This is a precompiled PDF. Please visit the Github repo for the executable vignette with code.

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

Fabian Scheipl
Ludwig-Maximilians-Universität München

Abstract

The R package **spikeSlabGAM** implements Bayesian variable selection, model choice, and regularized estimation in (geo-)additive mixed models for Gaussian, binomial, and Poisson responses. Its purpose is to (1) choose an appropriate subset of potential covariates and their interactions, (2) to determine whether linear or more flexible functional forms are required to model the effects of the respective covariates, and (3) to estimate their shapes. Selection and regularization of the model terms is based on a novel spike-and-slab-type prior on coefficient groups associated with parametric and semi-parametric effects.

Note 1: An earlier version of this introduction to **spikeSlabGAM** has been published as Fabian Scheipl (2011). **spikeSlabGAM**: Bayesian Variable Selection, Model Choice and Regularization for Generalized Additive Mixed Models in R. *Journal of Statistical Software*, **43**(14), 1–24.

Keywords: MCMC, P-splines, spike-and-slab prior, normal-inverse-gamma.

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 relationships between covariates and the response is a challenging and important task. From a Bayesian perspective, it can be translated into a question of estimating marginal posterior probabilities of whether a variable should be in the model and in what form (i.e., linear or smooth; as a main effect and/or as an effect modifier).

We introduce the R (R Development Core Team 2010) package **spikeSlabGAM** which implements fully Bayesian variable selection and model choice with a *spike-and-slab* prior structure that expands the approach in Ishwaran and Rao (2005) to select or deselect single coefficients as well as blocks of coefficients associated with specific model terms. The spike-and-slab priors we use are bimodal priors for the hyper-variances of the regression coefficients which result in a two component mixture of a narrow spike around zero and a slab with wide support for the marginal prior of the coefficients themselves. The posterior mixture weights for the spike component for a specific coefficient or coefficient batch can be interpreted as the posterior probability of its exclusion from the model.

The coefficient batches selected or deselected in this fashion can be associated with a wide variety of model terms such as simple linear terms, factor variables, basis expansions for the modeling of smooth curves or surfaces, intrinsically Gaussian Markov random fields (IGMRF), random effects, and all their interactions. **spikeSlabGAM** is able to deal with Gaussian, binomial and Poisson responses, and can be used to fit piecewise exponential models for time-to-event data. For these response types, the package presented here implements regularized estimation, term selection, model choice, and model averaging for a similarly broad class of models as that available in **mboost** (Hothorn *et al.* 2010) or BayesX (Brezger *et al.* 2005). To the best of our knowledge, it is the first implementation of a Bayesian model term selection method that: (1) is able to fit models for non-Gaussian responses from the exponential family; (2) selects and estimates many types of regularized effects with a (conditionally) Gaussian prior such as simple covariates (both metric and categorical), penalized splines (uni- or multivariate), random effects, spatial effects (kriging, IGMRF) and their interactions; (3) and can distinguish between smooth nonlinear and linear effects. The approach scales reasonably well to datasets with thousands of observations and a few hundred coefficients and is available in documented open source software.

Bayesian function selection, similar to the frequentist COSSO procedure (Lin and Zhang 2006), is usually based on decomposing the additive model in the spirit of a smoothing spline ANOVA (Wahba *et al.* 1995). Wood *et al.* (2002) and Yau *et al.* (2003) describe procedures for Gaussian and latent Gaussian models 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 that also allows for flexible modeling of the dispersion in double exponential regression models is described in Cottet *et al.* (2008), but no implementation is available. Reich *et al.* (2009) also use the smoothing spline ANOVA framework and perform variable and function selection via SSVS for Gaussian responses. Frühwirth-Schnatter and Wagner (2010) discuss various spike-and-slab prior variants for the selection of random intercepts for Gaussian and latent Gaussian models.

The remainder of this paper is structured as follows: Section 2 gives some background on the two main ideas used in **spikeSlabGAM**. 2.1 introduces the necessary notation for the generalized additive mixed model and 2.2 fills in some details on the spike-and-slab prior. Section 3 relates details of the implementation: how the design matrices for the model terms are constructed (Section 3.1) and how the MCMC sampler works (Section 3.2). Section 4 explains how to specify, visualize and interpret models fitted with **spikeSlabGAM** and contains an application to the Pima Indian Diabetes dataset.

2. Background

2.1. Generalized additive mixed models

The generalized additive mixed model (GAMM) is a broad model class that forms a subset of structured additive regression (Fahrmeir *et al.* 2004). In a GAMM, the distribution of the responses $\boldsymbol{y}$ given a set of covariates $\boldsymbol{x}_j$ $(j = 1, \ldots, p)$ belongs to an exponential family, i.e.,

$$\pi(y|x, \phi) = c(y, \phi) \exp\left(\frac{y\theta - b(\theta)}{\phi}\right),$$

with θ,ϕ,$b(\cdot)$ and $c(\cdot)$ determined by the type of distribution. The conditional expected value of the response $\mathrm{E}(\boldsymbol{y}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p) = h(\boldsymbol{\eta})$ is determined by the additive predictor $\boldsymbol{\eta}$ and a fixed response function $h(\cdot)$.

The additive predictor

$$\boldsymbol{\eta} = \boldsymbol{\eta}_o + \boldsymbol{X}_u\boldsymbol{\beta}_u + \sum_{j=1}^{p} f_j(\boldsymbol{x}) \tag{1}$$

has three parts: a fixed and known offset $\boldsymbol{\eta}_o$, a linear predictor $\boldsymbol{X}_u\boldsymbol{\beta}_u$ for model terms that are not under selection with coefficients $\boldsymbol{\beta}_u$ associated with a very flat Gaussian prior (this will typically include at least a global intercept term), and the model terms $f_j(\boldsymbol{x}) = (f_j(\boldsymbol{x}_1), \ldots, f_j(\boldsymbol{x}_n))^\top$ $(j = 1, \ldots, p)$ that are each represented as linear combinations of d_j basis functions $B_j(\cdot)$ so that

$$\begin{aligned} f_j(\boldsymbol{x}) = \sum_{k=1}^{d_j} \beta_{jk} B_{jk}(\boldsymbol{x}) = \boldsymbol{B}_j\boldsymbol{\beta}_j, \text{ with } B_{jk}(\boldsymbol{x}) = \left(B_{jk}(\boldsymbol{x}_1), \ldots, B_{jk}(\boldsymbol{x}_n)^\top\right) \\ \text{and } \boldsymbol{\beta}_j \overset{\text{prior}}{\sim} \text{peNMIG}(v_0, w, a_\tau, b_\tau) \text{ for } j = 1, \ldots, p. \end{aligned} \tag{2}$$

The peNMIG prior structure is explained in detail in Section 2.2.

Components $f_j(\boldsymbol{x})$ of the additive predictor represent a wide variety of model terms, such as (1) linear terms $(f_j(\boldsymbol{x}) = \beta_j \boldsymbol{x}_j)$, (2) nominal or ordinal covariates $(f(x_{ji}) = \beta_{x(k)}$ iff $x_{ji} = k$, i.e., if entry i in $\boldsymbol{x}_j$ is $k)$, (3) smooth functions of (one or more) continuous covariates (splines, kriging effects, tensor product splines or varying coefficient terms, e.g., Wood (2006)), (4) Markov random fields for discrete spatial covariates (e.g. Rue and Held 2005), (5) random effects (subject-specific intercepts or slope), and (6) interactions between the different terms (varying-coefficient models, effect modifiers, factor interactions). Estimates for semiparametric model terms and random effects are regularized in order to avoid overfitting and modeled with appropriate shrinkage priors. These shrinkage or regularization priors are usually Gaussian or can be parameterized as scale mixtures of Gaussians (e.g. Fahrmeir *et al.* 2010). The peNMIG variable selection prior used in **spikeSlabGAM** can also be viewed as a scale mixture of Gaussians.

