given by $\boldsymbol{\tau}^2$: The first approach uses a piecewise constant variance function with fixed number and positioning of changepoints as described in this section. In the second approach, we sample the locations of the changepoints while leaving their number fixed. In the third approach we use reversible jump MCMC methodology to sample the number of changepoints B as well.

7.1. Blockwise NEG P-spline

In this formulation, the number of blocks B as well as the positions and lengths of the blocks in the variance function are fixed. Simulation results were obtained using blocks of (approximately) equal length, but domain-specific prior knowledge about the likely locations of changes in variability or discontinuities can easily be incorporated into the model by specifying more appropriate locations for the changepoints. The resulting posterior variance function is piecewise constant. We investigate the robustness with respect to the number of blocks in section 8.7.2. The hierarchy for this model is given in Figure 7.1. In the following, this algorithm will be referred to as NEG.

7.2. Flexible blockwise NEG P-spline

In this model, we let B remain fixed and sample the locations of the steps $s_1, \dots, s_{B-1}$ at which the variance of $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ changes. The prior for the vector of interior changepoints $\boldsymbol{s} = (s_1, \dots, s_{B-1})$ is assumed to be the distribution of the order statistic of a discrete uniform distribution on $\{2, \dots, J-d-1\}$. The rest of the hierarchy and the samplers for $b_z, \boldsymbol{z}, \boldsymbol{\tau}^2$ and σ_ε^2 remain unchanged. Figure 7.2 shows the DAG for this prior structure.

7.2.1. Updating the changepoints

We use the following Metropolis-Hastings step to update the vector of changepoints $\boldsymbol{s}$:

- Define the tuning parameter m_s, which is the maximal number of indices that the new proposal can move the selected changepoint. In our implementation, m_s defaults to $\lceil (J-d)/B \rceil$, the length of the random walk divided by the number of blocks and rounded to the next highest integer.

- Draw $b^\star$ uniformly from the set of indices of movable changepoints

  $$\mathcal{B}_s = \{1, \dots, B-1\} \setminus \{b : l_b = 1 \text{ and } l_{b+1} = 1\}.$$

  Indices b where $l_b = 1$ and $l_{b+1} = 1$ are not eligible, because both neighboring intervals only span a single index so that the changepoint

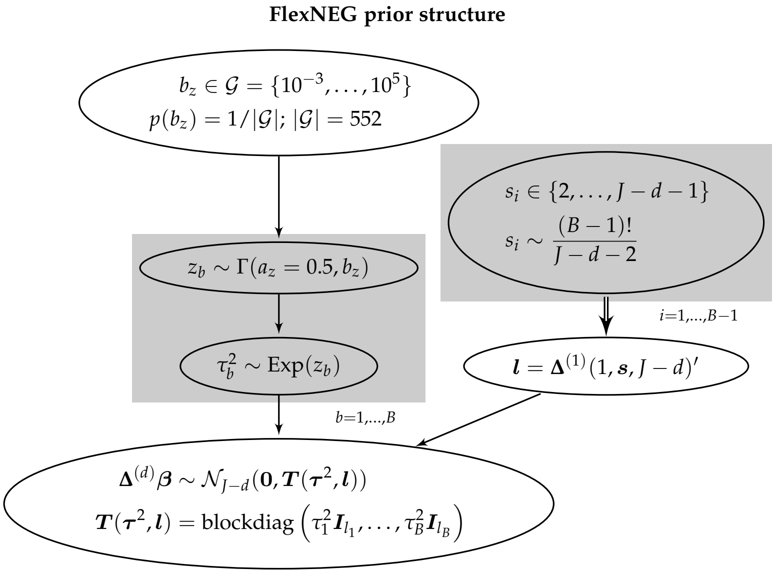

FlexNEG prior structure
$b_z \in \mathcal{G} = \{10^{-3}, \dots, 10^5\}$
$p(b_z) = 1/|\mathcal{G}|; |\mathcal{G}| = 552$
$s_i \in \{2, \dots, J-d-1\}$
$s_i \sim \frac{(B-1)!}{J-d-2}$
$z_b \sim \Gamma(a_z = 0.5, b_z)$
i=1,...,$B-1$
$\tau_b^2 \sim \text{Exp}(z_b)$
$\boldsymbol{l} = \boldsymbol{\Delta}^{(1)}(1, \boldsymbol{s}, J-d)'$
b=1,...,B
$\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta} \sim \mathcal{N}_{J-d}(\boldsymbol{0}, \boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}))$
$\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}) = \text{blockdiag}\left(\tau_1^2 \boldsymbol{I}_{l_1}, \dots, \tau_B^2 \boldsymbol{I}_{l_B}\right)$

Figure 7.2.: Directed acyclic graph of the FlexNEG prior structure. Ellipses are stochastic nodes. Single arrows are stochastic edges, double arrows are deterministic edges.

in the middle cannot move. Let $B_m = |\mathcal{B}_s|$ denote the number of movable changepoints.

- Determine the minimal index $i_- = \max(s_{b^\star - 1} + 1, s_{b^\star} - m_s)$ and maximal index $i_+ = \min(s_{b^\star + 1} - 1, s_{b^\star} + m_s)$ and draw the proposal $s^\star_{b^\star}$ to replace $s_{b^\star}$ uniformly from $\{i_-, \dots, i_+\}$.
- Update $\boldsymbol{s}^\star$, $\boldsymbol{l}^\star$, $i^\star_-$, $i^\star_+$ and $\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)$ accordingly. The prior covariance $\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)$ of $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ is given by blockdiag $\left(\tau_1^2 \boldsymbol{I}_{l_1}, \dots, \tau_B^2 \boldsymbol{I}_{l_B}\right)$.
- Accept the new vector of change points $\boldsymbol{s}^\star$ with probability $\alpha(\boldsymbol{s}^\star)$:

$$\begin{aligned} \log \alpha(\boldsymbol{s}^\star) \;=\; & \log\left((i_+ - i_-)\, B_m\right) - \log\left((i^\star_+ - i^\star_-)\, B^\star_m\right) \qquad (7.1) \\ & +0.5 \left(\frac{\text{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)) - \text{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}))}{\text{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)) \cdot \text{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}))} \right)' (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2 \\ & +0.5(\boldsymbol{l} - \boldsymbol{l}^\star)' \log(\boldsymbol{\tau}^2), \end{aligned}$$

where the expression in the first line is the proposal ratio for $\boldsymbol{s}^\star$, and the second and third line come from the prior ratio for the random walk. A detailed derivation of $\alpha(\boldsymbol{s}^\star)$ is given below.

This model gives substantially more flexibility with regard to the estimated variance function by averaging over the step functions drawn in each iteration.

In effect, we use Bayesian model averaging to arrive at posterior estimates for $f(x)$ and $\text{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}))$. In the following, this algorithm will be referred to as FlexNEG.

7.2.2. Acceptance probability for a proposed vector of changepoints

The following is a more detailed derivation of the acceptance probability $\alpha(\boldsymbol{s}^\star)$ for a proposed vector of changepoints $\boldsymbol{s}^\star$ given in (7.1). Only the proposal ratio for $\boldsymbol{s}^\star$ and the prior ratio for $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ determine $\alpha(\boldsymbol{s}^\star)$ as the prior for the vector of changepoints $\boldsymbol{s}$ has the same value for all possible configurations and the node for $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ is the only daughter node of $\boldsymbol{s}$ in the hierarchy of the model.

The proposal ratio for $\boldsymbol{s}^\star$ is the probability of proposing to move from $\boldsymbol{s}^\star$ to $\boldsymbol{s}$, which is $\frac{1}{B_m^\star}\frac{1}{i_+^\star - i_-^\star}$, the discrete uniform probability of selecting anyone of the movable changepoints multiplied with the discrete uniform probability of selecting any of the indices it is allowed to move to, divided by the probability of proposing to move from $\boldsymbol{s}$ to $\boldsymbol{s}^\star$, which is $\frac{1}{B_m}\frac{1}{i_+ - i_-}$.

We introduce some additional notation for the derivation of the prior ratio

$$\frac{p_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}|\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)\right)}{p_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}|\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})\right)}.$$

Let $T_{ii} = \left(\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})\right)_{ii}$ denote the i^{th} diagonal element of the covariance matrix $\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})$ of $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ and let $T_{ii}^\star = \left(\boldsymbol{T}^\star(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)\right)_{ii}$ denote the i^{th} diagonal element of the proposed covariance matrix implied by the proposed change in the variance function. Note that the diagonal of $\boldsymbol{T}$ will usually remain unchanged in most elements, at most, $\max(s_{b\star} - i_-, i_+ - s_{b\star})$ elements will change.

We repeatedly make use of the fact that $\boldsymbol{T}$ is a diagonal matrix – recall that $\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}) = \text{blockdiag}\left(\tau_1^2 \boldsymbol{I}_{l_1}, \ldots, \tau_B^2 \boldsymbol{I}_{l_B}\right)$. It then follows that

$$\log\left(\frac{p_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}|\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)\right)}{p_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}|\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})\right)}\right) = \log\left(\frac{|\boldsymbol{T}^\star|^{-1/2}\exp\left(-\frac{1}{2}\sum_{i=1}^{J-d}\frac{\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}\right)_i^2}{T_{ii}^\star}\right)}{|\boldsymbol{T}|^{-1/2}\exp\left(-\frac{1}{2}\sum_{i=1}^{J-d}\frac{\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}\right)_i^2}{T_{ii}}\right)}\right)$$

$$= \log \left(\frac{\left(\prod_{i=1}^{J-d} T_{ii}^{\star}\right)^{-1/2}}{\left(\prod_{i=1}^{J-d} T_{ii}^{\star}\right)^{-1/2}} \quad \exp\left(\frac{1}{2}\sum_{i=1}^{J-d} \frac{T_{ii}^{\star} - T_{ii}}{T_{ii}^{\star} T_{ii}} \left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}\right)_i^2 \right)\right)$$

$$= \frac{1}{2}\sum_{i=1}^{J-d} (\log T_{ii} - \log T_{ii}^{\star}) + \frac{1}{2}\sum_{i=1}^{J-d} \frac{T_{ii}^{\star} - T_{ii}}{T_{ii}^{\star} T_{ii}} \left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}\right)_i^2$$

$$= 0.5(\boldsymbol{l} - \boldsymbol{l}^{\star})' \log(\boldsymbol{\tau}^2) + 0.5 \left(\frac{\operatorname{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^{\star})) - \operatorname{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}))}{\operatorname{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^{\star})) \cdot \operatorname{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}))} \right)' (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2.$$

7.3. Flexible blockwise NEG P-spline with variable number of blocks

As the most flexible alternative, we also implemented a reversible jump-type algorithm (Green, 1995) to determine the number of changepoints B automatically in a data-driven way. Following Green (1995), we assigned a Poisson distribution truncated on $\{1, \dots, s_{\text{max}}\}$ with rate s_{mean} as prior $p_B(B)$ to the number of blocks B. The rest of the hierarchy remains unchanged. Figure 7.3 shows the DAG for this prior structure. The reversible jump algorithm has three move types: birth (adding a changepoint), death (removing a changepoint), and position change. The latter is identical to the update procedure for $\boldsymbol{s}$ described in the previous section. Let $p_b(B)$ and $p_d(B)$ denote the probabil-

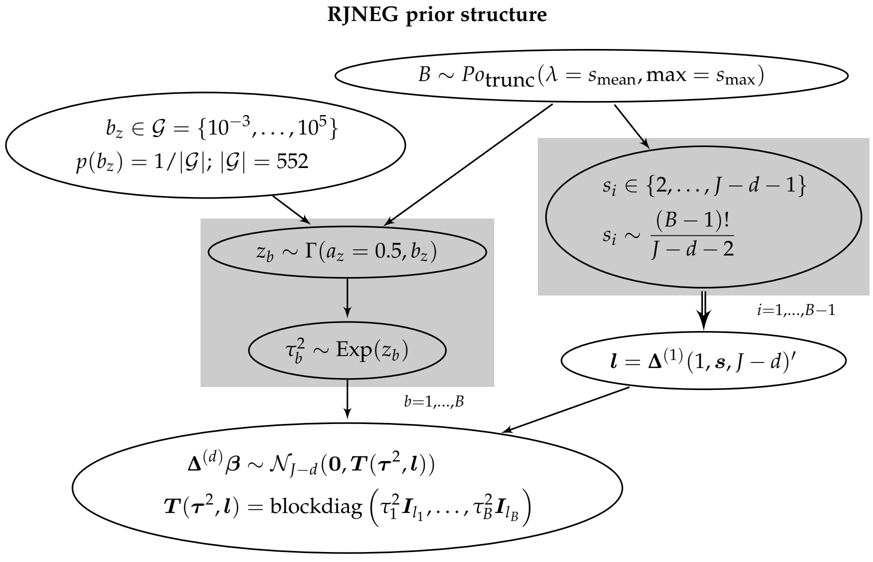

RJNEG prior structure
$B \sim Po_{\text{trunc}}(\lambda = s_{\text{mean}}, \max = s_{\text{max}})$
$b_z \in \mathcal{G} = \{10^{-3}, \dots, 10^5\}$
$p(b_z) = 1/|\mathcal{G}|$; $|\mathcal{G}| = 552$
$s_i \in \{2, \dots, J-d-1\}$
$s_i \sim \frac{(B-1)!}{J-d-2}$
$z_b \sim \Gamma(a_z = 0.5, b_z)$
$\tau_b^2 \sim \text{Exp}(z_b)$
$i=1,\dots,B-1$
$\boldsymbol{l} = \boldsymbol{\Delta}^{(1)}(1, \boldsymbol{s}, J-d)'$
$b=1,\dots,B$
$\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta} \sim \mathcal{N}_{J-d}(\boldsymbol{0}, \boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}))$
$\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}) = \text{blockdiag}\left(\tau_1^2 \boldsymbol{I}_{l_1}, \dots, \tau_B^2 \boldsymbol{I}_{l_B}\right)$

Figure 7.3.: Directed acyclic graph of the RJNEG prior structure. Ellipses are stochastic nodes. Single arrows are stochastic edges, double arrows are deterministic edges.

ity for a birth and death step, respectively, given the number of blocks B. To satisfy detailed balance, we set $p_b(B) = c \min(1, p_B(B+1)/p_B(B))$; $p_d(B) = c \min(1, p_B(B-1)/p_B(B))$, where c is chosen so that $p_b(B) + p_d(B) < 0.8 \; \forall \; B$ (Green, 1995). The birth and death moves to increase or decrease B are as follows:

7.3.1. Birth move

A birth move increases the number of blocks by one: $B \to (B+1)$. It is performed in the following steps:

- Draw the proposed new changepoint $s^\star$ uniformly from $\{2, \ldots, J-d-1\} \setminus \{\boldsymbol{s}\}$.
- Determine the affected block $b^\star : s_{b^\star - 1} < s^\star < s_{b^\star}$ and the (expanded) proposal vectors $\boldsymbol{s}^\star$ and $\boldsymbol{l}^\star$.
- Draw proposals $\left(z^\star_{b^\star}, z^\star_{b^\star+1}\right) \overset{\text{i.i.d.}}{\sim} \Gamma(a_z + 1, b_z + \tau^2_{b^\star})$ from the full conditional based on $\boldsymbol{\tau}^2$ from the *previous* iteration.
- Draw proposals $\tau^{2\star}_{b^\star}, \tau^{2\star}_{b^\star+1}$ from their full conditionals (see Section 7.5.1) based on the *updated* vector $\boldsymbol{z}^\star$.
- Accept $B^\star = B + 1$, $\boldsymbol{s}^\star$, $\boldsymbol{z}^\star$ and $\boldsymbol{\tau}^{2\star}$ with probability $\alpha_b = \mathcal{A}_b \mathcal{P}_b$, where $\mathcal{A}_b$ is the prior ratio and $\mathcal{P}_b$ is the proposal ratio for the attempted birth move.