2.2. Stochastic search variable selection and spike-and-slab priors

While analyses can benefit immensely from a flexible and versatile array of potential model terms, the large number of possible models in any given data situation calls for a principled procedure that is able to select the covariates that are relevant for the modeling effort (i.e., variable selection) as well as to determine the shapes of their effects (e.g., smooth vs. linear) and which interaction effects or effect modifiers need to be considered (i.e., model choice). SSVS and spike-and-slab priors are Bayesian methods for these tasks that do not rely on the often very difficult calculation of marginal likelihoods for large collections of complex models (e.g. Han and Carlin 2001).

The basic idea of the SSVS approach (George and McCulloch 1993) is to introduce a binary latent variable γ_j associated with the coefficients $\boldsymbol{\beta}_j$ of each model term so that the contribution of a model term to the predictor is forced to be zero – or at least negligibly small – if γ_j is in one state and left unchanged if γ_j is in the other state. The posterior distribution of γ_j can be interpreted as marginal posterior probabilities for exclusion or inclusion of the respective model term. The posterior distribution of the vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)^\top$ can be interpreted as posterior probabilities for the different models represented by the configurations of $\boldsymbol{\gamma}$. Another way to express this basic idea is to assume a spike-and-slab mixture prior for each $\boldsymbol{\beta}_j$, with one component being a narrow spike around the origin that imposes very strong shrinkage on the coefficients and the other component being a wide slab that imposes very little shrinkage on the coefficients. The posterior weights for the spike and the slab can then be interpreted analogously.

The flavor of spike-and-slab prior used in **spikeSlabGAM** is a further development based on Ishwaran and Rao (2005): The basic prior structure, which we call a Normal - mixture of inverse Gammas (NMIG) prior, uses a bimodal prior on the variance v^2 of the coefficients that results in a spike-and-slab type prior on the coefficients themselves. For a scalar β, the prior structure is given by:

$$\begin{aligned}
\beta|\gamma, \tau^2 &\overset{\text{prior}}{\sim} N(0, v^2) \text{ with } v^2 = \tau^2\gamma, \\
\gamma|w &\overset{\text{prior}}{\sim} wI_1(\gamma) + (1-w)I_{v_0}(\gamma), \\
\tau^2 &\overset{\text{prior}}{\sim} \Gamma^{-1}(a_\tau, b_\tau), \\
\text{and } w &\overset{\text{prior}}{\sim} \text{Beta}(a_w, b_w).
\end{aligned} \tag{3}$$

$I_x(y)$ denotes a function that is 1 in x and 0 everywhere else and v_0 is some small positive constant, so that the indicator γ is 1 with probability w and close to zero with probability $1-w$. This means that the effective prior variance v^2 is very small if $\gamma = v_0$ — this is the spike part of the prior. The variance τ^2 is sampled from an informative Inverse Gamma (Γ^{-1}) prior with density $p(x|a,b) = \frac{b^a}{\Gamma(a)} x^{(a-1)} \exp\left(-\frac{b}{x}\right)$.

This prior hierarchy has some advantages for selection of model terms for non-Gaussian data since the selection (i.e., the sampling of indicator variables γ) occurs on the level of the coefficient variance. This means that the likelihood itself is not in the Markov blanket of γ and consequently does not occur in the full conditional densities (FCD) for the indicator variables, so that the FCD for γ is available in closed form regardless of the likelihood. However, since only the regression coefficients and not the data itself occur in the Markov blanket of $\boldsymbol{\gamma}$, inclusion or exclusion of model terms is based on the magnitude of the coefficients $\boldsymbol{\beta}$ and not on the magnitude of the effect $\boldsymbol{B}\boldsymbol{\beta}$ itself. This means that design matrices have to be scaled similarly across all model terms for the magnitude of the coefficients to be a good proxy for the importance of the associated effect. In **spikeSlabGAM**, each term's design matrix is scaled to have a Frobenius norm of 0.5 to achieve this.

A parameter-expanded NMIG prior

While the conventional NMIG prior (3) works well for the selection of single coefficients, it is unsuited for the simultaneous selection or deselection of coefficient vectors, such as coefficients associated with spline basis functions or with the levels of a random intercept. In a nutshell, the problem is that a small variance for a batch of coefficients implies small coefficient values and small coefficient values in turn imply a small variance so that blockwise MCMC samplers are unlikely to exit a basin of attraction around the origin. Gelman *et al.* (2008) analyze this issue in the context of hierarchical models, where it is framed as a problematically strong dependence between a block of coefficients and their associated hypervariance. A bimodal prior for the variance, such as the NMIG prior, obviously exacerbates these difficulties as the chain has to be able to switch between the different components of the mixture prior. The problem is much less acute for coefficient batches with only a single or few entries since a small batch contributes much less information to the full conditional of its variance parameter. The sampler is then better able to switch between the less clearly separated basins of attraction around the two modes corresponding to the spike and the slab (Scheipl 2010, Section 3.2). In our context, "switching modes" means that entries in $\boldsymbol{\gamma}$ change their state from 1 to v_0 or vice versa. The practical importance for our aim is clear: Without fast and reliable mixing of $\boldsymbol{\gamma}$ for coefficient batches with more than a few entries, the posterior distribution cannot be used to define marginal probabilities of models or term inclusion. In previous approaches, this problem has been circumvented by either relying on very low dimensional bases with only a handful of basis functions (Reich *et al.* 2009; Cottet *et al.* 2008) or by sampling the indicators from a partial conditional density, with coefficients and their hypervariances integrated out (Yau *et al.* 2003).

A promising strategy to reduce the dependence between coefficient batches and their variance parameter that neither limits the dimension of the base nor relies on repeated integration of multivariate functions is the introduction of working parameters that are only partially identifiable along the lines of *parameter expansion* or *marginal augmentation* (Meng and van Dyk 1997; Gelman *et al.* 2008). The central idea implemented in **spikeSlabGAM** is a multiplicative parameter expansion that improves the shrinkage properties of the resulting marginal prior compared to NMIG (Scheipl 2010, Section 3.4) and enables simultaneous selection or deselection of large coefficient batches.

Figure 1 shows the peNMIG prior hierarchy for a model with p model terms: We set $\boldsymbol{\beta}_j = \alpha_j \boldsymbol{\xi}_j$ with mutually independent α_j and $\boldsymbol{\xi}_j$ for a coefficient batch $\boldsymbol{\beta}_j$ with length d_j and use a scalar parameter $\alpha_j \overset{\text{prior}}{\sim} \text{NMIG}(v_0, w, a_\tau, b_\tau)$, where NMIG denotes the prior hierarchy given in (3). Entries of the vector $\boldsymbol{\xi}_j$ are i. i. d. $\xi_{jk} \overset{\text{prior}}{\sim} N(m_{jk}, 1)$ $(k = 1, \ldots, d_j; j = 1, \ldots, p)$ with prior means m_{jk} either 1 or -1 with equal probability.