The acceptance probability has this simple form because the likelihood ratio for dimension changing moves is 1. In our context, the changed parameters do not occur in the likelihood but only in a higher stage of the hierarchy. The Jacobian is 1 as well since the mapping function between the parameter spaces is the identity. The prior ratio is given by

$$\begin{aligned}\mathcal{A}_b =& \frac{p_B(B^\star | s_{\text{mean}}, s_{\max})}{p_B(B | s_{\text{mean}}, s_{\max})} \cdot \frac{p_s(\boldsymbol{s}^\star | B^\star)}{p_s(\boldsymbol{s} | B)} \cdot \frac{p_z\left((z^\star_{b^\star}, z^\star_{b^\star+1}) | b_z\right)}{p_z(z_{b^\star} | b_z)} \cdot \\ &\cdot \frac{p_\tau\left((\tau^{2\star}_{b^\star}, \tau^{2\star}_{b^\star+1}) | \boldsymbol{z}^\star\right)}{p_\tau(\tau^2_{b^\star} | \boldsymbol{z})} \cdot \frac{p_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta} | \boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)\right)}{p_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta} | \boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})\right)},\end{aligned}$$

and the proposal ratio for the birth step is

$$\mathcal{P}_b = \frac{p_d(B^\star)}{p_b(B)} \cdot \frac{|\{2, \ldots, J-d-1\} \setminus \{\boldsymbol{s}\}|}{B} \cdot \frac{p\left(z_{b^\star} | a_z, b_z, \tau^2_{b^\star}\right)}{p\left((z^\star_{b^\star}, z^\star_{b^\star+1}) | a_z, b_z, \tau^2_{b^\star}\right)}.$$

$$\cdot \frac{p\left(\tau^2_{b^\star} | \boldsymbol{s}, \boldsymbol{l}, \tilde{z}_{b^\star} = 0.5(z^\star_{b^\star} + z^\star_{b^\star+1}), \boldsymbol{\beta}\right)}{p\left((\tau^{2\star}_{b^\star}, \tau^{2\star}_{b^\star+1}) | \boldsymbol{s}^\star, \boldsymbol{l}^\star, \boldsymbol{z}^\star, \boldsymbol{\beta}\right)}.$$

Section 7.3.3 contains more detailed expressions.

7.3.2. Death move

A death move decreases the number of blocks by one: $B \to B - 1$. It is performed in the following steps:

- Draw index $b^\star$ of the changepoint that is to be deleted uniformly from $\{1, \ldots, B-1\}$ and determine the reduced proposal vectors $\boldsymbol{s}^\star$, $\boldsymbol{l}^\star$.
- Draw the new proposal $\tau^{2\star}_{b^\star}$ to replace $(\tau^2_{b^\star}, \tau^2_{b^\star+1})$ from $p(\tau^{2\star}_{b^\star} | \boldsymbol{s}^\star, \tilde{z}_{b^\star} = 0.5(z_{b^\star} + z_{b^\star+1}))$
- Draw the new proposal $z^\star_{b^\star}$ to replace $(z_{b^\star}, z_{b^\star+1})$ from $p(z^\star_{b^\star} | a_z, b_z, \boldsymbol{\tau}^{2\star})$
- Accept $B^\star = B - 1$, $\boldsymbol{s}^\star$, $\boldsymbol{z}^\star$ and $\boldsymbol{\tau}^{2\star}$ with probability $\alpha_d = \mathcal{A}_d \mathcal{P}_d$: where $\mathcal{A}_d$, the prior ratio, and $\mathcal{P}_d$, the proposal ratio, are simply $\mathcal{A}_b^{-1}$ and $\mathcal{P}_b^{-1}$ with indices appropriately changed.

Section 7.3.3 contains more detailed expressions.

In the following, the algorithm with a variable number of changepoints will be referred to as RJNEG.

The required dimension matching (Green, 1995) is fulfilled since the dimension changes (in the notation for the birth step) proceed from parameter vector $(\boldsymbol{z}, \boldsymbol{\tau}^2, z^\star_{b^\star}, z^\star_{b^\star+1}, \tau^{2\star}_{b^\star}, \tau^{2\star}_{b^\star+1})$ with dimension $2B + 4$ to parameter vector $(\boldsymbol{z}^\star, \boldsymbol{\tau}^{2\star}, z_{b^\star}, \tau^2_{b^\star})$ with dimension $2(B+1) + 2$ and vice versa for the death step. Our sampler alternates between the dimension-changing transition kernel implied by the update procedure above and the fixed-dimension kernel described in section 7.2. While this usually increases the necessary burn-in period, it also stabilizes the estimation of the variance function in more complex settings.

In our simulations, acceptance probabilities for the dimension changing moves were in the range of 0.3 to 0.6 and usually around 0.4.

7.3.3. Acceptance probabilities for birth and death moves

The prior ratio for the birth step is

$$
\begin{aligned}
\mathcal{A}_b =& \frac{p_B(B^\star|s_{\text{mean}}, s_{\max})}{p_B(B|s_{\text{mean}}, s_{\max})} \frac{p_s(\boldsymbol{s}^\star|B^\star)}{p_s(\boldsymbol{s}|B)} \frac{p_z\left((z^\star_{b^\star}, z^\star_{b^\star+1})|b_z\right)}{p_z(z_{b^\star}|b_z)} \\
& \frac{p_\tau\left((\tau^{2\star}_{b^\star}, \tau^{2\star}_{b^\star+1})|\boldsymbol{z}^\star\right)}{p_\tau(\tau^2_{b^\star}|\boldsymbol{z})} \frac{p_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}|\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)\right)}{p_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}|\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})\right)} \\
=& \frac{s_{\text{mean}}}{B^\star} \frac{B}{(J-d-2)} \\
& \frac{b_z^{a_z}}{\Gamma(a_z)} \left(\frac{z^\star_{b^\star} z^\star_{b^\star+1}}{z_{b^\star}}\right)^{a_z} \frac{\exp\left(-(b_z + \tau^{2\star}_{b^\star}) z^\star_{b^\star} - (b_z + \tau^{2\star}_{b^\star+1}) z^\star_{b^\star+1}\right)}{\exp\left(-(b_z + \tau^2_{b^\star}) z_{b^\star}\right)} \\
& \frac{\sqrt{\tau^{2\star}_{b^\star}}^{-l^\star_{b^\star}} \sqrt{\tau^{2\star}_{b^\star+1}}^{-l^\star_{b^\star+1}} \exp\left(-\frac{1}{2}\sum_{b=1}^{B^\star}\sum_{k=s^\star_b}^{s^\star_{b+1}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k \tau_b^{-2\star}\right)}{\sqrt{\tau^2_{b^\star}}^{-l_{b^\star}} \exp\left(-\frac{1}{2}\sum_{b=1}^{B}\sum_{k=s_b}^{s_{b+1}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k \tau_b^{-2}\right)}
\end{aligned}
$$

and the proposal ratio for the birth step is

$$
\begin{aligned}
\mathcal{P}_b =& \frac{p_d(B^\star)}{p_b(B)} \frac{|\{2, \ldots, J-d-1\} \setminus \{\boldsymbol{s}\}|}{B} \frac{p\left(z_{b^\star}|a_z, b_z, \tau^2_{b^\star}\right)}{p\left((z^\star_{b^\star}, z^\star_{b^\star+1})|a_z, b_z, \tau^2_{b^\star}\right)} \\
& \frac{p\left(\tau^2_{b^\star}|\boldsymbol{s}, \boldsymbol{l}, \tilde{z}_{b^\star} = 0.5(z^\star_{b^\star} + z^\star_{b^\star+1}), \boldsymbol{\beta}\right)}{p\left((\tau^{2\star}_{b^\star}, \tau^{2\star}_{b^\star+1})|\boldsymbol{s}^\star, \boldsymbol{l}^\star, \boldsymbol{z}^\star, \boldsymbol{\beta}\right)} \\
=& \frac{p_d(B^\star)}{p_b(B)} \frac{J-d-B-1}{B} \frac{\Gamma(a_z+1)}{(b_z + \tau^2_{b^\star})^{a_z+1}} \frac{z_{b^\star}^{a_z}}{z^{\star a_z}_{b^\star} z^{\star a_z}_{b^\star+1}} \frac{\exp\left(-(b_z + \tau^2_{b^\star}) z_{b^\star}\right)}{\exp\left(-(b_z + \tau^2_{b^\star})(z^\star_{b^\star} + z^\star_{b^\star+1})\right)} \\
& \frac{(z^\star_{b^\star} + z^\star_{b^\star+1})^{1/2 - l_{b^\star}/4} \left(\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\right)^{1/2 - l^\star_{b^\star}/4} \left(\sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\right)^{1/2 - l^\star_{b^\star+1}/4}}{2\left(2 z^\star_{b^\star}\right)^{1/2 - l^\star_{b^\star}/4} \left(2 z^\star_{b^\star+1}\right)^{1/2 - l^\star_{b^\star+1}/4} \left(\sum_{k=s_{b^\star}}^{s_{b^\star+1}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\right)^{1/2 - l_{b^\star}/4}} \\
& \frac{K_{1-l^\star_{b^\star}/2}\left(\sqrt{2\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k z^\star_{b^\star}}\right) K_{1-l^\star_{b^\star+1}/2}\left(\sqrt{2\sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k z^\star_{b^\star+1}}\right)}{K_{1-l_{b^\star}/2}\left(\sqrt{\sum_{k=s_{b^\star}}^{s_{b^\star+1}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k (z^\star_{b^\star} + z^\star_{b^\star+1})}\right)} \\
& \frac{\sqrt{\tau^2_{b^\star}}^{-l_{b^\star}}}{\sqrt{\tau^{2\star}_{b^\star}}^{-l_{b^\star}} \sqrt{\tau^{2\star}_{b^\star+1}}^{-l_{b^\star+1}}} \\
& \frac{\exp\left(-\frac{1}{2}\left(\sum_{k=s_{b^\star}}^{s_{b^\star+1}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k \tau^{-2}_{b^\star} + (z^\star_{b^\star} + z^\star_{b^\star+1}) \tau^2_{b^\star}\right)\right)}{\exp\left(-\frac{1}{2}\left(\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k \tau^{-2\star}_{b^\star} + \sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k \tau^{-2\star}_{b^\star+1} + 2(z^\star_{b^\star} \tau^{2\star}_{b^\star} + z^\star_{b^\star+1} \tau^{2\star}_{b^\star+1})\right)\right)}.
\end{aligned}
$$

This yields acceptance probability

$$\begin{aligned}
\alpha_b = & \frac{p_d(B^\star)}{p_b(B)} \frac{s_{\text{mean}}}{B^\star} \frac{(J-d-B-1)}{(J-d-2)} \frac{a_z}{(b_z+\tau^2_{b^\star})^{a_z+1}} \frac{\exp\left(-\frac{1}{2}\sum_{b=1}^{B^\star}\sum_{k=s_b^\star}^{s_{b+1}^\star-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2\tau_b^{-2\star}\right)}{\exp\left(-\frac{1}{2}\sum_{b=1}^{B}\sum_{k=s_b}^{s_{b+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2\tau_b^{-2}\right)} \\
& \frac{\left(z^\star_{b^\star}+z^\star_{b^\star+1}\right)^{1/2-l_{b^\star}/4}\left(\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2\right)^{1/2-l^\star_{b^\star}/4}\left(\sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2\right)^{1/2-l^\star_{b^\star+1}/4}}{2\left(2z^\star_{b^\star}\right)^{1/2-l^\star_{b^\star}/4}\left(2z^\star_{b^\star+1}\right)^{1/2-l^\star_{b^\star+1}/4}\left(\sum_{k=s_{b^\star}}^{s_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2\right)^{1/2-l_{b^\star}/4}} \\
& \frac{K_{1-l^\star_{b^\star}/2}\left(\sqrt{2\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2 z^\star_{b^\star}}\right) K_{1-l^\star_{b^\star+1}/2}\left(\sqrt{2\sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2 z^\star_{b^\star+1}}\right)}{K_{1-l_{b^\star}/2}\left(\sqrt{\sum_{k=s_{b^\star}}^{s_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2(z^\star_{b^\star}+z^\star_{b^\star+1})}\right)} \\
& \frac{\exp\left(-\frac{1}{2}\left(\sum_{k=s_{b^\star}}^{s_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2\tau_{b^\star}^{-2}+(z^\star_{b^\star}+z^\star_{b^\star+1})\tau^2_{b^\star}\right)\right)}{\exp\left(-\frac{1}{2}\left(\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2\tau_{b^\star}^{-2\star}+\sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2\tau_{b^\star+1}^{-2\star}+2(z^\star_{b^\star}\tau^{2\star}_{b^\star}+z^\star_{b^\star+1}\tau^{2\star}_{b^\star+1})\right)\right)}.
\end{aligned}$$

7.4. Extension to non-Gaussian responses

We also adapted this procedure for binary Binomial responses (logit link) and Poisson responses (log link), replacing the Gibbs-type update of $\boldsymbol{\beta}$ with a penalized IWLS update (Gamerman, 1997; Brezger and Lang, 2006).

The principal idea of this method is to use Fisher scoring or IWLS for the estimation of regression coefficients in generalized linear models within a Metropolis-Hastings step. In essence, the full conditional of $\boldsymbol{\beta}$ is approximated by a multivariate Gaussian distribution, whose parameters are obtained by a single Fisher scoring step for each sweep of the sampling algorithm. The following is adapted from Brezger and Lang (2006).

Standard P-IWLS then proposes a multivariate Gaussian candidate vector $\boldsymbol{\beta}^\star \sim \mathcal{N}_J(\boldsymbol{m}, \boldsymbol{P}^{-1})$ based on the current value $\boldsymbol{\beta}$ with precision matrix $\boldsymbol{P}$ and mean $\boldsymbol{m}$, where

$$\begin{aligned}
\boldsymbol{P} &= \boldsymbol{X'W}(\boldsymbol{\beta})\boldsymbol{X} + \boldsymbol{T}^{-1}(\boldsymbol{\tau}^2, \boldsymbol{l}), \\
\boldsymbol{m} &= \boldsymbol{P}^{-1}\boldsymbol{X'W}(\boldsymbol{\beta})\left(\tilde{\boldsymbol{y}}(\boldsymbol{\beta}) - \boldsymbol{X\beta}\right).
\end{aligned}$$

The matrix of IWLS weights $\boldsymbol{W}(\boldsymbol{\beta}) = \operatorname{diag}(w_1(\boldsymbol{\beta}), \ldots, w_n(\boldsymbol{\beta}))$ and the vector of working observations $\tilde{\boldsymbol{y}}(\boldsymbol{\beta})$ for canonical link functions are given as

$$\begin{aligned}
w_i(\boldsymbol{\beta}) &= b''(\theta_i), \\
\tilde{y}_i(\boldsymbol{\beta}) &= \boldsymbol{x_i'\beta} + g'(\mu_i)(y_i - \mu_i).
\end{aligned}$$

Following Brezger and Lang (2006) (Sampling scheme 1, section 3.1.1) we use the posterior mode approximation $\boldsymbol{m}$ used in the *previous* iteration of the sampler (regardless of acceptance of the previous proposal) instead of $\boldsymbol{\beta}$ to compute the weights and working observations. This is advantageous, because the proposal densities $q(\boldsymbol{\beta} \to \boldsymbol{\beta}^\star)$ and $q(\boldsymbol{\beta}^\star \to \boldsymbol{\beta})$ then become independent of the current state of the chain ($q(\boldsymbol{\beta} \to \boldsymbol{\beta}^\star) = q(\boldsymbol{\beta}^\star)$; $q(\boldsymbol{\beta}^\star \to \boldsymbol{\beta})) = q(\boldsymbol{\beta})$), so that we don't need to re-compute $\boldsymbol{P}^\star$ and $\boldsymbol{m}^\star$ for $q(\boldsymbol{\beta}^\star \to \boldsymbol{\beta})$ to determine the acceptance probability of $\boldsymbol{\beta}^\star$. This also increases the acceptance rates for the high-dimensional proposals we use.

7.5. MCMC sampler

7.5.1. Posterior and full conditionals

For the hierarchy given for the reversible jump model, the full posterior with given hyperparameters $s_{\text{mean}}, s_{\text{max}}, a_z, b_z, a_\sigma$ and b_σ can be written as

$$
\begin{aligned}
&p(B, \boldsymbol{s}, \boldsymbol{l}, \boldsymbol{z}, \boldsymbol{\tau}^2, \boldsymbol{\beta}, \sigma^2_\varepsilon, y) = \\
&\quad \left(1 - \sum_{i=1}^{s_{\text{max}}} \frac{s^i_{\text{mean}}}{i!} e^{-s_{\text{mean}}}\right)^{-1} \frac{s^B_{\text{mean}}}{B!} e^{-s_{\text{mean}}} \cdot \frac{(B-1)!}{(J-d-2)^{B-1}} \cdot \\
&\quad \frac{b_z^{B a_z}}{\Gamma(a_z)^B} \prod_{b=1}^{B} z_b^{a_z - 1} \exp\left(-b_z z_b\right) \cdot \prod_{b=1}^{B} z_b \exp\left(-z_b \tau_b^2\right) \cdot \\
&\quad \frac{\prod_{b=1}^{B} \sqrt{\tau_b^2}^{-l_b}}{(2\pi)^{(J-d)/2}} \exp\left(-\frac{1}{2} \boldsymbol{\beta}' \boldsymbol{\Delta}^{(d)\prime} \boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})^{-1} \boldsymbol{\Delta}^{(d)} \boldsymbol{\beta}\right) \cdot \\
&\quad \frac{b_\sigma^{a_\sigma}}{\Gamma(a_\sigma)} \sigma_\varepsilon^{2(-a_\sigma - 1)} \exp\left(\frac{-b_\sigma}{\sigma^2_\varepsilon}\right) \\
&\quad \frac{1}{(2\pi\sigma^2_\varepsilon)^{n/2}} \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2}{2\sigma^2_\varepsilon}\right).
\end{aligned}
$$

Accordingly, the full conditionals are:

$$
\begin{aligned}
&p(z_b | a_z, b_z, \tau_b^2) \propto z_b^{a_z} \exp\left(-(b_z + \tau_b^2) z_b\right) \\
&\Rightarrow z_b | \cdot \sim \Gamma(a_z + 1, b_z + \tau_b^2) \\
&p(\tau_b | \boldsymbol{s}, \boldsymbol{l}, \boldsymbol{\beta}, z_b) \propto \sqrt{\tau_b^2}^{-l_b} \exp\left(-\frac{1}{2\tau_b^2} \sum_{k=s_b}^{s_{b+1}-1} (\boldsymbol{\Delta}^{(d)} \boldsymbol{\beta})_k^2 - z_b \tau_b^2\right)
\end{aligned}
$$

$$= (\tau_b^2)^{-l_b/2} \exp\left(-\frac{1}{2}\left(\sum_{k=s_b}^{s_{b+1}-1} (\mathbf{\Delta}^{(d)}\boldsymbol{\beta})_k^2 (\tau_b^2)^{-1} + 2z_b\tau_b^2\right)\right)$$

$$\Rightarrow \tau_b^2|\cdot \sim GIG\left(\chi = \sum_{k=s_b}^{s_{b+1}-1} (\mathbf{\Delta}^{(d)}\boldsymbol{\beta})_k^2; \psi = 2z_b; \lambda = 1 - \frac{l_b}{2}\right)$$

$GIG(\chi, \psi, \lambda)$ denotes the generalized inverse Gaussian distribution with density

$$f(x) = \frac{(\psi/\chi)^{\lambda/2}}{2K_\lambda(\sqrt{\psi\chi})} x^{\lambda-1} \exp\left(-\frac{1}{2}\left(\chi x^{-1} + \psi x\right)\right)$$

for $x > 0$, where $K_\lambda(\cdot)$ is the modified Bessel function of the third kind of (fractional) order λ (Jørgensen, 1982). We use our own C-code implementing the algorithm given by Dagpunar (1989) to sample from this distribution.

$$p(\boldsymbol{\beta}|\boldsymbol{\tau}^2, \boldsymbol{l}, \sigma_\varepsilon^2) \propto \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2}{2\sigma_\varepsilon^2} - \frac{\boldsymbol{\beta}'\mathbf{\Delta}^{(d)\prime}\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})^{-1}\mathbf{\Delta}^{(d)}\boldsymbol{\beta}}{2}\right)$$

$$\Rightarrow \boldsymbol{\beta}|\cdot \sim \mathcal{N}_J\left(\boldsymbol{\mu} = \sigma_\varepsilon^{-2}\boldsymbol{V}\boldsymbol{X}'\boldsymbol{y}; \boldsymbol{\Sigma} = \boldsymbol{V}\right);$$

$$\boldsymbol{V} = \left(\sigma_\varepsilon^{-2}\boldsymbol{X}'\boldsymbol{X} + \mathbf{\Delta}^{(d)\prime}\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})^{-1}\mathbf{\Delta}^{(d)}\right)^{-1}$$

$$p(\sigma_\varepsilon^2|a_\sigma, b_\sigma, \boldsymbol{\beta}) \propto \sigma_\varepsilon^{2(-a_\sigma - n/2 - 1)} \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + 2b_\sigma}{2\sigma_\varepsilon^2}\right)$$

$$\Rightarrow \sigma_\varepsilon^2|\cdot \sim IG(a_\sigma + n/2, b_\sigma + \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2/2)$$

7.5.2. Performance

MCMC sampling for high-dimensional hierarchical models such as the ones we consider can run into a number of difficulties such as failure to visit all the relevant modes in cases of multimodality, non-convergence, slow mixing and strong sensitivity to starting values. To address these concerns, we investigated the behavior of the proposed samplers for multiple runs initialized with highly overdispersed starting values for b_z and $\boldsymbol{\tau}^2$ generated from their respective diffuse priors. Sensitivity to hyperparameters is discussed in section 8.7.2.

We evaluated convergence of the runs by convergence in $\boldsymbol{\beta}$ (and σ_ε^2, for Gaussian responses), as the parameters $(\boldsymbol{z}, \boldsymbol{\tau}^2)$ change in meaning due to the changing shape of $\boldsymbol{s}$ for FlexNEG and RJ.NEG. We found that all the samplers for Gaussian responses converge quickly (<1000 iterations) even if initialized in highly improbable regions of the posterior and mix very well. Some qualifications apply for the basic NEG model: it is more sensitive to starting values due to its less flexible parametrization and occasionally gets stuck in local modes with too much or too little regularization of the spline coefficients in some regions if unfortunate starting values are chosen, especially if the initial value of b_z is extreme. This is not the case for FlexNEG and RJ.NEG, which are able to move away from inferior basins of attraction quickly due to the more flexible shape of the variance function.

Generally speaking, the quick convergence and excellent mixing is due to the availability of blockwise Gibbs sampling steps in the relevant levels of the hierarchy, which obviate the manual tuning of proposal densities entirely. We achieved stable results for the NEG for starting values of $\tau^2 = 100$ and $0.1 < b_z < 10$, which is the configuration we used in the simulation study.

The performance of the P-IWLS sampler for $\boldsymbol{\beta}$ for non-Gaussian responses is highly dependent on the starting values: If unsuitable starting values are provided, the sampler will get stuck in the initial configuration and fail to update, because the local approximation of the posterior used for the proposal density is unsuitable. In our implementation, suitable starting values for $\boldsymbol{\beta}$ to initialize the chain are found by performing a number of Fisher scoring steps for fixed values of $\boldsymbol{\tau}^2$ starting from the unregularized estimate of $\boldsymbol{\beta}$. Chains initialized in this way converge quickly regardless of the starting values for b_z and $\boldsymbol{\tau}^2$, with satisfactory acceptance rates and good mixing due to the automatic adaptation of the proposal density to the mode and curvature of the full conditional. In our simulation study, acceptance rates for the IWLS proposals are between 26% and 88% and usually around 60% for Poisson responses. Acceptance rates for binary Binomial responses are between 13% and 42%, and usually around 25%. Acceptance rates tend to decrease with increasing J since we update all elements of $\boldsymbol{\beta}$ simultaneously, which is necessary to achieve good mixing. Note that our implementation is therefore not well suited for very heavily parameterized models using more than 100 basis functions. This will rarely be an issue in practical applications, however.

We did encounter some numerical problems in the fitting of very challenging functional forms: On the one hand, the sampling of variates from the generalized inverse Gaussian distribution (the full conditional density of $\boldsymbol{\tau}^2$) can occasionally fail for extreme combinations of parameter values. In this case, we simply keep the previous iterates of the respective elements of $\boldsymbol{\tau}^2$. If, in the case of RJ.NEG, this is not possible because the dimension of the $\boldsymbol{\tau}^2$-proposal to be drawn is different from the dimension of the current $\boldsymbol{\tau}^2$, we calculate the expected values of the full conditional distributions for the problematic elements of $\boldsymbol{\tau}^2$ and use those as the updated values. The software gives out a warning if sampling from the generalized inverse Gaussian fails. In our experience, this ad-hoc fix works well in practice and the resulting samples are indistinguishable from regularly obtained samples, because, if at all, only ever a small fraction of elements in $\boldsymbol{\tau}^2$ fails to update in the regular

fashion so that the convergence of the chain is not affected.
A second type of problem – which we were unable to fix – occurs (rarely) in the sampling of $\boldsymbol{\beta}$:
As the expressions above show, generating a new $\boldsymbol{\beta}$-proposal requires the inversion of the $J \times J$ matrix $\left(\sigma_\varepsilon^{-2}\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{\Delta}^{(d)\prime}\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})^{-1}\boldsymbol{\Delta}^{(d)}\right)$. Occasionally, despite its construction ensuring positive definiteness, this matrix will be numerically not positive definite (or not even semi-definite). In this case the Cholesky-root based matrix inversion we employ fails and we switch to an inversion based on the singular value decomposition (SVD). When the underlying BLAS routine for SVD fails as well, the program crashes. This caused the 12 aborted fits for RJ.NEG for $m_3(x)$ in Section 8.3. Using the numerically more stable QR- or LU-decompositions for matrix inversion is unfortunately not effective in this case since we also require the matrix root $\boldsymbol{V}^{1/2}$ to generate the multivariate normal proposal vector $\boldsymbol{\beta}_{prop} = \boldsymbol{\mu} + \boldsymbol{V}^{1/2}\boldsymbol{\eta}$, where $\eta \sim \mathcal{N}(0, 1)$. This matrix root can be obtained, however, from the Cholesky or the SVD of $\boldsymbol{V}^{-1}$.

7.5.3. Alternative proposals for the birth and death moves

We also experimented with a more complex proposal scheme for the birth and death moves. Specifically, for the birth step we select the interval
$b^\star \in \{1, \ldots, B-1\} \setminus \{b : l_b = 1\}$ with probability

$$p(b) \propto l_b^2 \frac{\text{Var}\left((\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_{s_{b-1},\ldots,s_b-1}\right)}{\sum_{k=s_{b-1}}^{s_b-1} |(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k|},$$

placing a higher proposal density on selecting long intervals with a large variation coefficient of the increments of the random walk. This increases the chance of splitting intervals in which both the proportion of small changes in $\boldsymbol{\beta}$ and the variability in the entries of $(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})$ are large. Intervals with those properties are not homogeneous and can potentially benefit from at least one additional changepoint separating the small changes, which may warrant stronger regularization, from the larger ones responsible for the larger variation which potentially reflect jumps or curvature changes in the function to be fitted. The location of the new changepoint $s_{b^\star}^\star$ is then drawn uniformly from
$\{s_{b^\star} + 1, \ldots, s_{b^\star+1} - 1\}$.
In the death step, we select the changepoint $s_{b^\star}$; $b^\star \in \{1, \ldots, B-1\}$ to be

removed with probability

$$p(b) \propto \frac{1}{l_b + l_{b+1}} \left| \frac{\sum_{k=s_{b-1}}^{s_b - 1} (\mathbf{\Delta}^{(d)}\boldsymbol{\beta})_k}{l_b} - \frac{\sum_{k=s_b}^{s_{b+1}-1} (\mathbf{\Delta}^{(d)}\boldsymbol{\beta})_k}{l_{b+1}} \right|.$$

This increases the chance of removing a changepoint s_b with short adjacent intervals and small difference between the neighboring local means of $\mathbf{\Delta}^{(d)}\boldsymbol{\beta}$. The fitted functions based on these proposals and a uniform prior for the number of knots B were practically identical to fitted functions for the simpler algorithm with a truncated Poisson prior for B. We did not observe any improvement in the sense of a more parsimonious representation of the variance function of the random walk and acceptance probabilities for the dimension changing moves were unreasonably low ($0.1 - 0.2$) in most cases.

8. Simulation results

This chapter presents the results of a simulation study we conducted to evaluate the performance of our methods. We compared the performance of our approach to the performance of the spatially adaptive Bayesian P-Splines suggested by Baladandayuthapani et al. (2005) and the frequentist equivalent of their model described in Krivobokova et al. (2008) and implemented in the R (R Development Core Team, 2010) package AdaptFit (Krivobokova, 2007). In the following, these approaches will be referred to as BMC and AdaptFit, respectively. For both algorithms we used the published hyperparameter settings, number of knots etc. Both BMC and AdaptFit are based on a representation of the logarithm of the variance function $\log(\tau^2(x))$ as a second P-spline. We additionally compare our approaches to Bayesian adaptive regression splines (Dimatteo et al., 2001) (R implementation by Wallstrom (2006)), which is a fully Bayesian method that employs a reversible jump type algorithm to sample from an approximate marginal posterior distribution of the possible sets of knots, followed by sampling from the full conditional of the spline coefficients. We use these methods for benchmarking since their performances are reportedly superior – or at least equivalent – to those of the competing wavelet approach of Donoho and Johnstone (1994), to the knot-selection based approach by Denison et al. (1998) and the approach based on a heteroscedastic heavy-tailed random walk priors for $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ by Lang and Brezger (2004). We also compared the performance of our approach to the performance of the latter. Average MSEs were consistently larger by an order of magnitude for the latter and we omit a detailed analysis for these results in the following. In order to quantify the relative merits of the adaptive approaches, we also calculated the average MSE (AMSE) for non-adaptive fits with `spm` (R-package `SemiPar`, Wand, Coull, French, Ganguli, Kammann, Staudenmayer, and Zanobetti (2005)), which estimates the smoothing parameter of a (generalized) additive model via REML. This is the empirical Bayes equivalent of a fully Bayesian P-spline. Computation times for the fully Bayesian methods were obtained on a Pentium P4 (2.8 GHz, 1 GB RAM) for hyperprior settings resulting in models with similar model complexity.

We consider four widely used benchmarking functions that, together, represent a cross section of challenging functional forms encountered in real-world data. We generated 100 datasets for every function and obtained the fits of the considered methods. Pointwise coverage values (calculated for a nominal level of 90%) should therefore be treated with caution, since the number of simulation replications is not really large enough for reliable estimation. The specifications of hyperparameters and in particular the number of basis functions employed for each of the functions were chosen in concordance with specifications considered in the cited previous simulation studies (e.g. Baladandayuthapani et al., 2005; Krivobokova et al., 2008). Section 8.7.2 contains a discussion on the use of DIC to determine suitable values for hyperparameters B and $s_{\text{mean}}, s_{\text{max}}$ which were used in the simulation study.

Graphical panels showing boxplots of $\log_{10}(\sqrt{MSE})$ as well as average pointwise bias and pointwise coverage (nominal level: 0.9) for the Gaussian responses can be found in Figs. 8.1 to 8.3. The discrepancies between our results for BMC in section 8.1 and published results are due to a minor glitch in the simulation code used in that work (Baladandayuthapani, 2008).