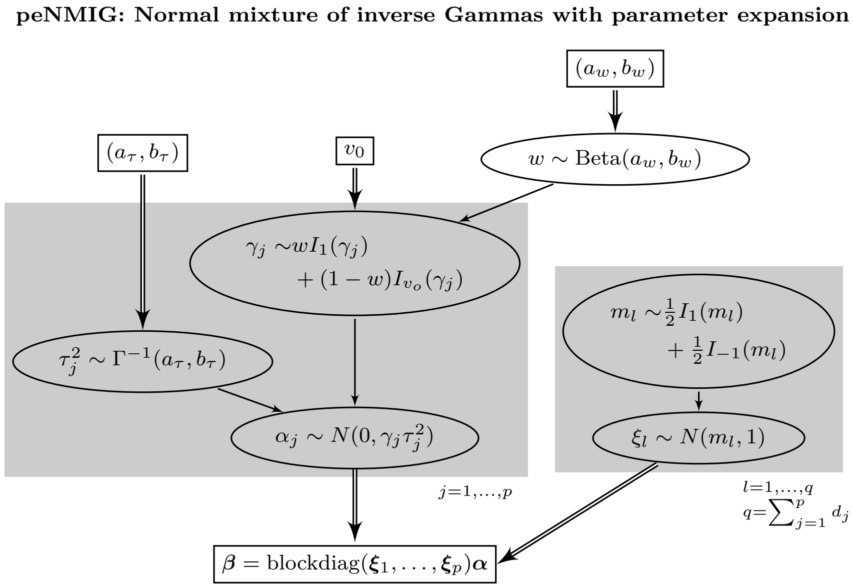
peNMIG: Normal mixture of inverse Gammas with parameter expansion
(a_w, b_w)
(a_τ, b_τ)
v_0
$w \sim \text{Beta}(a_w, b_w)$
$\gamma_j \sim wI_1(\gamma_j) + (1-w)I_{v_o}(\gamma_j)$
$m_l \sim \frac{1}{2}I_1(m_l) + \frac{1}{2}I_{-1}(m_l)$
$\tau_j^2 \sim \Gamma^{-1}(a_\tau, b_\tau)$
$\alpha_j \sim N(0, \gamma_j\tau_j^2)$
$\xi_l \sim N(m_l, 1)$
$j=1,\dots,p$
$l=1,\dots,q$
$q=\sum_{j=1}^p d_j$
$\boldsymbol{\beta} = \text{blockdiag}(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_p)\boldsymbol{\alpha}$

Figure 1: Directed acyclic graph of the peNMIG prior structure. Ellipses are stochastic nodes, rectangles are deterministic nodes. Single arrows are stochastic edges, double arrows are deterministic edges.

We write

$$\boldsymbol{\beta}_j \sim \text{peNMIG}(v_0, w, a_\tau, b_\tau) \tag{4}$$

as shorthand for this parameter expanded NMIG prior. The effective dimension of each coefficient batch associated with a specific γ_j and τ_j^2 is then just one, 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 solves the mixing problems for $\boldsymbol{\gamma}$ described above. The long vector $\boldsymbol{\xi} = (\boldsymbol{\xi}_1^\top, \dots, \boldsymbol{\xi}_p^\top)^\top$ is decomposed into subvectors $\boldsymbol{\xi}_j$ associated with the different coefficient batches and their respective entries α_j $(j = 1, \dots, p)$ in $\boldsymbol{\alpha}$. The parameter w is a global parameter that influences all model terms, it can be interpreted as the prior probability of a term being included in the model. The parameter α_j parameterizes the “importance” of the j-th model term, while $\boldsymbol{\xi}_j$ “distributes” α_j across the entries in the coefficient batch $\boldsymbol{\beta}_j$. Setting the conditional expectation $\text{E}(\xi_{jk}|m_{jk}) = \pm 1$ shrinks $|\xi_{jk}|$ towards 1, the multiplicative identity, so that the interpretation of α_j as the “importance” of the j-th coefficient batch can be maintained.

The marginal peNMIG prior, i.e., the prior for $\boldsymbol{\beta}$ integrated over the intermediate quantities α, $\boldsymbol{\xi}$, τ^2, γ and w, combines an infinite spike at zero with heavy tails. This desirable combination is similar to the properties of other recently proposed shrinkage priors such as the horseshoe prior (Carvalho *et al.* 2010) 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 (Polson and Scott 2010). The shape of the marginal peNMIG prior is fairly close to the original spike-and-slab prior suggested by Mitchell and Beauchamp (1988), which used a mixture of a point mass in zero and a uniform distribution on a finite interval, but it has the benefit of (partially) conjugate and proper priors. A detailed derivation of the properties of the peNMIG prior and an investigation of its sensitivity to hyperparameter settings is in Scheipl (2010), along with performance comparisons against **mboost** and other approaches with regard to term selection, sparsity recovery, and estimation error for Gaussian, binomial and Poisson responses on real and simulated data sets. The default settings for the hyperparameters, validated through many simulations and data examples are $a_\tau = 5, b_\tau = 25, v_0 = 2.5 \cdot 10^{-4}$. By default, we use a uniform prior on w, i.e., $a_w = b_w = 1$, and a very flat $\Gamma^{-1}(10^{-4}, 10^{-4})$ prior for the error variance in Gaussian models.

3. Implementation

3.1. Setting up the design

All of the terms implemented in **spikeSlabGAM** have the following structure: First, their contribution to the predictor $\boldsymbol{\eta}$ is represented as a linear combination of basis functions, i.e., the term associated with covariate $\boldsymbol{x}$ is represented as $\tilde{f}(\boldsymbol{x}) = \sum_{k=1}^{K} \delta_k \tilde{B}_k(\boldsymbol{x}) = \tilde{\boldsymbol{B}}\boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is a vector of coefficients associated with the basis functions $\tilde{B}_k(\cdot)$ $(k = 1, \ldots, K)$ evaluated in $\boldsymbol{x}$. Second, $\boldsymbol{\delta}$ has a (conditionally) multivariate Gaussian prior, i.e., $\boldsymbol{\delta}|v^2 \overset{\text{prior}}{\sim} N(\boldsymbol{0}, v^2\boldsymbol{P}^{-})$, with a fixed scaled precision matrix $\boldsymbol{P}$ that is often positive *semi*-definite. Table 1 gives an overview of the model terms available in **spikeSlabGAM** and how they fit into this framework.

Formula (2) glosses over the fact that every coefficient batch associated with a specific term will have some kind of prior dependency structure determined by $\boldsymbol{P}$. 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). The precision matrix for an 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 remains included in the model, but we need 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 be able to select or deselect its penalized and unpenalized parts separately. The second issue is that, since 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 with partially improper priors into two parts – one associated with the improper/unpenalized part of the prior and one associated with the proper/penalized part of the prior; and (2) absorbing the fixed prior correlation structure of the coefficients implied by $\boldsymbol{P}$ into a transformed design matrix $\boldsymbol{B}$ associated with then *a priori* independent coefficients $\boldsymbol{\beta}$ 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

R-syntax	**Description**	$\tilde{\boldsymbol{B}}$	$\boldsymbol{P}$
`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`	$\mathtt{C}^{-1}$
`sm(x)`	univariate penalized spline: defaults to cubic B-splines with 2^{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 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). Unpenalized terms not associated with a peNMIG prior (i.e., the columns in $\boldsymbol{X}_u$ in (1)) are specified with term type `u()`.

remainder of the unpenalized component enters the model in a separate term, e.g., P-splines (term type `sm()`, see Table 1) leave polynomial functions of a certain order unpenalized and these enter the model in a separate `lin()`-term.