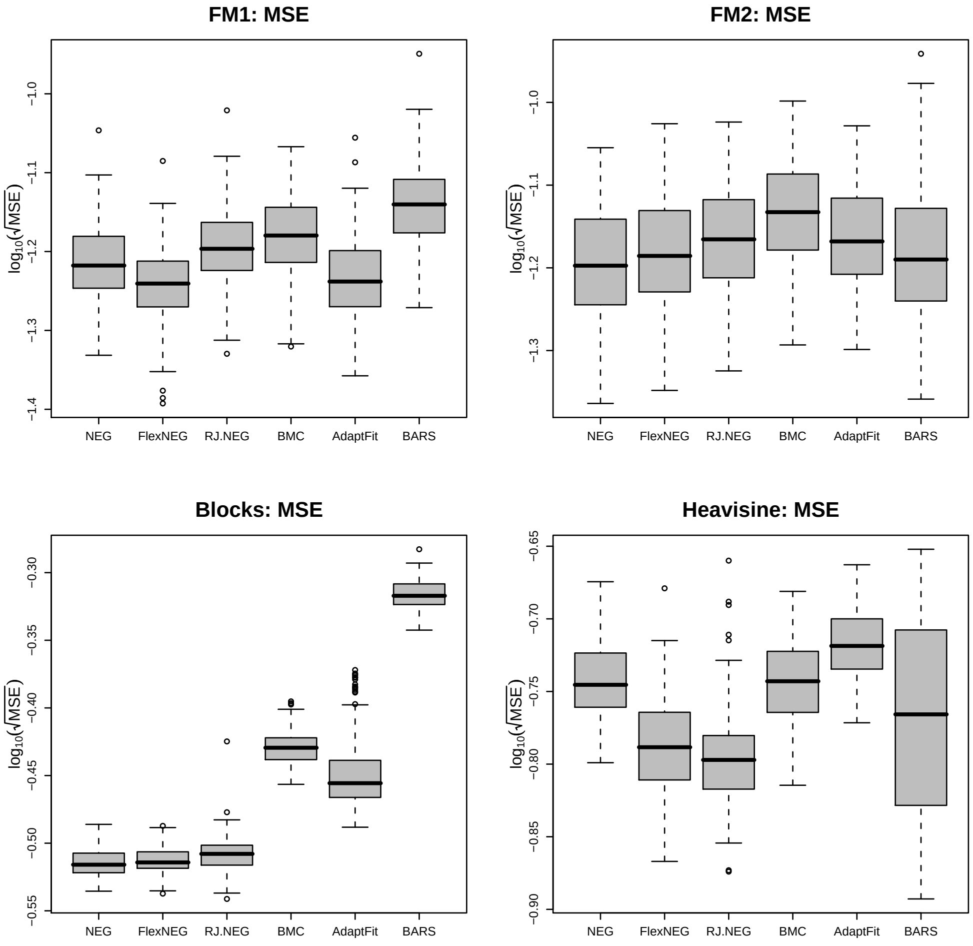


Figure 8.1.: Boxplots of $\log_{10}(\sqrt{MSE})$ for the four benchmark functions. (100 data sets)

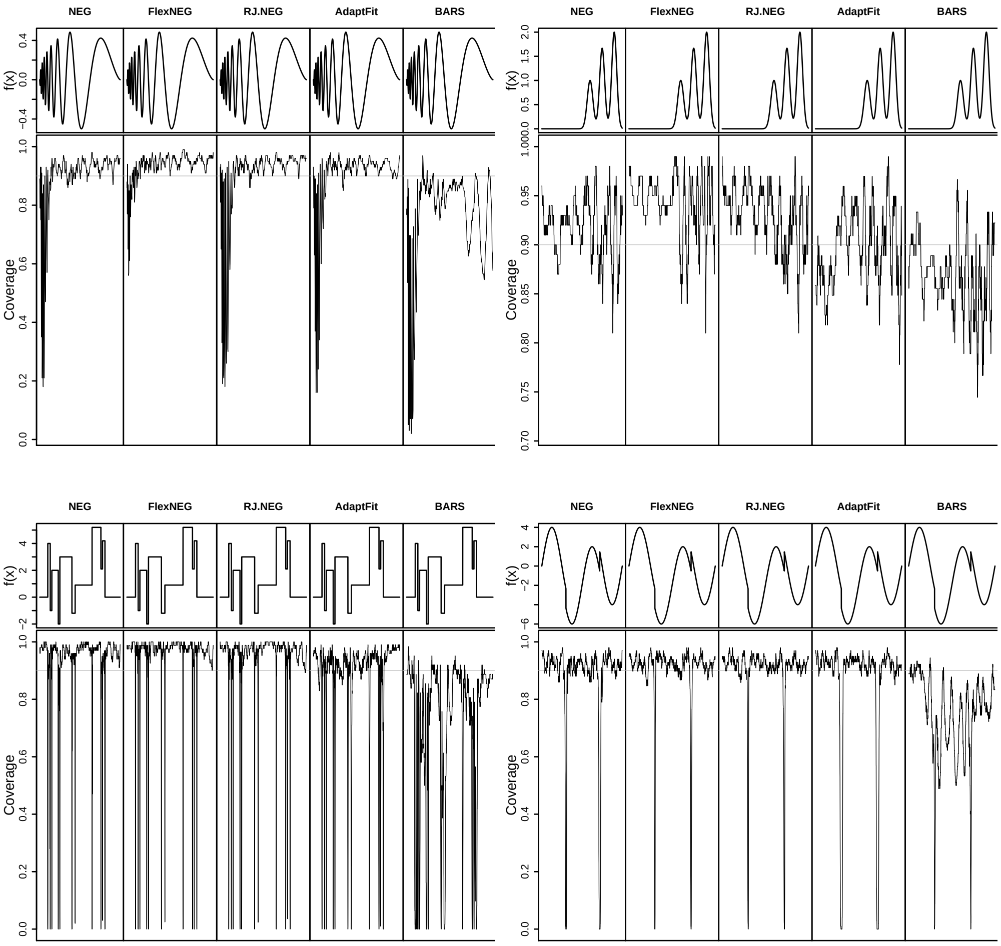


Figure 8.2.: Pointwise coverage (nominal level 90% denoted by grey line) for the four benchmark functions. (clockwise from topleft: FM1, FM2, Heavisine, Blocks; 100 data sets)

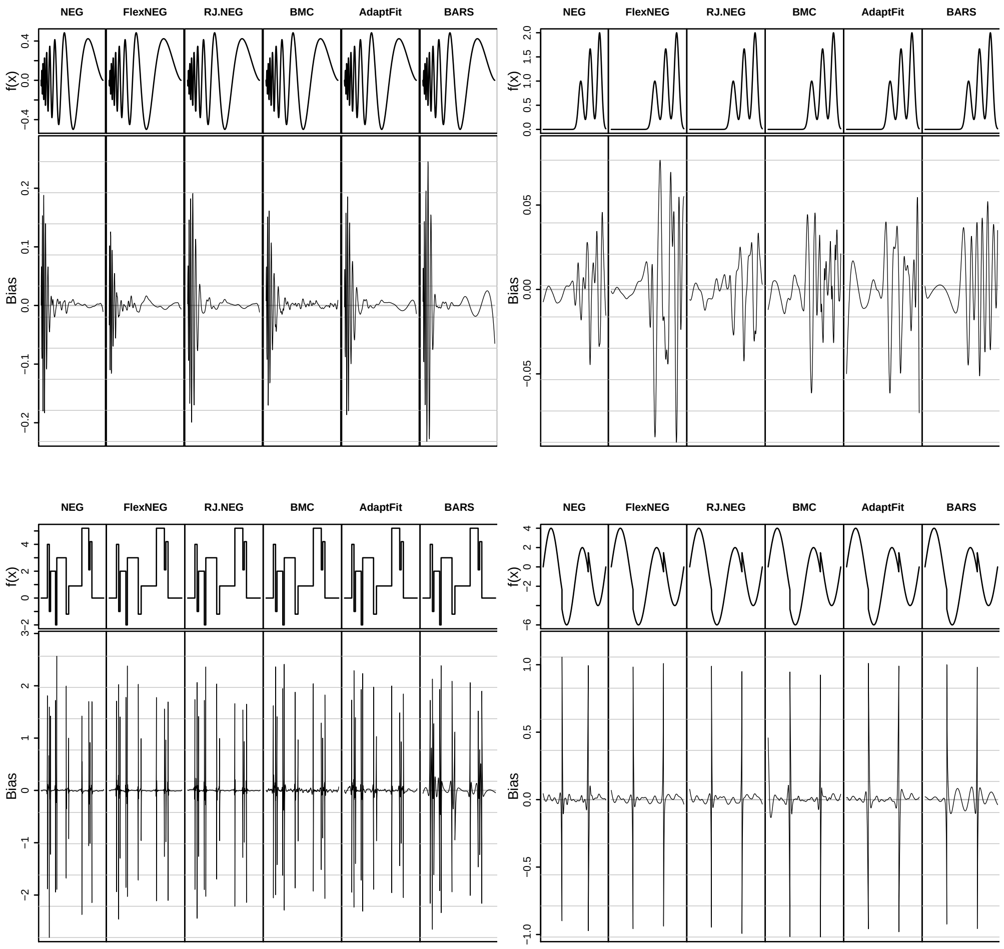


Figure 8.3.: Pointwise observed average bias for the four benchmark functions. (clockwise from topleft: FM1, FM2, Heavisine, Blocks; 100 data sets)

8.1. Oscillating function: FM1

As an example for a function with smoothly varying curvature, we used the doppler-like function

$$m_1(x) = \sqrt{x(1-x)} \sin \frac{18\pi}{8x+1}$$

with $n = 400$ observations and $\sigma_\varepsilon^2 = 0.04$ (SNR $\approx$ 2.1) in accordance with the set-up in Baladandayuthapani et al. (2005) and Krivobokova et al. (2008). Results are based on cubic P-splines ($J = 90; d = 2$) with $B = 5$ for NEG, $B = 10$ for FlexNEG and $s_{\text{mean}} = 5, s_{\text{max}} = 40$ for RJNEG. The mean posterior median of B over the 100 simulations for RJNEG is 5.

Although differences in MSE between the two top competitors FlexNEG (average MSE (AMSE): 0.0034) and AdaptFit (AMSE: 0.0035) are negligible (Fig. 8.1), FlexNEG has a better coverage (Fig. 8.2) and less bias (Fig. 8.3) in the difficult region of the third to sixth oscillations from the left. We were unable to reproduce the results in Baladandayuthapani et al. (2005) which report an AMSE of 0.00028. This is due to a mistake in their simulation design (Baladandayuthapani, 2008), our results give an AMSE of 0.0044 for BMC and an AMSE of 0.00542 for BARS. NEG achieves an AMSE of 0.00385 and RJ.NEG 0.00420. AMSE for nonadaptive fits with `spm` is 0.00659. Average coverage for FlexNEG is slightly conservative (.930), the average coverages of the other methods except BARS are between .895 and .905, while BARS is markedly anti-conservative (0.762). It takes about 60 sec. to generate 5000 iterations of the chain for NEG, 100 sec. for Flex and RJ.NEG, and about 190 sec. for BARS and BMC.

It should be noted that, in the case of FlexNEG, convergence of b_z for this function can be fairly slow if the chain is started with smallish (< 10) values of b_z. For most datasets, there seem to be multiple modes corresponding to different values of b_z and the chain has to be long enough (> 30000 iterations, in one case) to include visits to all of them. Differences between the function estimates from the basins of attraction of the various modes are negligible, however.

8.2. Constant to oscillating function: FM2

An even greater smooth variation in curvature properties is evident in the function

$$m_2(x) = \mathrm{e}^{-400(x-0.6)^2} + \frac{5}{3}\,\mathrm{e}^{-500(x-0.75)^2} + 2\,\mathrm{e}^{-500(x-0.9)^2}.$$

We use datasets with $n = 1000$ observations and $\sigma_\varepsilon^2 = 0.25$ (SNR ≈ 1.2) generated in accordance with the set-up in Baladandayuthapani et al. (2005) and Krivobokova et al. (2008). Results are based on cubic P-splines ($J = 40; d = 2$) with $B = 2$ for NEG, $B = 16$ for FlexNEG and $s_{\text{mean}} = 2, s_{\text{max}} = 10$ for RJNEG. Mean posterior median B for RJNEG is 3.

The NEG-based methods show slightly stronger regularization and, therefore, smaller average bias for the region $x < 0.4$ where the function is constant (Fig. 8.3). We assume this is due to the larger shrinkage of the strongly peaked NEG-prior. AMSEs for RJNEG and AdaptFit are about the same (0.0049) and slightly larger than those for BARS (0.0046), FlexNEG (0.0045) and NEG 0.0043 (Fig. 8.1). AMSE for nonadaptive fits with `spm` is 0.0066. Average coverage for FlexNEG (.94), RJNEG(.93) and NEG (.91) is conservative and anticonservative for BARS .86 (Fig. 8.2). The smaller MSE of the least flexible method – NEG – in this setting can be explained by the shape of the target function: One changepoint located exactly in the middle of the range of the data is obviously close to the optimal choice and the additional flexibility allowed for by the competing methods seems to introduce some detrimental "noise" into the other fits. It takes about 31 sec. for NEG, 67 sec. for Flex and RJ.NEG, about 90 sec. for BMC and about 420 sec. for BARS to generate 5000 iterations of the chain.

8.3. Step function: Blocks

As an example for a very un-smooth function with many discontinuities, we consider the blocks function as specified in Donoho and Johnstone (1994) with $n = 2048$ observations and $\sigma_\varepsilon^2 = 1$ (SNR ≈ 3.7). The blocks function is given as

$$\begin{aligned} m_3(x) &= \sum_{i=1}^{11} \boldsymbol{h}_i \left(1 + \operatorname{sign}(x - \boldsymbol{t}_i)\right), \\ \boldsymbol{h} &= (4, -5, 3, -4, 5, -4.2, 2.1, 4.3, -3.1, 2.1, -4.2)', \\ \boldsymbol{t} &= (.1, .13, .15, .23, .25, .40, .44, .65, .76, .78, .81)'. \end{aligned}$$

Results are based on cubic P-splines ($J = 300; d = 1$) with $B = 150$ for NEG, $B = 45$ for FlexNEG and $s_{\text{mean}} = 50, s_{\text{max}} = 100$ for RJNEG. Mean posterior median B for RJNEG is 43.

As might be expected, both AdaptFit and BMC, which attempt to model a smooth variance function do not perform as well in this situation as the NEG models which use a more flexible piecewise constant representation of the variance function. This can also be seen from the bias plot (Fig. 8.3): Although bias is similarly large at the edges of the respective plateaus for all methods

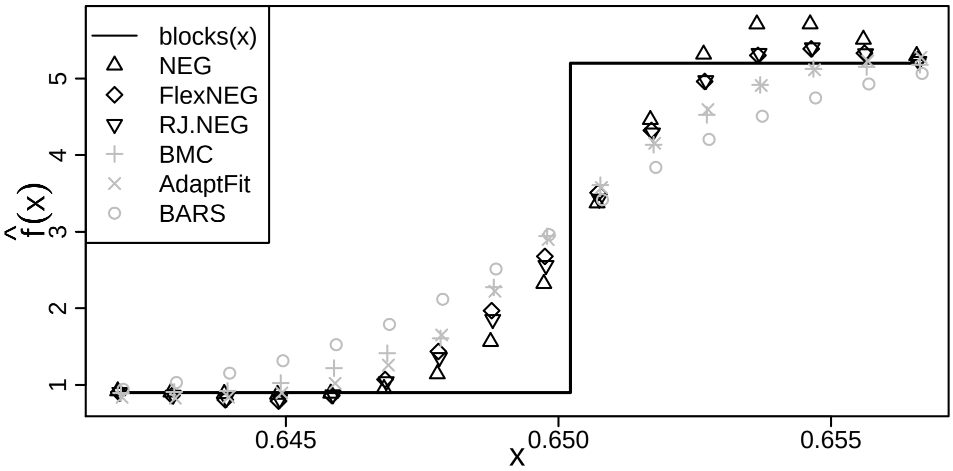


Figure 8.4.: Mean estimates for the blocks function for the discontinuity at $x = 0.65$

– which is due to the inappropriate assumption of a continuous $f(x)$ common to all the models we consider – the NEG-based fits have smaller bias for the plateau regions, because their underlying variance functions return more quickly to much smaller values implying strong regularization and less wiggliness of the fitted function. This can also be seen from the coverage plot (Fig. 8.2): At the discontinuities, FlexNEG's coverage returns above the nominal level more quickly. Fig. 8.4 shows the mean over the 100 estimated functions at the discontinuity at $x = 0.65$ for the various methods. Both BMC and AdaptFit and especially BARS result in a flatter curve that therefore does not reproduce the infinite slope at the jump very well. The basic NEG model shows a steeper slope at the step but overcompensates by adding undesirable variability in the areas where the function should be constant. FlexNEG and RJNEG provide a compromise where we observe a steep slope in combination with very flat curves in the constant part of the function.