Orthogonal decomposition The first strategy, used by default, employs a reduced rank approximation of the implied covariance of $\tilde{f}(\boldsymbol{x})$ to construct $\boldsymbol{B}$, similar to the approaches used in Reich *et al.* (2009) and Cottet *et al.* (2008):
Since $\tilde{f}(\boldsymbol{x}) = \tilde{\boldsymbol{B}}\delta \overset{\text{prior}}{\sim} N(0, v^2\tilde{\boldsymbol{B}}\boldsymbol{P}^{-}\tilde{\boldsymbol{B}}^\top)$, we can use the spectral decomposition $\tilde{\boldsymbol{B}}\boldsymbol{P}^{-}\tilde{\boldsymbol{B}}^\top = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top$ with orthonormal $\boldsymbol{U}$ and diagonal $\boldsymbol{D}$ with entries ≥ 0 to find an orthogonal basis representation for $\mathsf{COV}\left(\tilde{f}(\boldsymbol{x})\right)$. For $\tilde{\boldsymbol{B}}$ with $\tilde{d}$ columns and full column rank and $\boldsymbol{P}$ with rank $\tilde{d} - n_P$, where n_P is the dimension of the nullspace of $\boldsymbol{P}$, all eigenvalues of $\mathsf{COV}\left(\tilde{f}(\boldsymbol{x})\right)$ except the first $\tilde{d} - n_P$ are zero. Now write $\mathsf{COV}\left(\tilde{f}(\boldsymbol{x})\right) = [\boldsymbol{U}_{+}\boldsymbol{U}_0] \left[\begin{smallmatrix} \boldsymbol{D}_{+} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{smallmatrix}\right] [\boldsymbol{U}_{+}\boldsymbol{U}_0]^\top$, 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 $\boldsymbol{B} = \boldsymbol{U}_{+}\boldsymbol{D}_{+}^{1/2}$ and $\boldsymbol{\beta} \overset{\text{prior}}{\sim} N(0, v^2\boldsymbol{I})$, $f(\boldsymbol{x}) = \boldsymbol{B}\boldsymbol{\beta}$ has a proper Gaussian distribution that is proportional to that of the partially improper prior of $\tilde{f}(\boldsymbol{x})$ (Rue and Held 2005, eq. (3.16)) and parameterizes only the penalized/proper part of $\tilde{f}(\boldsymbol{x})$, while the unpenalized part of the function is represented by $\boldsymbol{U}_0$.

In practice, it is unnecessary and impractically slow to compute all n eigenvectors and values for a full spectral decomposition $\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top$. Only the first $\tilde{d} - n_P$ are needed for $\boldsymbol{B}$, and of those the first few typically represent most of the variability in $f(\boldsymbol{x})$. **spikeSlabGAM** makes use of a fast truncated bidiagonalization algorithm (Baglama and Reichel 2006) implemented in **irlba** (Lewis 2009) to compute only the largest $\tilde{d} - n_P$ eigenvalues of $\mathsf{COV}\left(\tilde{f}(\boldsymbol{x})\right)$ and their associated eigenvectors. Only the first 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 $\boldsymbol{B}$ with d columns. e.g., for a cubic P-spline with second order difference penalty and 20 basis functions (i.e., $\tilde{d} = 20$ columns in $\tilde{\boldsymbol{B}}$ and $n_P = 2$), $\boldsymbol{B}$ will typically have only 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: $\boldsymbol{\delta} = \boldsymbol{X}_u\boldsymbol{\beta}_u + \boldsymbol{X}_p\boldsymbol{\beta}$, 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 $\boldsymbol{P} = [\boldsymbol{\Lambda}_{+}\boldsymbol{\Lambda}_0] \left[\begin{smallmatrix} \boldsymbol{\Gamma}_{+} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{smallmatrix}\right] [\boldsymbol{\Lambda}_{+}\boldsymbol{\Lambda}_0]^\top$, 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 $\tilde{\boldsymbol{B}}\boldsymbol{\delta} = \tilde{\boldsymbol{B}}(\boldsymbol{X_u}\boldsymbol{\beta_u} + \boldsymbol{X_p}\boldsymbol{\beta}) = \boldsymbol{B_u}\boldsymbol{\beta_u} + \boldsymbol{B}\boldsymbol{\beta}$ with $\boldsymbol{B}_u$ as the design matrix associated with the unpenalized part and $\boldsymbol{B}$ 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 $\boldsymbol{\beta} \sim N(\boldsymbol{0}, v^2\boldsymbol{I})$, while $\boldsymbol{\beta}_u$ has a flat prior (c.f. Kneib 2006, ch. 5.1).

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. 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.

“Centering” 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 as in Yau *et al.* (2003). Every $\boldsymbol{B}$ is transformed via $\boldsymbol{B} \to \boldsymbol{B}\left(\boldsymbol{I} - \boldsymbol{Z}(\boldsymbol{Z}^\top\boldsymbol{Z})^{-1}\boldsymbol{Z}^\top\right)$. 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, $\boldsymbol{Z} = [\mathbf{1}\ \boldsymbol{B}_1\ \boldsymbol{B}_2 \dots \boldsymbol{B}_d]$, where $\boldsymbol{B}_1, \dots, \boldsymbol{B}_d$ are the design matrices of 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.

3.2. Markov chain Monte Carlo implementation

spikeSlabGAM uses the blockwise Gibbs sampler summarized in Algorithm 1 for MCMC inference. The sampler cyclically updates the nodes in Figure 1. The FCD for $\boldsymbol{\alpha}$ is based on the “collapsed” design matrix $\boldsymbol{X}_\alpha = \boldsymbol{X}\,\mathrm{blockdiag}(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_p)$, while $\boldsymbol{\xi}$ is sampled based on a “rescaled” design matrix $\boldsymbol{X}_\xi = \boldsymbol{X}\,\mathrm{blockdiag}(\mathbf{1}_{d1}, \dots, \mathbf{1}_{dp})\boldsymbol{\alpha}$, where $\mathbf{1}_d$ is a $d \times 1$ vector of ones and $\boldsymbol{X} = [\boldsymbol{X}_u\ \boldsymbol{B}_1 \dots \boldsymbol{B}_p]$ is the concatenation of the designs for the different model terms (see (1)). The full conditionals for $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$ for Gaussian responses are given by

$$\begin{aligned}
&\boldsymbol{\alpha}|\cdot \sim N(\boldsymbol{\mu}_\alpha, \Sigma_\alpha) \text{ with}\\
&\boldsymbol{\Sigma}_\alpha = \left(\frac{1}{\phi}\boldsymbol{X}_\alpha^\top \boldsymbol{X}_\alpha + \mathrm{diag}\left(\boldsymbol{\gamma}\boldsymbol{\tau}^2\right)^{-1}\right)^{-1},\ \boldsymbol{\mu}_j = \frac{1}{\phi}\boldsymbol{\Sigma}_\alpha \boldsymbol{X}_\alpha^\top \boldsymbol{y}, \text{ and}\\
&\boldsymbol{\xi}|\cdot \sim N(\boldsymbol{\mu}_\xi, \boldsymbol{\Sigma}_\xi) \text{ with}\\
&\boldsymbol{\Sigma}_\xi = \left(\frac{1}{\phi}\boldsymbol{X}_\xi^\top \boldsymbol{X}_\xi + \boldsymbol{I}\right)^{-1};\ \boldsymbol{\mu}_j = \boldsymbol{\Sigma}_\xi\left(\frac{1}{\phi}\boldsymbol{X}_\xi^\top \boldsymbol{y} + \boldsymbol{m}\right).
\end{aligned} \tag{5}$$

For non-Gaussian responses, we use penalized iteratively re-weighted least squares (P-IWLS) proposals (Lang and Brezger 2004) in a Metropolis-Hastings step to sample $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$, i.e., Gaussian proposals are drawn from the quadratic Taylor approximation of the logarithm of the intractable FCD. Because of the prohibitive computational cost for large q and p (and low acceptance rates for non-Gaussian response for high-dimensional IWLS proposals), neither $\boldsymbol{\alpha}$ nor $\boldsymbol{\xi}$ are updated all at once. Rather, both $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$ are split into b_α (b_ξ) update blocks that are updated sequentially conditional on the states of all other parameters.