AMSE for NEG, FlexNEG and RJNEG is similar $(0.0938, 0.0945$, and 0.0970, respectively). AMSE for AdaptFit is 0.130 and 0.139 for BMC (Fig. 8.1). AMSE for the non-adaptive fits is 0.1786. Note that BARS (AMSE: 0.2336) does not return a fit for 38 of the 100 datasets. Its comparatively large AMSE is probably partly caused by a hyperparameter setting which limits the number of knots to a maximum of 60, which constrains the possible flexibility of the fitted function considerably. Increasing this parameter, however, leads to an even larger number of abortive function calls. RJ.NEG did not return a fit for 12 of the 100 datasets due to numerical difficulties. Average coverage for BARS is severely anti-conservative (.71), slightly anti-conservative for Adapt-

Fit (.87) and conservative (.92 – .94) for the NEG-based methods (Fig. 8.2). It takes about 630 sec. for NEG, 810 sec. for FlexNEG, 830 sec. for RJ.NEG, about 1600 sec. for BMC and about 1200 sec. for BARS to generate 5000 iterations of the chain. A call to AdaptFit, which usually converges in less than 20 sec. for the other settings, typically takes about 420 sec. for this setting due to the complexity of the variance function that has to be estimated.

8.4. Smooth function with discontinuities: Heavisine

A second function with discontinuities but non-constant function values between the jumps is given by the heavisine function as specified in Donoho and Johnstone (1994) with $n = 2048$ observations and $\sigma_\varepsilon^2 = 1$ (SNR $\approx$ 8.8). The heavisine function is given as

$$m_4(x) = 4\sin(4\pi x) - \text{sign}(x - 0.3) - \text{sign}(0.72 - x).$$

Results are based on cubic P-splines ($J = 100; d = 2$) with $B = 10$ for NEG, $B = 30$ for FlexNEG and $s_{\text{mean}} = 60, s_{\text{max}} = 95$ for RJNEG. Mean posterior median B for RJNEG is 42. As for the blocks function, the NEG models are better able to deal with the discontinuities in this function because of the heavy tails of NEG prior and the ability of the piecewise constant variance function to model short spikes in variability. While the maximal bias values at the discontinuities themselves are practically identical for all methods, FlexNEG and RJNEG have smaller bias (Fig. 8.3) and better coverage (Fig. 8.2) in the proximity of the discontinuities. Fig. 8.5 shows the square root of estimated variance functions for an exemplary dataset. FlexNEG, RJNEG and, to a lesser extent due to its less flexible parametrization, NEG all show pronounced spikes in variance around the two discontinuities of the function, while the variance function estimated by AdaptFit does not capture the true structure of the variability.

AMSEs for RJNEG (0.0261) and FlexNEG (0.0269) are similar, followed by BARS (0.0302), NEG (0.0328), and BMC (0.0330). AdaptFit (0.369) is outperformed by the non-adaptive `spm` (0.0366) in this case. Note the large variability in the MSEs for BARS, which achieves both the best and the worst fits depending on the dataset (Fig. 8.1). Average coverage for both FlexNEG and RJ.NEG is .90, .88 for NEG, .87 for AdaptFit and severely anti-conservative (0.75) for BARS (Fig. 8.2). Generating 5000 iterations of the chain takes about 95 sec. for NEG, 145 sec. for FlexNEG and RJ.NEG, about 185 sec. for BMC and about 1000 sec. for BARS.

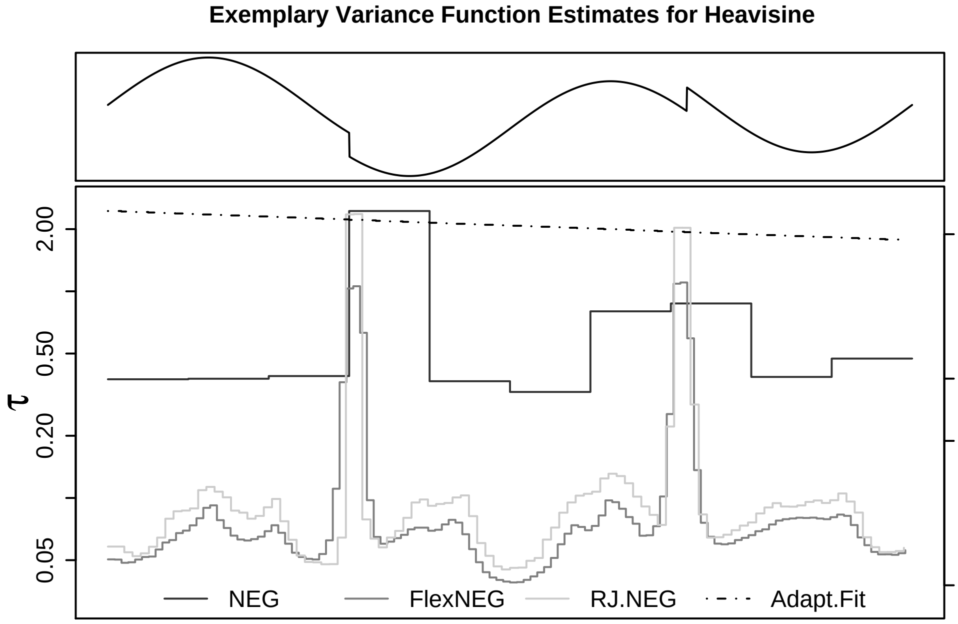


Figure 8.5.: Square root of exemplary estimated variance functions for the heavisine function. Note the much larger scale for AdaptFit given on the right side of the plot.

8.5. Simulations for non-Gaussian responses

We evaluate the performance of FlexNEG for both binary Binomial and Poisson responses with logit and log link functions, respectively. We use $\log_{10}\left(\sqrt{MSE}\right) = \log_{10}\left(\sqrt{\frac{1}{n}\sum(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}))^2}\right)$ on the scale of the linear predictor as performance measure to evaluate the quality of the achieved fits.

For the first setting, we generate 100 Poisson data sets ($n = 1000$) with means $\exp(m_2(x))$, where $m_2(\cdot)$ is the same as in section 8.2 and compare the performance of FlexNEG to both AdaptFit and BARS. We use cubic P-splines with a first order difference penalty on $J = 40$ basis functions with $B = 10$. Chains are run for 8000 iterations after a burn-in of 2000 iterations. AdaptFit is also run with 40 basis functions, but only 5 basis functions for the variance function since the use of 10 basis functions would have caused non-convergence for a large majority of the simulated datasets. We use the default parameters for BARS. The upper panel of Fig. 8.6 displays exemplary fits and data. The distribution of observed $\log_{10}(\sqrt{MSE})$ on the scale of the linear predictor for this setting is displayed in the left panel of Fig. 8.7. Differences in $\log_{10}(\sqrt{MSE})$ are fairly small, average $\log_{10}(\sqrt{MSE})$ based on the available cases is -1.025 for FlexNEG, -0.959 for AdaptFit and -0.988 or BARS. Note, however, that BARS does not return a fit for 40 of the 100 datasets and that AdaptFit does not converge for 14 datasets, while FlexNEG works for all 100 datasets. Average $\log_{10}(\sqrt{MSE})$ for the non-adaptive fits is -0.907 in this setting. Especially for FlexNEG, local adaptivity produces an improved fit for the first part of the function by removing the spurious oscillations of the non-adaptive fit. Average coverage for both FlexNEG and BARS is reasonably close to the nominal level of 0.9 (FlexNEG: 0.94, BARS: 0.87) and markedly anti-conservative for AdaptFit (0.79). Running a chain with 5000 iterations takes about 320 sec. with BARS and 140 sec. with FlexNEG. A call to AdaptFit that converges usually takes about 30 sec. , while non-convergent calls can take up to 90 sec.

We also generated 100 Poisson data sets ($n = 1200$) with means $\exp(m_3(x))$ for $0 < x < .5$, where the blocks function $m_3(\cdot)$ is the same as in section 8.3 and compare the performance of FlexNEG to BARS. A comparison with AdaptFit is not possible in this case because AdaptFit fails to converge 9 out of 10 times for this setting. Results for FlexNEG are based on cubic P-splines with a first order difference penalty on $J = 90$ basis functions with $B = 20$. Chains are run for 10000 iterations after a burn-in of 2000 iterations. The

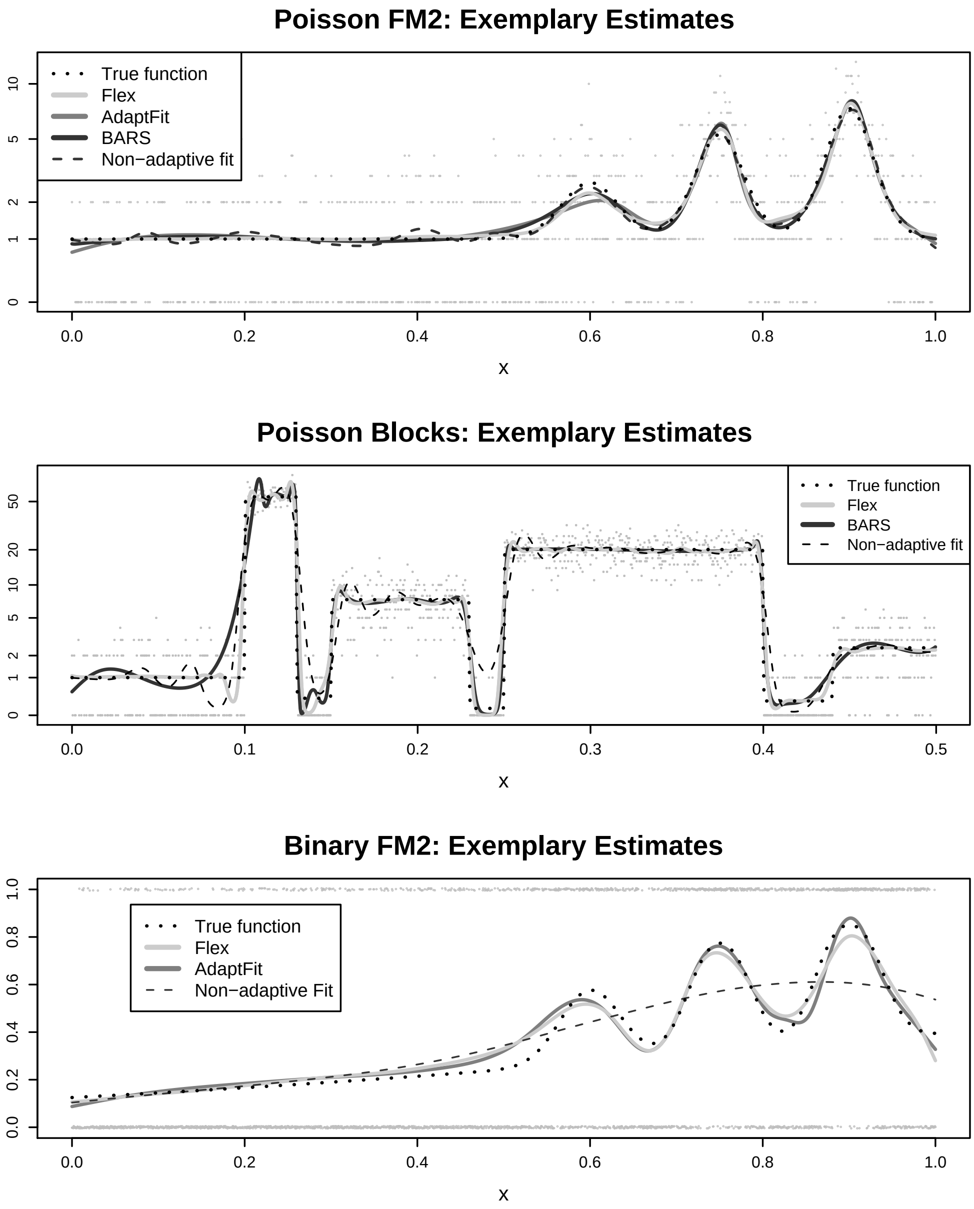


Figure 8.6.: Exemplary fits for non-Gaussian responses. Data are indicated by grey dots.

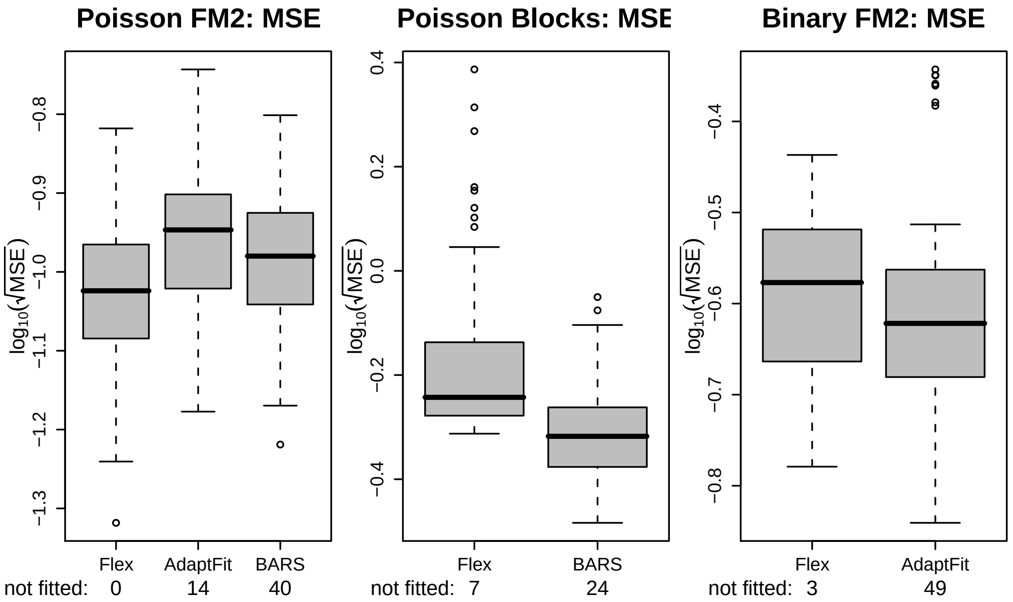


Figure 8.7.: Boxplots of $\log_{10}(\sqrt{MSE})$ (on the scale of the linear predictor) for non-Gaussian responses. (100 data sets, boxplots display all available MSEs)

middle panel of Fig. 8.6 displays exemplary fits, the middle panel of Fig. 8.7 shows observed $\log_{10}(\sqrt{MSE})$ on the scale of the linear predictor. Average $\log_{10}(\sqrt{MSE})$ for BARS is -0.314 and -0.179 for FlexNEG. However, BARS does not return a fit for 24 of the 100 datasets while FlexNEG only fails to fit 7. These 7 failures are all caused by numerical problems in the automatic method to find suitable starting values for β. The locally adaptive methods are better at reproducing the small plateaus and valleys between $x = .1$ and $x = .3$ than the non-adaptive method (`spm`) which achieved an average $\log_{10}(\sqrt{MSE})$ of 0.269. Average coverage is markedly anti-conservative for both FlexNEG (0.79) and BARS (0.81) in this setting. Running a chain with 5000 iterations takes about 620 sec. with BARS and 300 sec. with FlexNEG.