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 (5))/ 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 (5))/ 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: $w^{(t)}|\cdot \sim \text{Beta}\left(a_w + \sum_j^p I_1(\gamma_j^{(t)}), b_w + \sum_j^p I_{v0}(\gamma_j^{(t)})\right)$
  if $\boldsymbol{y}$ is Gaussian then
    update $\phi^{(t)}$ from its FCD: $\phi^{(t)}|\cdot \sim \Gamma^{-1}\left(a_{\phi} + n/2, b_{\phi} + \frac{\sum_i^n (y_i - \eta_i^{(t)})^2}{2}\right)$
```

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 then computed via $\alpha_j^{(0)} = d_j^{-1}\sum_i^{d_j} |\beta_{ji}^{(0)}|$ and $\boldsymbol{\xi}_j^{(0)} = \boldsymbol{\beta}_j^{(0)}/\alpha_j^{(0)}$. Section 4 in Scheipl (2010) contains more details on the sampler.

4. Using spikeSlabGAM

4.1. 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 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. As of now, indicators $\boldsymbol{\gamma}$ are sampled 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`: $\texttt{noise1} = \texttt{sm1} + \boldsymbol{e}$; $e_i \overset{\text{i.i.d.}}{\sim} N(0, 1)$,
- $\boldsymbol{\eta} = f(\texttt{sm1}) + f(\texttt{sm2}, \texttt{f}) + 0.1 \cdot \texttt{lin1} + 0.2 \cdot \texttt{lin2} + 0.3 \cdot \texttt{lin3}$ (see Figures 4 and 5 for the shapes of the nonlinear effects $f(\texttt{sm1})$ and $f(\texttt{sm2}, \texttt{f})$),
- the response vector $\boldsymbol{y} = \boldsymbol{\eta} + \frac{\text{sd}(\boldsymbol{\eta})}{\texttt{snr}}\boldsymbol{\epsilon}$ is generated under signal-to-noise ratio $\texttt{snr} = 3$ with i.i.d. t_5-distributed errors ϵ_i $(i = 1, \dots, n)$.

```
set.seed(1312424)
n <- 200
snr <- 3

sm1 <- runif(n)
fsm1 <- dbeta(sm1, 7, 3)/2

sm2 <- runif(n, 0, 1)
f <- gl(3, n/3)
ff <- as.numeric(f)/2
fsm2f <- ff + ff * sm2 +
  ((f == 1) * -dbeta(sm2, 6, 4) + (f == 2) * dbeta(sm2, 6, 9) +
     (f == 3) * dbeta(sm2, 9, 6))/2

lin <- matrix(rnorm(n * 3), n, 3)
colnames(lin) <- paste("lin", 1:3, sep = "")

noise1 <- sm1 + rnorm(n)
noise2 <- runif(n)
noise3 <- runif(n)
noise4 <- sample(gl(4, n/4))
```

```
eta <- drop(fsm1 + fsm2f + lin %*% c(.1, .2, .3))
y <- eta + sd(eta)/snr * rt(n, df = 5)
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:

```
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[1]:

```
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 such 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 $\frac{1}{T}\sum_t^T -2l(\boldsymbol{y}|\boldsymbol{\eta}^{(t)}, \phi^{(t)})$, 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.

```
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)
## <environment: 0x55ab923f2080>
## 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] Note that you can speed up the sampler somewhat by setting `options(mc.cores = <how many processes you want>)` before calling `spikeSlabGAM`

```
##    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:           714
## Mean posterior deviance: 323
##
## Marginal posterior inclusion probabilities and term importance:
##                    P(gamma = 1)     pi dim
## u                            NA     NA   1
## lin(sm1)                  1.000  0.090   1 ***
## sm(sm1)                   1.000  0.063   8 ***
## lin(sm2)                  0.994  0.025   1 ***
## sm(sm2)                   0.997  0.020   8 ***
## fct(f)                    1.000  0.607   2 ***
## lin(lin1)                 0.766 -0.003   1  **
## sm(lin1)                  0.042  0.000   9
## lin(lin2)                 0.975  0.020   1 ***
## sm(lin2)                  0.051  0.000   9
## lin(lin3)                 1.000  0.041   1 ***
## sm(lin3)                  0.093  0.000   9
## lin(noise1)               0.037  0.001   1
## sm(noise1)                0.036  0.000   9
## lin(noise2)               0.036  0.001   1
## sm(noise2)                0.040  0.000   8
## lin(noise3)               0.025  0.000   1
## sm(noise3)                0.044  0.000   8
## fct(noise4)               0.069  0.001   3
## lin(sm1):lin(sm2)         0.025  0.000   1
## lin(sm1):sm(sm2)          0.070  0.000   7
## lin(sm1):fct(f)           0.032  0.000   2
## lin(sm1):lin(lin1)        0.028  0.001   1
## lin(sm1):sm(lin1)         0.114  0.000   7
## sm(sm1):lin(sm2)          0.082  0.000   7
## sm(sm1):sm(sm2)           0.065  0.000  27
## sm(sm1):fct(f)            0.274  0.001  13   *
## sm(sm1):lin(lin1)         0.100  0.000   7
## sm(sm1):sm(lin1)          0.091  0.000  28
## lin(sm2):fct(f)           0.999  0.044   2 ***
## lin(sm2):lin(lin1)        0.080 -0.001   1
## lin(sm2):sm(lin1)         0.063  0.000   8
## sm(sm2):fct(f)            1.000  0.092  13 ***
## sm(sm2):lin(lin1)         0.051  0.000   7
## sm(sm2):sm(lin1)          0.075  0.000  28
## fct(f):lin(lin1)          0.028  0.000   2
## fct(f):sm(lin1)           0.065  0.000  14
## *:P(gamma = 1)>.25 **:P(gamma = 1)>.5 ***:P(gamma = 1)>.9
```

```
##
## Posterior model probabilities (inclusion threshold = 0.5 ):
##                        1     2     3     4     5     6     7     8
## prob.:             0.236  0.08 0.057 0.019 0.017 0.015 0.015 0.013
## 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           x     x     x           x     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)
## lin(sm1):fct(f)
## lin(sm1):lin(lin1)
## lin(sm1):sm(lin1)                        x
## sm(sm1):lin(sm2)
## sm(sm1):sm(sm2)                                            x
## sm(sm1):fct(f)                      x                 x
## sm(sm1):lin(lin1)                              x
## 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)
## fct(f):sm(lin1)
## cumulative:        0.236 0.316 0.373 0.392 0.409 0.424 0.439 0.451
```

Figure 2: 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 correct model (column 1) has a posterior probability of about 0.236.

In most applications, the primary focus will be on the marginal posterior inclusion probabilities `P(gamma = 1)`, given along with a measure of term importance `pi` and the size of the associated coefficient batch `dim`. `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 sum of squares of the (non-constant) linear predictor (Gu 1992). Note that they can assume negative values as well for terms whose contributions to the linear predictor $\bar{\boldsymbol{\eta}}_j$ are negatively correlated with the remainder of the (non-constant) linear predictor $\bar{\boldsymbol{\eta}}_{-1} - \bar{\boldsymbol{\eta}}_j$. 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 2 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 $P(\gamma_j = 1) > .5, j = 1, \ldots, p$ 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`.

4.2. 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 shows the estimated effects for `m1`.

Plots for specific terms can be requested with the `label` argument, Figures 4 and 5 show code snippets and their output for $f($`sm1`$)$ and $f($`sm2`$,$`f`$)$. The fits are quite close to the truth despite the heavy-tailed errors and the many noise terms included in the model[2].

4.3. Assessing convergence

spikeSlabGAM uses the convergence diagnostics implemented in **R2WinBUGS** (Sturtz *et al.* 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.

[2] Full disclosure: The code used to render Figures 4 and 5 is a little more intricate than the code snippets shown here, but the additional code only affects details (font and margin sizes and the arrangement of the panels).

```r
plot(m)
```

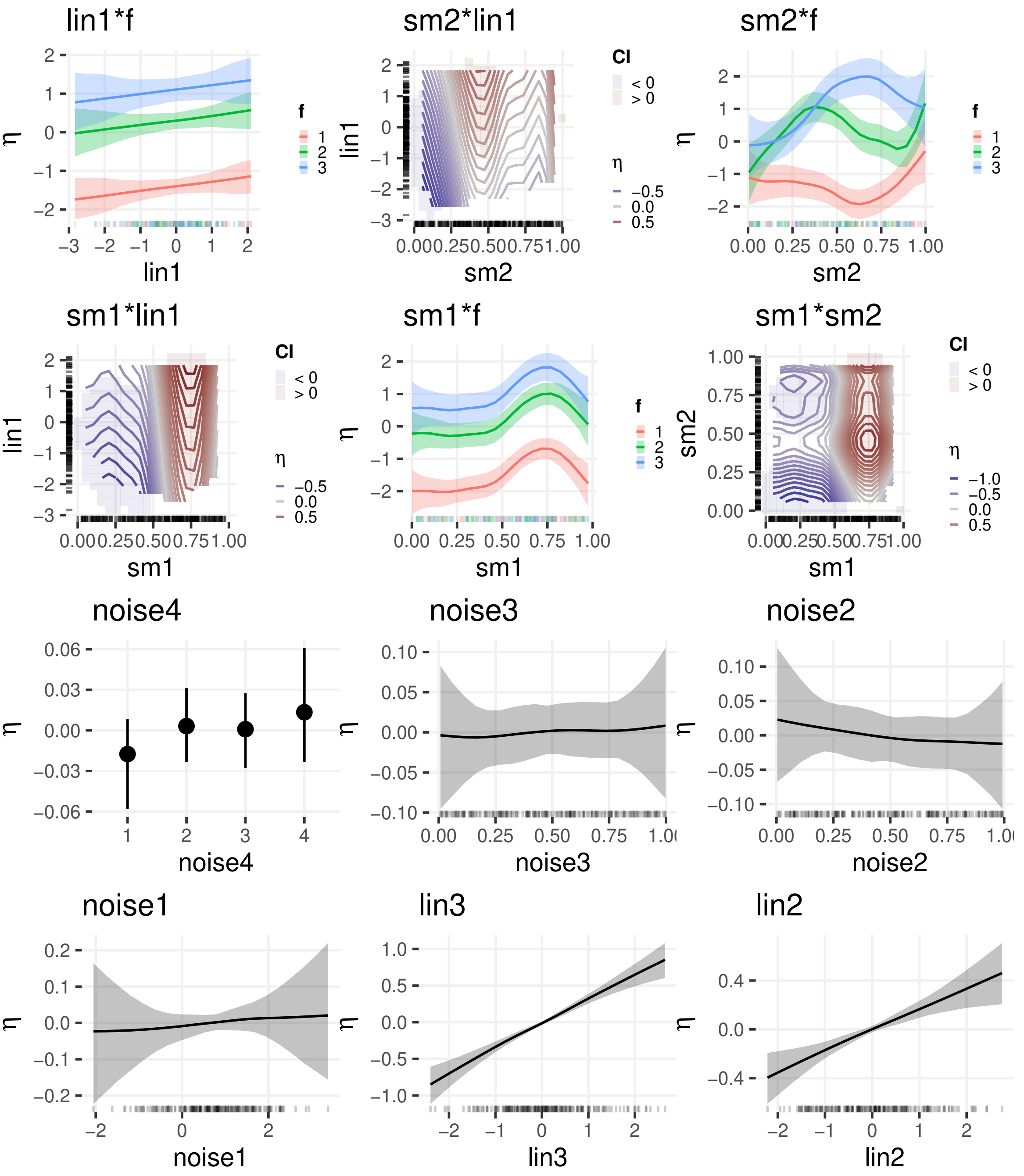

Figure 3: Posterior means and pointwise 80% credible intervals for model m. Interaction surfaces of two numerical covariates are displayed as color coded contour plots, with regions in which the credible interval does not overlap zero marked in blue ($\eta < 0$) or red ($\eta > 0$). Each panel contains 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.

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

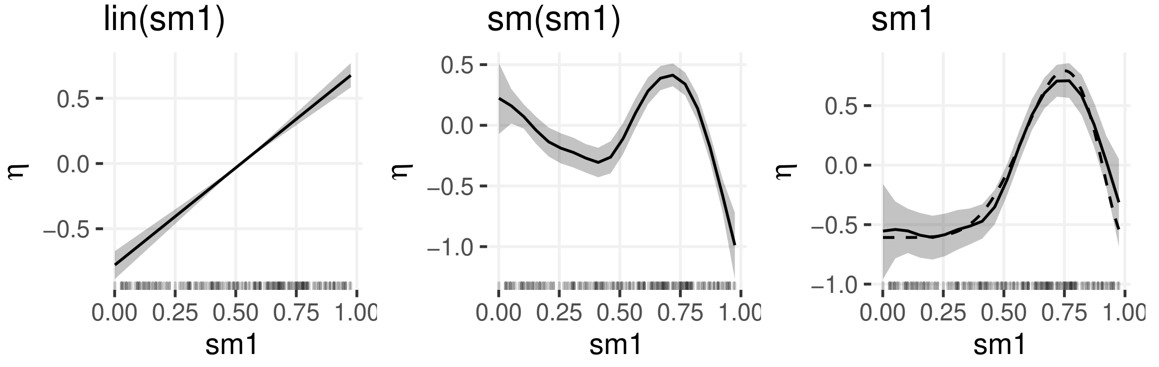

Figure 4: Posterior means and pointwise 80% credible intervals for $f($`sm1`$)$ in `m1`. Left and middle panel show the separate `lin()`- and `sm()`-terms returned by the first call to `plot`, right panel shows their sum. True shape of $f($`sm1`$)$ added as a dashed line with the `ggElems` option of `plot.spikeSlabGAM`, see lines of code above the figure.

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

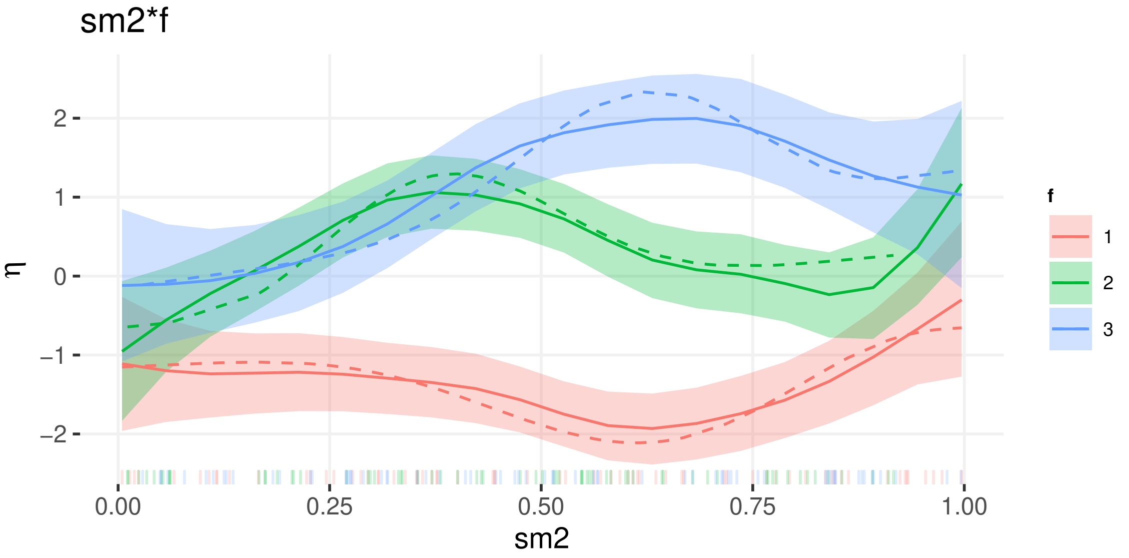

Figure 5: Posterior means and pointwise 80% credible intervals for f(`sm2`, `f`) in `m1`. True shape of f(`sm2`|`f`) added as dashed line for each level of `f`, see lines of code above the figure.

4.4. Pima indian diabetes

We use the time-honored Pima Indian Diabetes dataset as an example for real non-gaussian data: This dataset from the UCI repository (Newman *et al.* 1998) 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 covariates and the response (diabetes Yes/No) for 524 women to estimate the model. We set aside 200 observations as a test set:

```
data("PimaIndiansDiabetes2", package = "mlbench")
pimaDiab <- na.omit(PimaIndiansDiabetes2[, -c(4, 5)])
pimaDiab <- within(pimaDiab,{
  diabetes <- 1*(diabetes == "pos")
})
set.seed(1109712439)
testInd <- sample(1:nrow(pimaDiab), 200)
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 8 parallel[3] chains for an additive main effects model:

```
mcmc <- list(nChains = 8, chainLength = 1000, burnin = 500, thin = 5)
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:

```
pr0 <- predict(m0, newdata = pimaDiab[testInd,])

## Warning in (function (xnew) :  predictions outside fitted range for lin(pregnant).

## Warning in (function (xnew) :  predictions outside fitted range for sm(pregnant).

## Warning in (function (xnew) :  predictions outside fitted range for lin(pressure).

## Warning in (function (xnew) :  predictions outside fitted range for sm(pressure).

print(summary(m0), printModels = FALSE)
```

[3] on Windows, a parallel socket cluster is set up, on other OSs forking via `parallel:mclapply` is used

```
## 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))
## <environment: 0x55ab97650020>
## 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 8000 samples from 8 chain(s), each ran 5000 iterations after a
##   burn-in of 500 ; Thinning: 5
## P-IWLS acceptance rates: 0.91 for alpha; 0.63 for xi.
##
## Null deviance:           673
## Mean posterior deviance: 454
##
## Marginal posterior inclusion probabilities and term importance:
##               P(gamma = 1)     pi dim
## u                       NA     NA   1
## lin(pregnant)        0.908  0.084   1 ***
## sm(pregnant)         0.015  0.000   8
## lin(glucose)         1.000  0.513   1 ***
## sm(glucose)          0.016  0.000   9
## lin(pressure)        0.017 -0.001   1
## sm(pressure)         0.018  0.000   9
## lin(mass)            1.000  0.204   1 ***
## sm(mass)             0.729  0.063   9  **
## lin(pedigree)        1.000  0.085   1 ***
## sm(pedigree)         0.015  0.000   8
## lin(age)             0.119  0.007   1
## sm(age)              0.705  0.044   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 base-learners:

```
b <- gamboost(
  as.factor(diabetes) ~ pregnant + glucose + pressure + mass + pedigree + age,
  family = Binomial(), data = pimaDiabTrain)[300]
```

```
aic <- AIC(b, method = "classical")
prB <- predict(b[mstop(aic)], newdata = pimaDiab[testInd,])

## Warning in bsplines(mf[[i]], knots = args$knots[[i]]$knots, boundary.knots =
args$knots[[i]]$boundary.knots, :  Some 'x' values are beyond 'boundary.knots';
Linear extrapolation used.

## Warning in bsplines(mf[[i]], knots = args$knots[[i]]$knots, boundary.knots =
args$knots[[i]]$boundary.knots, :  Some 'x' values are beyond 'boundary.knots';
Linear extrapolation used.
```

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

##     bbs(mass, df = dfbase)  bbs(glucose, df = dfbase)
##                   0.347826                   0.229814
## bbs(pedigree, df = dfbase) bbs(pregnant, df = dfbase)
##                   0.130435                   0.105590
##      bbs(age, df = dfbase) bbs(pressure, df = dfbase)
##                   0.099379                   0.086957
```

Finally, we compare the deviance (i.e., twice negative log-likelihood) on the test set for the two fitted models:

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

## spikeSlabGAM     gamboost
##       199.52       204.57
```

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:

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

```
## Warning in (function (xnew) :  predictions outside fitted range for lin(pregnant).

## Warning in (function (xnew) :  predictions outside fitted range for sm(pregnant).

## Warning in (function (xnew) :  predictions outside fitted range for lin(pressure).

## Warning in (function (xnew) :  predictions outside fitted range for sm(pressure).
```

```
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))
## <environment: 0x55ab988d9720>
## 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 8000 samples from 8 chain(s), each ran 5000 iterations after a
##   burn-in of 500 ; Thinning: 5
## P-IWLS acceptance rates: 0.84 for alpha; 0.63 for xi.
##
## Null deviance:           673
## Mean posterior deviance: 446
##
## Marginal posterior inclusion probabilities and term importance:
##               P(gamma = 1)     pi dim
## u                       NA     NA   1
## lin(pregnant)        0.951  0.078   1 ***
## sm(pregnant)         0.069  0.000   8
## lin(glucose)         1.000  0.474   1 ***
## sm(glucose)          0.078  0.000   9
## lin(pressure)        0.106 -0.008   1
## sm(pressure)         0.082  0.001   9
## lin(mass)            1.000  0.220   1 ***
## sm(mass)             0.995  0.104   9 ***
## lin(pedigree)        0.987  0.079   1 ***
## sm(pedigree)         0.068  0.000   8
## lin(age)             0.140  0.012   1
## sm(age)              0.346  0.041   8   *
## *:P(gamma = 1)>.25 **:P(gamma = 1)>.5 ***:P(gamma = 1)>.9

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

## [1] 199.72
```

The selected terms are very similar, and the prediction is slightly more accurate (predictive deviance for `m0` was 199.52). Note that, due to the multi-modality of the target posterior, stable estimation of precise posterior inclusion and model probabilities requires more parallel chains than were used in this example.

5. Summary

A novel approach for Bayesian variable selection, model choice, and regularized estimation in (geo-)additive mixed models for Gaussian, binomial, and Poisson responses implemented in **spikeSlabGAM** has been described. The package uses the established R formula syntax so that complex models can be specified very concisely. It features powerful and user friendly visualizations of the fitted models. Major features of the software have been demonstrated on an example with artificial data with t-distributed errors and on the Pima Indians Diabetes data set. In future work, the author plans to add capabilities for "always included" semiparametric terms and for sampling the inclusion indicators under hierarchical constraints, i.e., never including an interaction if the associated main effects are excluded.