Finally, we also generated 100 data sets ($n = 2000$) of binary Binomial responses with $\mu(x) = \left(\text{logit}^{-1}(x + m_2(x))\right)^3$, where $m_2(\cdot)$ is the same as in section 8.2. We chose this shape for $\mu(x)$ in order to increase the signal in the data in this difficult setting compared to $\mu(x) = \text{logit}^{-1}(m_2(x))$, where $\mu(x)$ mostly varies between .5 and .7. We only compared the performance of FlexNEG to that of AdaptFit since there is no implementation of BARS for Binomial responses. Results for FlexNEG are based on cubic P-splines with a first order difference penalty on $J = 40$ basis functions with $B = 10$. AdaptFit is also run with 40 basis functions, but only 8 basis functions for the variance function since the use of 10 basis functions would have caused non-convergence for an even larger portion of the simulated datasets. The lower panel of Fig. 8.6 displays exemplary fits and data. The distribution of observed $\log_{10}(\sqrt{MSE})$ on the scale of the linear predictor is depicted in the lower panel of Fig. 8.7. The difference in average $\log_{10}(\sqrt{MSE})$ (FlexNEG: -0.592, AdaptFit: -0.611) is fairly small for the available cases. Note, however, that AdaptFit failed to fit 49 of the 100 datasets while FlexNEG only failed in 3 cases. These 3 problematic cases were all caused by a failure of the automatic method to find suitable starting values for β. Also note the large benefit associated with using local adaptivity. Average $\log_{10}(\sqrt{MSE})$ for non-adaptive fits with `spm` is -0.442 in this setting. Average coverage for a nominal level of 0.9 is anti-conservative for FlexNEG (0.80) and even more so for AdaptFit (0.69). Running a chain with 5000 iterations takes about 240 sec. with FlexNEG. A call to AdaptFit that converges usually takes about 80 sec., while non-convergent calls can take up to 200 sec.

8.6. Quantitative analysis of simulation performances

Following the suggestions in Hothorn, Leisch, Zeileis, and Hornik (2005), we quantify the observed differences in $\log_{10}(\sqrt{MSE})$ for Gaussian response via a linear mixed effects model (R-package `lme4` (Bates and Maechler, 2009)). We include random effects for the simulated samples to account for their varying "difficulty" as well as random effects for the interaction between benchmark functions and algorithms. Fig. 8.8 shows estimated (partial) Tukey contrasts for the main effects of the algorithms with 95% confidence intervals corrected for multiple comparisons (single-step correction as implemented in R-package `multcomp` (Hothorn, Bretz, Westfall, and Heiberger, 2008)). Differences in performance between the 3 NEG-based approaches are not significant, with some evidence for a slight advantage for FlexNEG. FlexNEG performs significantly better than AdaptFit. AdaptFit, BARS and BMC are outperformed by all our methods, BMC significantly so. The very large intervals for the comparisons with BARS are due to the reduced sample sizes for BARS caused by its frequent crashing and by its comparatively large variability in MSE, especially in settings FM2 and Heavisine. Note that the estimated differences are quite relevant: an average difference in $\log_{10}(\sqrt{MSE})$ of -0.05 corresponds to a decrease in AMSE by about 20%.

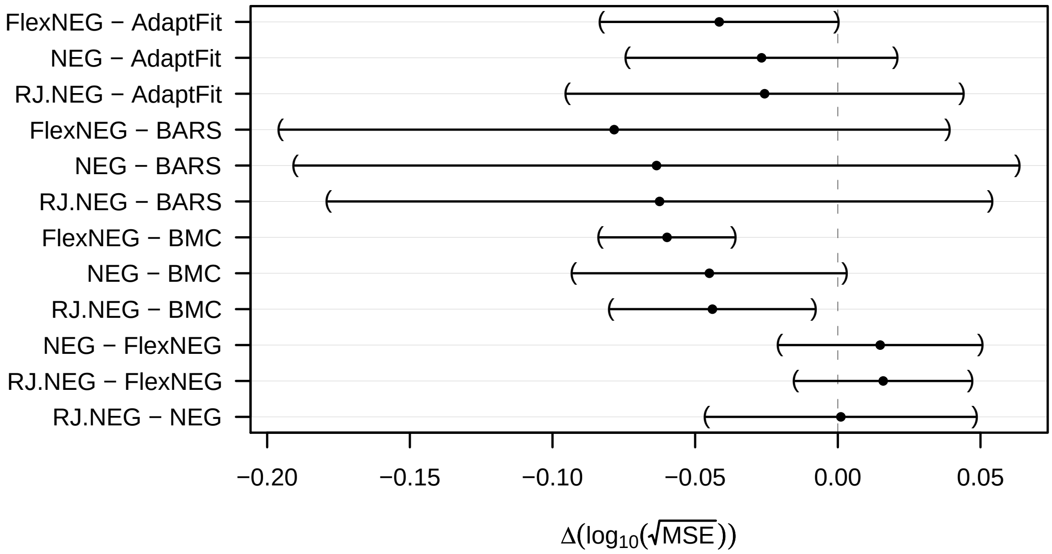


Figure 8.8.: 95% family-wise confidence intervals and point estimates for differences in $\log_{10}(\sqrt{MSE})$ between algorithms

8.7. Robustness

8.7.1. Signal-to-noise ratio

We investigated the change in MSE for various signal-to-noise ratios (SNR) for the four benchmark functions (see sections 8.1 to 8.4) for FlexNEG and compared it with the results of AdaptFit. Figure 8.9 shows that the change in MSE is about the same for both methods, with slight differences that do not yield a conclusive picture for small and medium SNR. With the possible exception of the Blocks function, FlexNEG seems to improve more strongly than AdaptFit for large SNR.

8.7.2. Number of changepoints

We investigated the change in MSE for varying specifications of B or $(s_{\text{mean}}, s_{\text{max}})$. Other parameters correspond to the settings given in sections 8.1 to 8.4. Figure 8.10 shows that RJNEG's performance is mostly stable as long as the number of admitted changepoints is large enough, while FlexNEG and NEG can lose a little performance for both too small and too large B. Note, however, that the performances of both NEG and FlexNEG still compare favorably to those of the non-NEG methods we considered even for sub-optimal, but reasonable values of B and that the increase in MSE is relatively small in most cases. In order to see whether the best number of

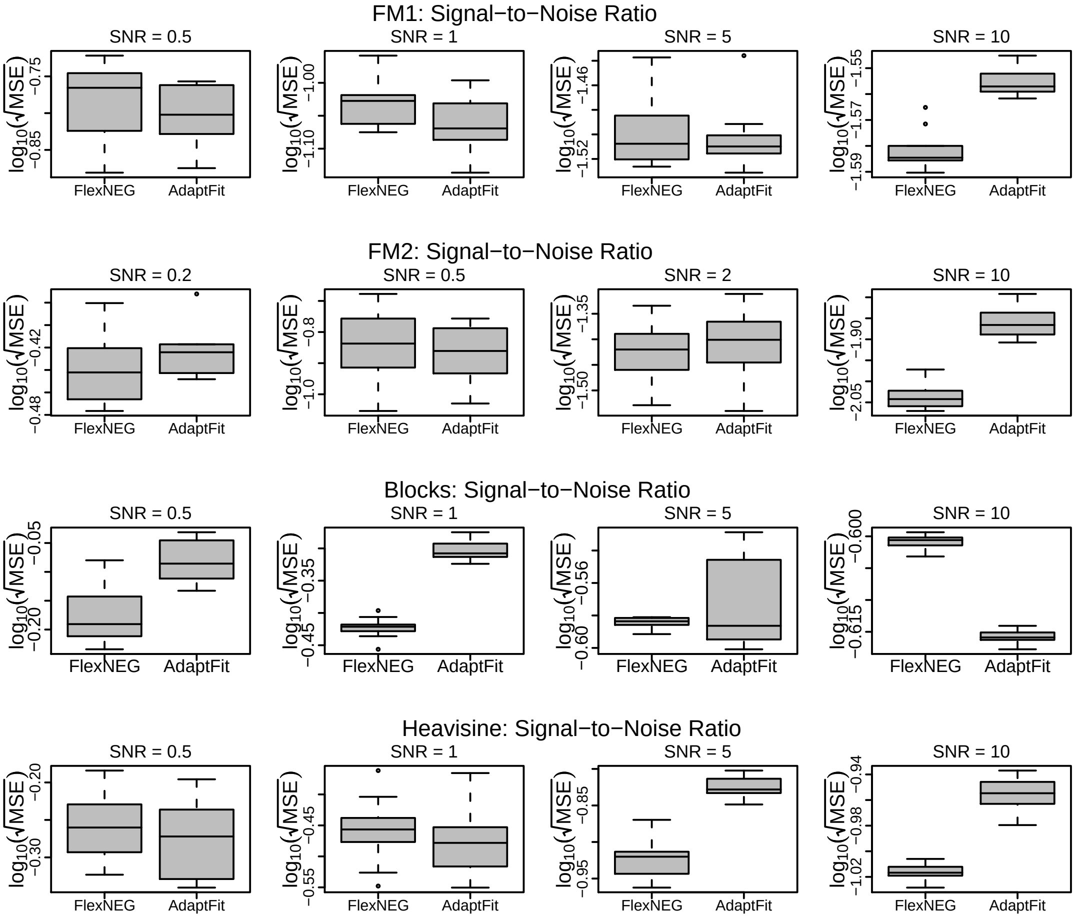


Figure 8.9.: Boxplots of MSE for FlexNEG and AdaptFit for various signal-to-noise ratios. 10 datasets per SNR; settings correspond to sections 8.1 to 8.4, respectively.

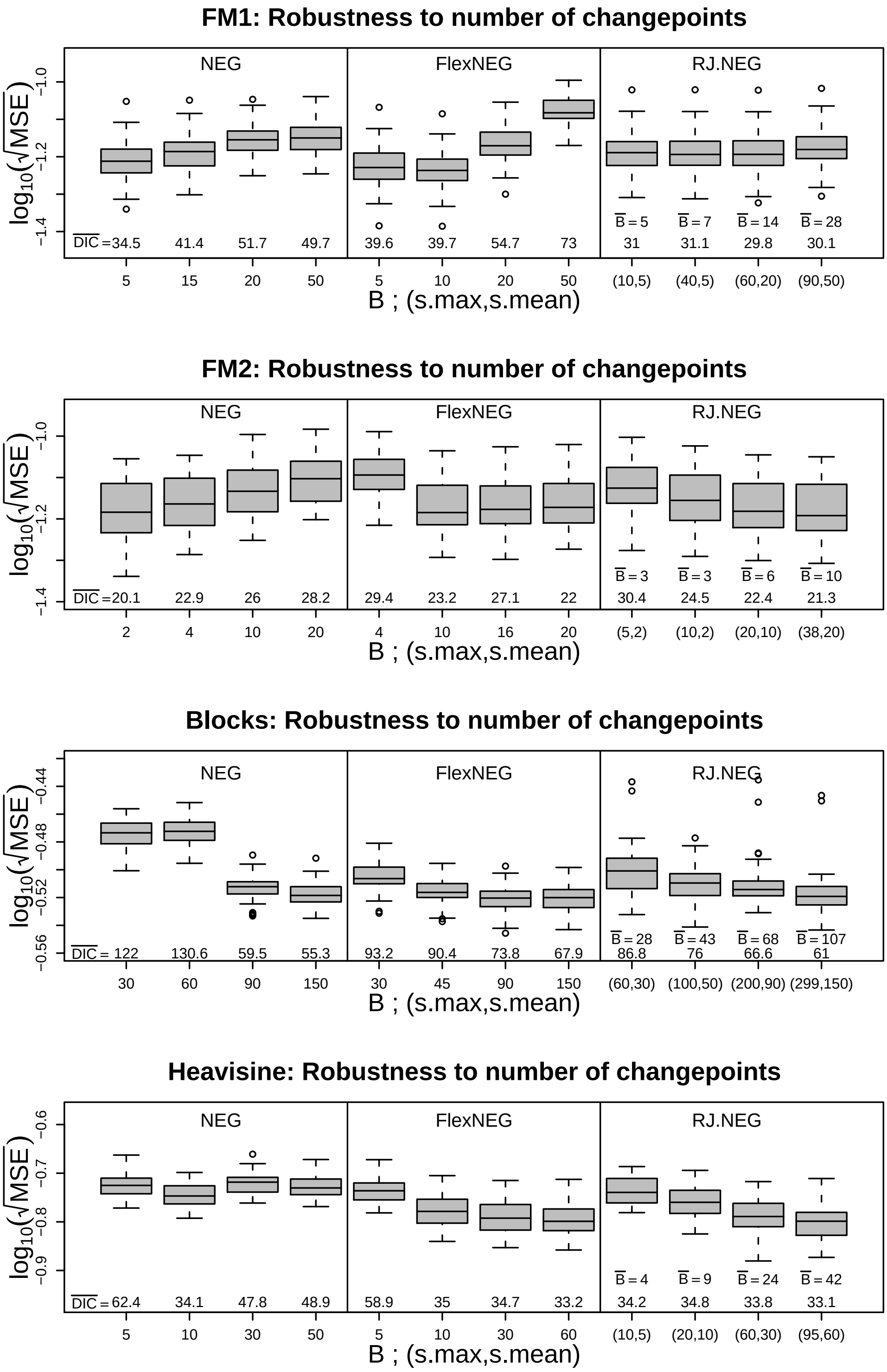


Figure 8.10.: Boxplots of MSEs for various settings of B or $(s_{\max}, s_{\text{mean}})$. 50 datasets per value of B; other parameters as in settings of sections 8.1 to 8.4, respectively. $\bar{B}$ is the (rounded) mean of posterior means of B for RJ.NEG. $\overline{\text{DIC}}$ is the (rounded) mean DIC over the 50 datasets.

changepoints in real-world applications could be determined by the deviance information criterion (DIC) (Spiegelhalter, Best, Carlin, and van der Linde, 2002) we computed DICs for the simulation runs. Model selection based on DIC works well for all three methods and simulated datasets: As Fig. 8.10 indicates, the AMSE-optimal setting corresponds to the one with the lowest average DIC whenever there is relevant sensitivity of the fits to the number of changepoints. DIC-based model selection works best for NEG, while more or less complex models than the MSE-optimal models (albeit with very similar MSEs) are selected some of the time for FlexNEG and RJ.NEG. The relative increase in MSE for the models with sub-optimally DIC-selected hyperparameters compared to the MSE-optimal model on the same dataset was typically between 2% and 10% ($+0.0045$ to $+0.023$ on $\log_{10}(\sqrt{MSE})$ scale). We conclude that, while DIC may not always succeed at identifying the most parsimonious model among the models with similar MSEs, it seems to select useful and sensible hyperparameter values fairly reliably in the settings we considered.

9. Applications

9.1. Fractionation curves

We apply our method to exemplary data from "Specificity Assessment From Fractionation Experiments" (SAFE) (Drobyshev, Machka, Horsch, Seltmann, Liebscher, de Angelis, Beckers, and Journals, 2003) which are used for quality control of cDNA microarray experiments. Specifically, SAFE is used to investigate the degree of undesirable cross-hybridization of specific probe strands, e.g. how often cDNA sections pair with cDNA probes on the chip which have a similar, but not exactly equal, base sequence. For SAFE, microarray chips are repeatedly treated with formamide solutions of increasing concentration and intensities are recorded after each washing. The series of resulting intensities for each probe on the chip is called a fractionation curve. As the cohesion between cross-hybridizing cDNA strands is weaker than between perfect matches, they are washed away at lower concentrations. If cross-hybridization occurs, there usually is a critical concentration in the lower range where a certain kind of cDNA sequence cross-hybridizing the probe sequence is abruptly washed away and a drop in signal intensity occurs.

9.1.1. Results

Fits are based on P-splines of degree 0 with $J = 20$ basis functions and first order difference penalty for both the NEG-based methods and the non-adaptive fit with `mgcv::gam` (Wood, 2010a) we used for comparison. Note that the response vector was standardized to have zero mean and unit variance to allow fitting with the standard choices for the hyperprior parameters.