Computational Details

This vignette was created with:

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Linux Mint 20
## 
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
## 
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C              
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8    
##  [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=en_US.UTF-8   
##  [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C                 
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C            
## [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C       
## 
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets 
## [7] methods   base     
## 
## other attached packages:
## [1] gtable_0.3.0        ggplot2_3.3.6       mboost_2.9-5       
## [4] stabs_0.6-4         spikeSlabGAM_1.1-15
## 
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8.3       here_1.0.1         mvtnorm_1.1-3     
##  [4] lattice_0.20-45    digest_0.6.29      assertthat_0.2.1  
##  [7] rprojroot_2.0.3    utf8_1.2.2         R6_2.5.1          
## [10] plyr_1.8.7         nnls_1.4           MatrixModels_0.5-0
## [13] evaluate_0.15      coda_0.19-4        highr_0.9         
## [16] pillar_1.7.0       rlang_1.0.2        SparseM_1.81      
## [19] rpart_4.1.16       Matrix_1.4-0       partykit_1.2-15   
## [22] labeling_0.4.2     splines_4.1.0      stringr_1.4.0     
## [25] munsell_0.5.0      compiler_4.1.0     xfun_0.30         
## [28] pkgconfig_2.0.3    libcoin_1.0-9      mcmc_0.9-7        
## [31] tidyselect_1.1.2   tibble_3.1.7       gridExtra_2.3     
## [34] codetools_0.2-18   R2WinBUGS_2.1-21   quadprog_1.5-8    
## [37] reshape_0.8.9      fansi_1.0.3        crayon_1.5.1      
## [40] dplyr_1.0.9        withr_2.5.0        MASS_7.3-55       
## [43] grid_4.1.0         lifecycle_1.0.1    DBI_1.1.2
## [46] magrittr_2.0.3     scales_1.2.0       cli_3.3.0
## [49] stringi_1.7.6      farver_2.1.0       sp_1.4-6
## [52] akima_0.6-2.3      ellipsis_0.3.2     generics_0.1.2
## [55] vctrs_0.4.1        boot_1.3-28        Formula_1.2-4
## [58] tools_4.1.0        glue_1.6.2         purrr_0.3.4
## [61] survival_3.2-11    colorspace_2.0-3   cluster_2.1.2
## [64] isoband_0.2.5      inum_1.0-4         knitr_1.39
## [67] quantreg_5.88      MCMCpack_1.6-1
```

Acknowledgements

Discussions with and feedback from Thomas Kneib, Ludwig Fahrmeir and Simon N. Wood were enlightening and inspiring. Susanne Konrath shared an office with the author and his slovenly ways without complaint. Brigitte Maxa fought like a lioness to make sure the author kept getting a paycheck. An anonymous referee at JSS reinvigorated the author's faith in the peer review process with his diligent and constructive review of an earlier draft of this vignette. Financial support from the German Science Foundation (grant FA 128/5-1) is gratefully acknowledged.

References

Bae K, Mallick B (2004). “Gene Selection Using a Two-Level Hierarchical Bayesian Model.” *Bioinformatics*, **20**(18), 3423–3430.

Baglama J, Reichel L (2006). “Augmented Implicitly Restarted Lanczos Bidiagonalization Methods.” *SIAM Journal on Scientific Computing*, **27**(1), 19–42.

Brezger A, Kneib T, Lang S (2005). “BayesX: Analyzing Bayesian Structural Additive Regression Models.” *Journal of Statistical Software*, **14**(11).

Carvalho C, Polson N, Scott J (2010). “The Horseshoe Estimator for Sparse Signals.” *Biometrika*, **97**(2), 465–480.

Cottet R, Kohn R, Nott D (2008). “Variable Selection and Model Averaging in Semiparametric Overdispersed Generalized Linear Models.” *Journal of the American Statistical Association*, **103**(482), 661–671.

Fahrmeir L, Kneib T, Konrath S (2010). “Bayesian Regularisation in Structured Additive Regression: a Unifying Perspective on Shrinkage, Smoothing and Predictor Selection.” *Statistics and Computing*, **20**(2), 203–219.

Fahrmeir L, Kneib T, Lang S (2004). “Penalized Structured Additive Regression for Space-Time Data: a Bayesian Perspective.” *Statistica Sinica*, **14**, 731–761.

Frühwirth-Schnatter S, Wagner H (2010). “Bayesian Variable Selection for Random Intercept Modelling of Gaussian and Non-Gaussian Data.” In J Bernardo, M Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, M West (eds.), *Bayesian Statistics 9*. Oxford University Press.

Gelman A, Van Dyk D, Huang Z, Boscardin J (2008). “Using Redundant Parameterizations to Fit Hierarchical Models.” *Journal of Computational and Graphical Statistics*, **17**(1), 95–122.

George E, McCulloch R (1993). “Variable Selection via Gibbs Sampling.” *Journal of the American Statistical Association*, **88**(423), 881–889.

Gu C (1992). “Diagnostics for Nonparametric Regression Models with Additive Terms.” *Journal of the American Statistical Association*, **87**(420), 1051–1058.

Han C, Carlin B (2001). “Markov Chain Monte Carlo Methods for Computing Bayes Factors.” *Journal of the American Statistical Association*, **96**(455), 1122–1132.

Hothorn T, Buehlmann P, Kneib T, Schmid M, Hofner B (2010). ***mboost**: Model-Based Boosting.* R package version 2.0-7.

Ishwaran H, Rao J (2005). “Spike and Slab Variable Selection: Frequentist and Bayesian Strategies.” *The Annals of Statistics*, **33**(2), 730–773.

Kneib T (2006). *Mixed Model Based Inference in Structured Additive Regression.* Dr. Hut Verlag. URL http://edoc.ub.uni-muenchen.de/archive/00005011/.

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

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

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

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

Meng X, van Dyk D (1997). "The EM Algorithm–an Old Folk-Song Sung to a Fast New Tune." *Journal of the Royal Statistical Society B*, **59**(3), 511–567.

Mitchell T, Beauchamp J (1988). "Bayesian Variable Selection in Linear Regression." *Journal of the American Statistical Association*, **83**(404), 1023–1032.

Newman D, Hettich S, Blake C, Merz C (1998). "UCI Repository of machine learning databases." URL http://www.ics.uci.edu/~mlearn/MLRepository.html.

Polson N, Scott J (2010). "Shrink Globally, Act Locally: Sparse Bayesian Regularization and Prediction." In J Bernardo, M Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, M West (eds.), *Bayesian Statistics 9*. Oxford University Press.

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

Reich B, Storlie C, Bondell H (2009). "Variable Selection in Bayesian Smoothing Spline ANOVA Models: Application to Deterministic Computer Codes." *Technometrics*, **51**(2), 110.

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

Scheipl F (2010). "Normal-Mixture-of-Inverse-Gamma Priors for Bayesian Regularization and Model Selection in Generalized Additive Models." *Technical Report 84*, Department of Statistics, LMU Muenchen. URL http://epub.ub.uni-muenchen.de/11785/.

Sturtz S, Ligges U, Gelman A (2005). "**R2WinBUGS**: A Package for Running WinBUGS from R." *Journal of Statistical Software*, **12**(3), 1–16.

Wahba G, Wang Y, Gu C, Klein R, Klein B (1995). "Smoothing Spline ANOVA for Exponential Families, with Application to the Wisconsin Epidemiological Study of Diabetic Retinopathy." *The Annals of Statistics*, **23**(6), 1865–1895.

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

Wood S (2006). *Generalized Additive Models: an Introduction with R*. CRC Press.

Wood S, Kohn R, Shively T, Jiang W (2002). "Model Selection in Spline Nonparametric Regression." *Journal of the Royal Statistical Society B*, **64**(1), 119–139.

Yau P, Kohn R, Wood S (2003). "Bayesian Variable Selection and Model Averaging in High-Dimensional Multinomial Nonparametric Regression." *Journal of Computational and Graphical Statistics*, **12**(1), 23–54.

Affiliation:

Fabian Scheipl
Institut für Statistik

Ludwig-Maximilians-Universität München
Ludwigstr. 33
80539 München, Germany
E-mail: Fabian.Scheipl@stat.uni-muenchen.de
URL: http:/www.stat.uni-muenchen.de/~scheipl/