The left panel of Fig. 9.1 shows an example of a spot binding only the correct complementary cDNA. The location of the sharp decrease at about 65% indicates that the binding energy between complementary strands was no longer sufficient for cohesion at this concentration. The right panel shows an example of a spot with cross-hybridization, where cross-hybridizing strands are washed away at a concentration of about 15%. We use the deviance information criterion (DIC) to choose B from 3, 5, and 10 and $(s_{\text{max}}, s_{\text{mean}})$ from $(19, 10)$, $(10, 5)$, and $(5, 3)$ for NEG, FlexNEG and RJNEG, respectively. We validated the use of DIC for model selection in this setting on simulated data of similar structure and noise level, as described in Section 9.1.2. Generating

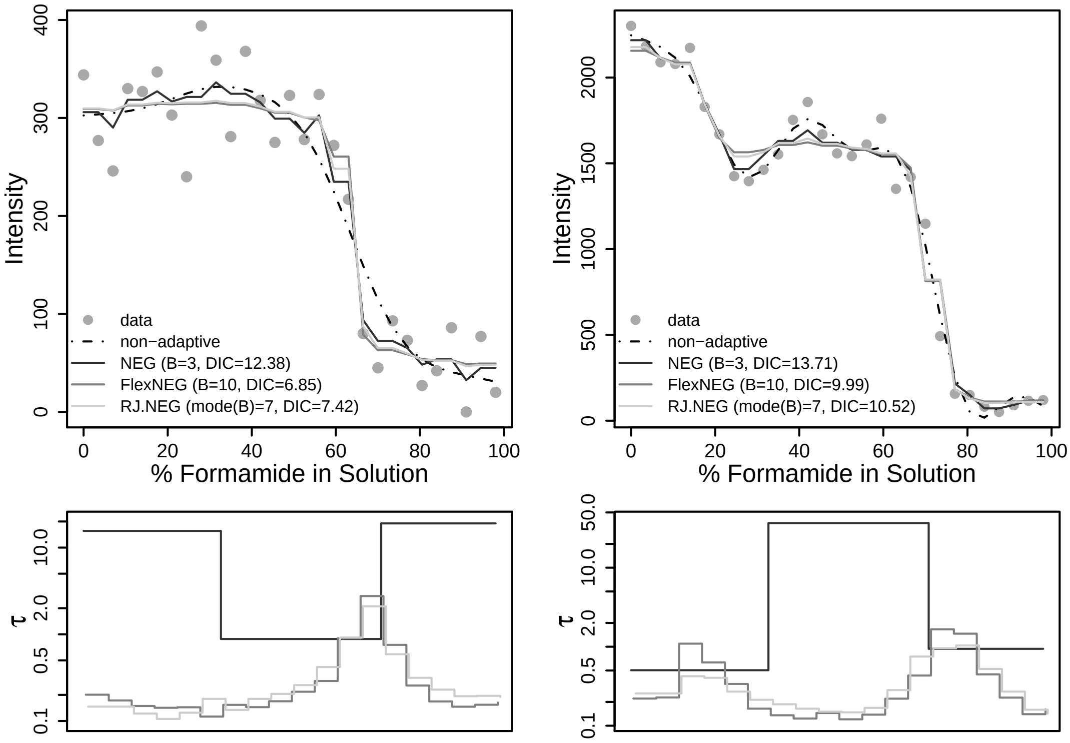


Figure 9.1.: Two exemplary fractionation data sets and fitted functions. Lower panel shows the square root of the estimated variance functions for NEG, FlexNEG and RJNEG (log-scale). Values on the abscissa of the lower panel are jittered to avoid overplotting.

5000 iterations of the chain takes about 17 sec. for NEG., 50 sec. for FlexNEG and 57 sec. for RJ.NEG.

Fig.9.1 shows that, even without the explicit monotonicity constraints appropriate for this data, both FlexNEG and RJNEG reproduce the piecewise constant and decreasing structure that is expected fairly well. Note also that despite ignoring the expected structure of a step function, the results also give an indication of the number and the location of jumps in the estimated function when looking at the estimated variance function for FlexNEG and RJNEG. While the left panel clearly shows a single jump, there are two distinct jump points in the right panel. As a consequence, our approach would also support inference about the location of the jump points that could be extracted from the MCMC samples making it a valuable alternative to step-function based approaches where this is typically difficult.

While the peaks in the variance function for both FlexNEG and RJNEG correspond exactly to the observable changepoints in the data, the variance function of NEG shows a somewhat surprising behavior (at least in the left panel) due to the very low number of blocks in the DIC-optimal model. However, this can be explained when comparing the corresponding fit to the non-adaptive one. For example, in the first interval containing the first third of the data, both the non-adaptive fit and the NEG fit are very wiggly, corresponding to a large value of the variance function. This seems to be caused by an attempt to fit local outliers due to the low signal in this area. A similar behavior is observed for the third interval, whereas the second interval, containing the jump, is assigned a small variance to obtain a smooth fit due to the high signal induced by the jump. In general, the non-adaptive fit exhibits excessive wiggliness for low concentrations in the left panel and for intermediate concentrations in the right panel which shows the improvement that can be gained by an adaptive fit in this context.

9.1.2. Validating DIC as model selection criterion

We ran an additional simulation study on datasets similar in structure and noise level to the fractionation curve data to validate the use of DIC for model selection in the application. Specifically, we generated 50 datasets with $n = 30$ for a regularly spaced grid on $0.3 < x < 0.5$ with $\boldsymbol{y} = m_3(\boldsymbol{x}) + \varepsilon$; $\varepsilon_i \sim \mathcal{N}(0, 0.2)$. Two exemplary datasets are plotted in Fig. 9.2. For all 50 datasets we fit NEG and FlexNEG with $B = 3, 5, 10$ and RJNEG with $(s_{\max}, s_{\text{mean}})$ from $(19, 10)$, $(10, 5)$, and $(5, 3)$ as for the fractionation curve data and calculated DIC. Model selection based on DIC works very well for RJ.NEG and FlexNEG, selecting the MSE-optimal model 49 times and 50 times out of 50, respectively. Since results for NEG did not improve noticeably between $B = 3$ and $B = 5$, model selection based on DIC selected the MSE-optimal model

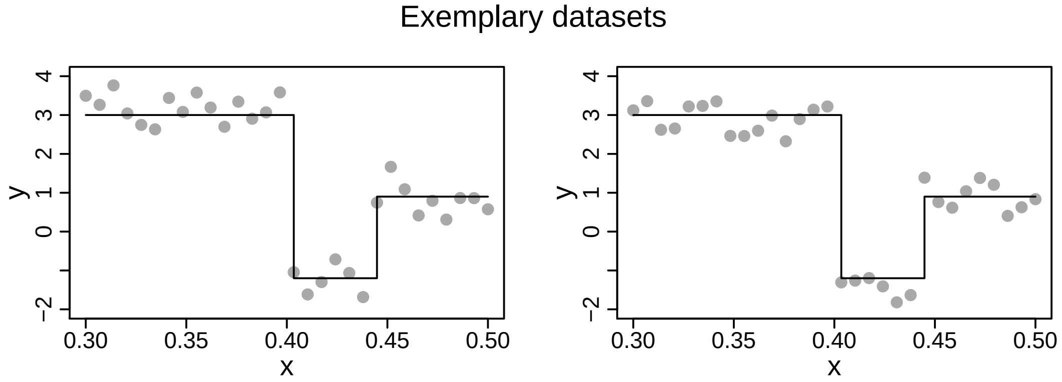


Figure 9.2.: 2 exemplary simulated datasets used for the validation of DIC as model selection criterion. ($n = 30$ data points in dark grey, true function in black)

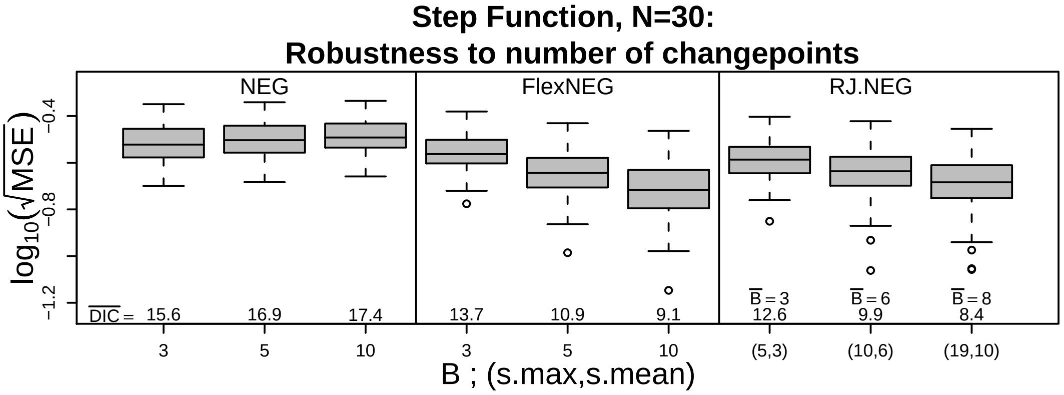


Figure 9.3.: Boxplots of MSEs for various settings of B or $(s_{\text{max}}, s_{\text{mean}})$. 50 datasets per value of B; other parameters as in the application on the real fractionation curve data. $\bar{B}$ is the (rounded) mean of posterior means of B for RJ.NEG. $\overline{\text{DIC}}$ is the (rounded) mean DIC over the 50 datasets.

only 17 times out of 50 and selected the most parsimonious model – which had the second best MSE – for 28 of the 33 remaining cases. This behavior makes sense from a modeling point of view, as the relative increase in MSE for the models with sub-optimally DIC-selected hyperparameters compared to the MSE-optimal model was typically only between 5% and 16% ($+0.011$ to $+0.032$ on $\log_{10}(\sqrt{MSE})$ scale).

9.2. Neuron spike train data

We consider the performances of BARS, FlexNEG and non-adaptive fits done with `mgcv::gam` for three peri-stimulus time histograms of neuronal spiking events across time displayed in figure 9.4. The data are taken from the `e060817mix` dataset available in the R-package `STAR` (Pouzat, 2008). Neuronal spiking events are assumed to follow a Poisson process, so that the numbers of events in subsequent small time intervals form a sequence of Poisson-distributed counts. In the experiment we consider here, the activity of 3 neurons in the antennal lobe of cockroaches during spontaneous activity and during an odor impulse occurring between 6.01s and 6.51s were recorded for 20 replications of 15 seconds each and aggregated into 50 bins of 0.3s. Neurophysiological prior knowledge implies that the rate of the underlying Poisson process may vary fairly little most of the time, but possibly rapidly in the short interval of the experimental stimulus, which suggests the use of locally adaptive methods for estimation of the intensity function. Fits are based on cubic P-splines with $J = 40$ basis functions and second order difference penalty for both FlexNEG and the non-adaptive fit with `mgcv::gam` we used for comparison. We again employ the deviance information criterion (DIC) to choose B from 3, 5, and 10 for FlexNEG. Differences between the FlexNEG fits and their DICs were small, and the smallest DIC was obtained for $B = 10$ for all three datasets.

Fig. 9.4 shows that adaptive as well as non-adaptive methods fit the expected large spikes or drops between 6.01s and 6.51s well for all 3 neurons. The fits for the data set in the left panel are very similar: BARS estimates the smoothest function on both sides of the spike, while the FlexNEG fit is a little more wiggly. FlexNEG as well as BARS avoid the likely spurious small oscillations of the non-adaptive fit.

The data in the middle panel seems to be much more volatile, and the differences between the three method are more pronounced. Note that BARS reproduces all of the oscillations before the spike – the fit is very similar to the overly ragged non-adaptive fit – and none of them after the spike, while FlexNEG arguably shows more reasonable estimates of both the slower oscillations on the left and the faster oscillations on the right of the spike.

For the data on the left, FlexNEG may undersmooth before the drop and

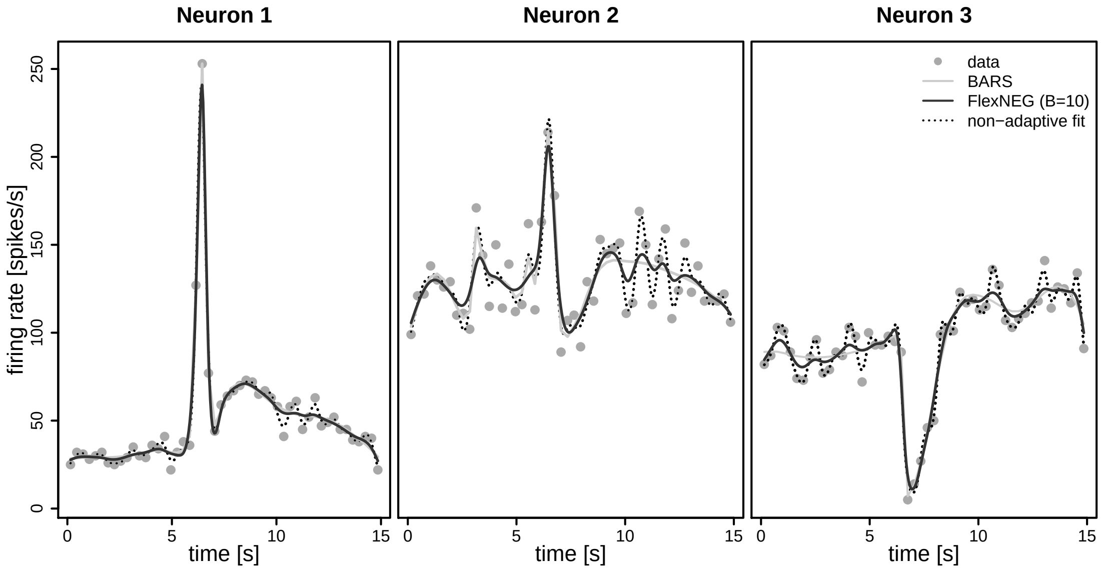


Figure 9.4.: Three neuron spike train data sets (peri-stimulus time histograms) and fitted functions.

seems to offer a compromise between the extremely wiggly fit estimated by the non-adaptive method and the possibly oversmoothed fit produced by BARS.

All three data sets show the benefits that can be obtained by locally adaptive smoothing. Even for this kind of application which seems to have motivated the development of BARS (Wallstrom, 2006), FlexNEG offers reasonable and competitive fits that avoid the excessive wiggliness of the non-adaptive function estimates and still reproduce both large spikes and drops and smaller features well.

10. Conclusions

In this part, we showed how the normal-exponential-gamma prior hierarchy, combined with a flexible piecewise constant representation of the local smoothing parameter, can be used for locally adaptive smoothing linear and generalized linear models. We see the main strengths of this approach in

1. its ability to deal with both discontinuous changes in the complexity of the fitted function and smoothly varying local complexity. We found the adaptive NEG prior P-splines to be a good competitor to previous approaches for smoothly varying variability and improved performance for functions with discontinuities.

2. its fast convergence and, for FlexNEG and RJ.NEG, wide insensitivity to starting values due to the satisfactory mixing provided by the blockwise Gibbs samplers. Even for the very heavily parameterized Blocks function (> 400 parameters) a burn-in period of about 5000 iterations is sufficient, while, for example, a burn-in period of at least 50000 iterations is recommended (personal comm. V. Baladandayuthapani) for the MH-based sampler by Baladandayuthapani et al. (2005). As a comparison of the computation times shows, the implementations of the NEG-based methods are also very competitive in terms of speed to the other fully Bayesian methods we considered.

3. its automatic applicability, since results for FlexNEG and RJNEG are fairly robust against the user-specified hyperparameters which limit the maximal complexity of the implied variance function for the random walk increments of the spline coefficients.

Although we generally found robustness of FlexNEG and RJNEG with respect to hyperparameter settings, selecting a DIC-optimal number of changepoints often allowed to find competitive solutions even with the basic NEG prior with fixed number and location of changepoints. Still, FlexNEG and RJNEG have the advantage that no multiple runs are required to find the optimal number of changepoints and that they are rather insensitive to starting values. They also allow to carry forward the uncertainty introduced by estimating the number and location of the changepoints.

Further work could embed our approach in a Bayesian backfitting algorithm to enable locally adaptive function estimation in the more general framework of structured additive regression models where only some effects require local adaptivity. A further direction of future investigations could be the consideration of bivariate smoothing with adaptive penalized splines as considered for example in Krivobokova et al. (2008) and an extension for heteroscedastic errors as in Crainiceanu et al. (2007). The modularity of MCMC will be of particular value when considering such extensions.

Bibliography

A. Asuncion and D.J Newman. *UCI Machine Learning Repository*, 2007. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

K. Bae and B.K. Mallick. Gene selection using a two-level hierarchical Bayesian model. *Bioinformatics*, 20(18):3423–3430, 2004.

J. Baglama and L. Reichel. Augmented implicitly restarted Lanczos bidiagonalization methods. *SIAM Journal on Scientific Computing*, 27(1):19–42, 2006.

V. Baladandayuthapani. Correction to "Spatially adaptive Bayesian penalized regression splines (P-splines)", 2008. URL `http://pubs.amstat.org/doi/pdf/10.1198/106186008X322517`.

V. Baladandayuthapani, B.K. Mallick, and R.J. Carroll. Spatially adaptive Bayesian penalized regression splines (P-splines). *Journal of Computational and Graphical Statistics*, 14(2):378–394, 2005.

D. Bates and M. Maechler. *lme4: Linear mixed-effects models using S4 classes*, 2009. URL `http://CRAN.R-project.org/package=lme4`. R package version 0.999375-33.

C.B. Begg and R. Gray. Calculation of polychotomous logistic regression parameters using individualized regressions. *Biometrika*, 71:11–18, 1984.

C. Biller. Adaptive Bayesian regression splines in semiparametric generalized linear models. *Journal of Computational and Graphical Statistics*, 9(1):122–140, 2000.

A. Brezger and S. Lang. Generalized structured additive regression based on Bayesian P-splines. *Computational Statistics & Data Analysis*, 50(4):967–991, 2006.

B. Cai and D.B. Dunson. Bayesian covariance selection in generalized linear mixed models. *Biometrics*, 62(2):446–457, 2006.

B.P. Carlin and S. Chib. Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(3):473–484, 1995.

C.M. Carvalho, N.G. Polson, and J.G. Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

R. Cottet, R.J. Kohn, and D.J. Nott. Variable selection and model averaging in semiparametric overdispersed generalized linear models. *Journal of the American Statistical Association*, 103 (482):661–671, 2008.

C.M. Crainiceanu, D. Ruppert, R.J. Carroll, A. Joshi, and B. Goodner. Spatially adaptive Bayesian penalized splines with heteroscedastic errors. *Journal of Computational and Graphical Statistics*, 16(2):265–288, 2007.

J.S. Dagpunar. An easily implemented generalised inverse Gaussian generator. *Communications in Statistics-Simulation and Computation*, 18(2):703–710, 1989.

P. Dellaportas, J.J. Forster, and I. Ntzoufras. On Bayesian model and variable selection using MCMC. *Statistics and Computing*, 12(1):27–36, 2002.

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

I. Dimatteo, C.R. Genovese, and R.E. Kass. Bayesian curve-fitting with free-knot splines. *Biometrika*, 88(4):1055, 2001.

D.L. Donoho and I.M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81:425–455, 1994.

A.L. Drobyshev, C. Machka, M. Horsch, M. Seltmann, V. Liebscher, M.H. de Angelis, J. Beckers, and O. Journals. Specificity assessment from fractionation experiments (SAFE): a novel method to evaluate microarray probe specificity based on hybridisation stringencies. *Nucleic Acids Research*, 31(2):e1, 2003.

P.H.C. Eilers and B.D. Marx. Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–102, 1996.

M.J.A. Eugster, T. Hothorn, and F. Leisch. Exploratory and inferential analysis of benchmark experiments. Technical Report 30, Department of Statistics, LMU München, 2008. URL `http://epub.ub.uni-muenchen.de/4134/`.

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

L. Fahrmeir, T. Kneib, and S. Konrath. Bayesian regularisation in structured additive regression: a unifying perspective on shrinkage, smoothing and predictor selection. *Statistics and Computing*, 20(2):203–219, 2010.

M. Friedman. Piecewise exponential models for survival data with covariates. *The Annals of Statistics*, pages 101–113, 1982.

S. Frühwirth-Schnatter and R. Tüchler. Bayesian parsimonious covariance estimation for hierarchical linear mixed models. *Statistics and Computing*, 18(1):1–13, 2008.

S. Frühwirth-Schnatter and H. Wagner. Bayesian variable selection for random intercept modelling of gaussian and non-gaussian data. In J.M. Bernardo, M.J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, editors, *Bayesian Statistics 9*. Oxford University Press, 2010.

D. Gamerman. Efficient sampling from the posterior distribution in generalized linear models. *Statistics and Computing*, 7:57–68, 1997.

A. Gelman, D.A. Van Dyk, Z. Huang, and J.W. Boscardin. Using redundant parameterizations to fit hierarchical models. *Journal of Computational and Graphical Statistics*, 17(1):95–122, 2008.

R. Gentleman. *muhaz: Hazard Function Estimation in Survival Analysis*, 2010. URL `http://CRAN.R-project.org/package=muhaz`. R package version 1.2.5.

E.I. George and R.E. McCulloch. Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

R.B. Gramacy. *monomvn: Estimation for multivariate normal and Student-t data with monotone missingness.*, 2010. URL `http://CRAN.R-project.org/package=monomvn`. R package version 1.8-3.

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

S. Greven. *Non-Standard Problems in Inference for Additive and Linear Mixed Models*. Cuvillier Verlag, 2007.

J.E. Griffin and P.J. Brown. Alternative prior distributions for variable selection with very many more variables than observations. Technical Report UKC/IMS/05/08, IMS, University of Kent, 2005.

J.E. Griffin and P.J. Brown. Bayesian adaptive lassos with non-convex penalization. Technical Report No. 07-02, University of Warwick, 2007. URL `http://www2.warwick.ac.uk/fac/sci/statistics/crism/research/working_papers/2007/paper07-2/07-2wv2.pdf`.

C. Gu. Diagnostics for nonparametric regression models with additive terms. *Journal of the American Statistical Association*, 87(420):1051–1058, 1992.

C. Gu. *Smoothing Spline ANOVA Models*. Springer-Verlag, 2002.

E.G. Harrell. *Regression Modeling Strategies*. Springer New York, 2001.

T. Hastie. Pseudosplines. *Journal of the Royal Statistical Society B*, 58(2):379–396, 1996.

J.S. Hodges and B.J. Reich. Adding spatially-correlated errors can mess up the fixed effect you love. *The American Statistician*, 64(4):325–334, 2010.

J.A. Hoeting, D. Madigan, A.E. Raftery, and C.T. Volinsky. Bayesian model averaging: a tutorial. *Statistical Science*, 14(4):382–417, 1999.

B. Hofner, T. Kneib, W. Hartl, and H. Küchenhoff. Building cox-type structured hazard regression models with time-varying effects. *Statistical Modelling*, 2010. to appear.

T. Hothorn, F. Leisch, A. Zeileis, and K. Hornik. The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics*, 14(3):675–699, 2005.

T. Hothorn, F. Bretz, P. Westfall, and R.M. Heiberger. *multcomp: Simultaneous Inference for General Linear Hypotheses*, 2008. R package version 0.993-1.

T. Hothorn, P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner. *mboost: Model-Based Boosting*, 2010. R package version 2.0-0.

J. Hughes and M. Haran. Dimension reduction and alleviation of confounding for spatial generalized linear mixed models. *Journal of the Royal Statistical Society B*, 2011. submitted.

H. Ishwaran and J.S. Rao. Spike and slab variable selection: Frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773, 2005.

S. Jackman. *Bayesian Analysis for the Social Sciences*. Wiley, 2009.

B. Jørgensen. *Statistical properties of the generalized inverse Gaussian distribution*. Springer-Verlag, 1982.

A. Jullion and P. Lambert. Robust specification of the roughness penalty prior distribution in spatially adaptive Bayesian P-splines models. *Computational statistics & data analysis*, 51(5): 2542–2558, 2007.

T. Kneib. *Mixed model based inference in structured additive regression*. Dr. Hut Verlag, 2006. URL `http://edoc.ub.uni-muenchen.de/archive/00005011/`.

T. Kneib, S. Konrath, and L. Fahrmeir. High-dimensional structured additive regression models: Bayesian regularisation, smoothing and predictive performance. *Applied Statistics*, 2010. *to appear*.

A. Komárek, E. Lesaffre, and J.F. Hilton. Accelerated failure time model for arbitrarily censored data with smoothed error distribution. *Journal of Computational and Graphical Statistics*, 14(3):726–745, 2005.

T. Krivobokova. *AdaptFit: Adaptive Semiparametric Regression*, 2007. R package version 0.2-1.

T. Krivobokova, C.M. Crainiceanu, and G. Kauermann. Fast adaptive penalized splines. *Journal of Computational and Graphical Statistics*, 17(1):1–20, 2008.

L. Kuo and B. Mallick. Variable selection for regression models. *Sankhyā: The Indian Journal of Statistics, Series B*, 60(1):65–81, 1998.

N. Laird and D. Olivier. Covariance analysis of censored survival data using log-linear analysis techniques. *Journal of the American Statistical Association*, pages 231–240, 1981.

S. Lang and A. Brezger. Bayesian P-splines. *Journal of Computational and Graphical Statistics*, 13(1):183–212, 2004.

F. Leisch and E. Dimitriadou. *mlbench: Machine Learning Benchmark Problems*, 2010. R package version 2.1-0.

B.W. Lewis. *irlba: Fast Partial SVD by Implicitly-Restarted Lanczos Bidiagonalization*, 2009. URL http://www.rforge.net/irlba/. R package version 0.1.1.

F. Li and N.R. Zhang. Bayesian variable selection in structured high-dimensional covariate spaces with applications in genomics. *JASA Theory and Methods*, 105:1202–1214, 2010.

Y. Lin and H.H. Zhang. Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, 34(5):2272–2297, 2006.

J. Lokhorst, B. Venables, B. Turlach, and M. Maechler. */pkglasso2: L1 constrained estimation.*, 2009. URL http://CRAN.R-project.org/package=lasso2. R package version 1.2-10.

X.L. Meng and D. van Dyk. The EM algorithm–an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society B*, 59(3):511–567, 1997.

D. Meyer, F. Leisch, and K. Hornik. The support vector machine under test. *Neurocomputing*, 55(1-2):169–186, 2003.

T.J. Mitchell and J.J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

R.B. O'Hara and M.J. Sillanpää. A review of Bayesian variable selection methods: What, how, and which? *Bayesian Analysis*, 4(1):85–118, 2009.

A. Panagiotelis and M. Smith. Bayesian identification, selection and estimation of semiparametric functions in high-dimensional additive models. *Journal of Econometrics*, 143(2): 291–316, 2008.

T. Park and G. Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103 (482):681–686, 2008.

N.G. Polson and J.G. Scott. Shrink globally, act locally: Sparse Bayesian regularization and prediction. In J.M. Bernardo, M.J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, editors, *Bayesian Statistics 9*. Oxford University Press, 2010.

C. Pouzat. *STAR: Spike Train Analysis with R*, 2008. URL http://www.biomedicale.univ-paris5.fr/physcerv/C_Pouzat/STAR.html. R package version 0.1-9.

A.E. Raftery and Y. Zheng. Discussion of "frequentist model average estimators". *Journal of the American Statistical Association*, 98(464):931–938, 2003.

A.E. Raftery, J. Hoeting, C. Volinsky, I. Painter, and K.Y. Yeung. *BMA: Bayesian Model Averaging*, 2011. URL `http://CRAN.R-project.org/package=BMA`. R package version 3.14.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. URL `http://www.R-project.org`.

B.J. Reich, C.B. Storlie, and H.D. Bondell. Variable selection in Bayesian smoothing spline anova models: Application to deterministic computer codes. *Technometrics*, 51(2):110, 2009.

H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications*. Chapman & Hall, 2005.

F. Ruëff, B. Przybilla, M.B. Biló, U. Müller, F. Scheipl, W. Aberer, J. Birnbaum, A. Bodzenta-Lukaszyk, F. Bonifazi, C. Bucher, et al. Predictors of severe systemic anaphylactic reactions in patients with hymenoptera venom allergy: Importance of baseline serum tryptase–a study of the European Academy of Allergology and Clinical Immunology Interest Group on Insect Venom Hypersensitivity. *Journal of Allergy and Clinical Immunology*, 124(5):1047–1054, 2009.

D. Ruppert and R.J. Carroll. Spatially-adaptive penalties for spline fitting. *Australian & New Zealand Journal of Statistics*, 42(2):205–223, 2000.

F. Scheipl. *RLRsim: Exact (Restricted) Likelihood Ratio tests for mixed and additive models.*, 2010a. URL `http://CRAN.R-project.org/package=RLRsim`. R package version 2.0-4.

F. Scheipl. Normal-mixture-of-inverse-gamma priors for Bayesian regularization and model selection in generalized additive models. Technical Report 84, Department of Statistics, LMU München, 2010b. URL `http://epub.ub.uni-muenchen.de/11785/`.

F. Scheipl. *amer: Additive Mixed Models with lme4*, 2010c. URL `http://CRAN.R-project.org/package=amer`. R package version 0.6.6.

F. Scheipl. *spikeSlabGAM: Bayesian Model Selection for Generalized Additive Mixed Models*, 2010d. R package version 0.3-12.

F. Scheipl, S. Greven, and H. Küchenhoff. Size and power of tests for a zero random effect variance or polynomial regression in additive and linear mixed models. *Computational Statistics & Data Analysis*, 52(7):3283–3299, 2008.

M. Smith and L. Fahrmeir. Spatial Bayesian variable selection with application to functional magnetic resonance imaging. *Journal of the American Statistical Association*, 102(478):417–431, 2007.

D.J. Spiegelhalter, N.G. Best, B.P. Carlin, and A. van der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, 64 (4):583–616, 2002.

C.B. Storlie, H.D. Bondell, B.J. Reich, and H.H. Zhang. Surface estimation, variable selection, and the nonparametric oracle property. *Statistica Sinica*, 2011. *to appear*.

S. Sturtz, U. Ligges, and A. Gelman. `R2WinBUGS`: A package for running WinBUGS from R. *Journal of Statistical Software*, 12(3):1–16, 2005.

L. Tierney, A.J. Rossini, Na Li, and H. Sevcikova. *snow: Simple Network Of Workstations*, 2010. R package version 0.3-3.

S. Urbanek. *multicore: Parallel Processing of R Code on Machines with Multiple Cores or CPUs*, 2010. URL http://www.rforge.net/multicore/. R package version 0.1-3.

G. Wahba, Y. Wang, C. Gu, R. Klein, and B. Klein. Smoothing spline anova for exponential families, with application to the wisconsin epidemiological study of diabetic retinopathy. *The Annals of Statistics*, 23(6):1865–1895, 1995.

G. Wallstrom. BARS: Bayesian adaptive regression splines, 2006. URL http://www.stat.cmu.edu/~kass/bars/bars.html.

M.P. Wand, B.A. Coull, J.L. French, B. Ganguli, E.E. Kammann, J. Staudenmayer, and A. Zanobetti. *SemiPar 1.0*, 2005. URL http://cran.r-project.org/web/packages/SemiPar/index.html. R package version 1.0.

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, 2009. URL http://had.co.nz/ggplot2/book.

S. Wood. *mgcv: GAMs with GCV/AIC/REML smoothness estimation and GAMMs by PQL*, 2010a. R package version 1.7-2.

S. Wood, R. Kohn, T. Shively, and W. Jiang. Model selection in spline nonparametric regression. *Journal of the Royal Statistical Society B*, 64(1):119–139, 2002.

S.N. Wood. Thin-plate regression splines. *Journal of the Royal Statistical Society B*, 65(1):95–114, 2003.

S.N. Wood. *Generalized Additive Models: an Introduction with R*. CRC Press, 2006.

S.N. Wood. Fast stable direct fitting and smoothness selection for generalized additive models. *Journal of the Royal Statistical Society B*, 70(3):495, 2008.

S.N. Wood. *gamm4: Generalized additive mixed models using mgcv and lme4*, 2010b. URL http://CRAN.R-project.org/package=gamm4. R package version 0.1-0.

S.N. Wood, F. Scheipl, and J.J. Faraway. On intermediate rank tensor product smoothing. submitted, 2011.

P. Yau, R. Kohn, and S. Wood. Bayesian variable selection and model averaging in high-dimensional multinomial nonparametric regression. *Journal of Computational and Graphical Statistics*, 12(1):23–54, 2003.

List of Figures

List of Tables