Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: https://www.tandfonline.com/loi/ucgs20

BAMLSS: Bayesian Additive Models for Location, Scale, and Shape (and Beyond)

Nikolaus Umlauf, Nadja Klein & Achim Zeileis

To cite this article: Nikolaus Umlauf, Nadja Klein & Achim Zeileis (2018) BAMLSS: Bayesian Additive Models for Location, Scale, and Shape (and Beyond), Journal of Computational and Graphical Statistics, 27:3, 612-627, DOI: 10.1080/10618600.2017.1407325

To link to this article: https://doi.org/10.1080/10618600.2017.1407325

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2018, VOL. 27, NO. 3, 612–627
https://doi.org/10.1080/10618600.2017.1407325

Taylor & Francis
Taylor & Francis Group

BAMLSS: Bayesian Additive Models for Location, Scale, and Shape (and Beyond)

Nikolaus Umlauf[a], Nadja Klein[b], and Achim Zeileis[a]

[a]Department of Statistics, Universität Innsbruck, Innsbruck, Austria; [b]University of Melbourne, Melbourne Business School, Melbourne, Australia

ABSTRACT

Bayesian analysis provides a convenient setting for the estimation of complex generalized additive regression models (GAMs). Since computational power has tremendously increased in the past decade, it is now possible to tackle complicated inferential problems, for example, with Markov chain Monte Carlo simulation, on virtually any modern computer. This is one of the reasons why Bayesian methods have become increasingly popular, leading to a number of highly specialized and optimized estimation engines and with attention shifting from conditional mean models to probabilistic distributional models capturing location, scale, shape (and other aspects) of the response distribution. To embed many different approaches suggested in literature and software, a unified modeling architecture for distributional GAMs is established that exploits distributions, estimation techniques (posterior mode or posterior mean), and model terms (fixed, random, smooth, spatial, …). It is shown that within this framework implementing algorithms for complex regression problems, as well as the integration of already existing software, is relatively straightforward. The usefulness is emphasized with two complex and computationally demanding application case studies: a large daily precipitation climatology, as well as a Cox model for continuous time with space-time interactions. Supplementary material for this article is available online.

ARTICLE HISTORY
Received February 2017
Revised October 2017

KEYWORDS
BUGS; Distributional regression; GAMLSS; MCMC; R; Software

1. Introduction

The generalized additive model for location, scale, and shape (GAMLSS, Rigby and Stasinopoulos 2005) relaxes the distributional assumptions of a response variable to allow for modeling the mean (location) as well as higher moments (scale and shape) in terms of covariates. This is especially useful in cases where, for example, the response does not follow the exponential family or particular interest lies on scale and shape parameters. Moreover, covariate effects can have flexible forms such as, for example, linear, nonlinear, spatial, or random effects. Hence, each parameter of the distribution is linked to an additive predictor in similar fashion as for the well-established generalized additive model (GAM, Hastie and Tibshirani 1990).

The terms of an additive predictor are most commonly represented by basis function approaches. This leads to a very generic model structure and can be further exploited because each term can be transformed into a mixed model representation (Ruppert, Wand, and Carrol 2003). In a fully Bayesian setting, this generality remains because priors on parameters can also be formalized in a general way, for example, by assigning normal priors to the regression coefficients of smooth terms (Fahrmeir, Kneib, and Lang 2004; Fahrmeir et al. 2013).

The fully Bayesian approach using Markov chain Monte Carlo (MCMC) simulation techniques is particularly attractive since the inferential framework provides valid credible intervals for estimators in situations where confidence intervals for corresponding maximum likelihood estimators based on asymptotic properties fail. This is specifically the case in more complex GAMLSS models (Klein, Kneib, and Lang 2015). In addition, extensions such as variable selection, nonstandard priors for hyperparameters, or multilevel models are easily included. Due to this and due to the tremendous increase in computational power over the past decade, the number of both, Bayesian and frequentist, estimation engines for such complicated inferential problems has been receiving increasing attention. Existing estimation engines already provide infrastructures for a number of regression problems exceeding univariate responses, for example, for multinomial, multivariate normal, censored, truncated response variables, etc.

However, many engines use different model setups and output formats, making it difficult to compare properties of different algorithms or to select the appropriate distribution and variables, etc. The reasons are manifold: the use of different model specification languages like BUGS (Lunn et al. 2009) or R (R Core Team 2016); different standalone statistical software packages like BayesX (Umlauf et al. 2015; Belitz et al. 2017), JAGS (Plummer 2003), Stan (Carpenter et al. 2017), WinBUGS (Lunn et al. 2000), etc.

In this article, we present a unified conceptional "Lego toolbox" for complex regression models. We show that iterative estimation algorithms, for example, for posterior mode or mean estimation based on MCMC simulation, exhibit very similar structures such that the process of model building becomes relatively straightforward, since the model architecture is only a

CONTACT Nikolaus Umlauf nikolaus.umlauf@uibk.ac.at Department of Statistics, Universität Innsbruck, 6020 Innsbruck, Austria.
Color versions of one or more of the figures in the article are available online at www.tandfonline.com/r/JCGS.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

combination of single "bricks." Due to many parallels to the GAMLSS class, the conceptional framework is called BAMLSS (*Bayesian additive models for location, scale, and shape*). However, it also encompasses many more general model terms *beyond* linear combinations in a design matrix with regression coefficients. The toolbox can be exploited in three ways: First, to quickly develop new models and algorithms. Second, to compare existing algorithms and samplers. Third, to easily integrate existing implementations. A proof of concept is given in the corresponding R package bamlss (Umlauf et al. 2017).

The remainder of the article is structured as follows. In Section 2, the models supported by this framework are briefly introduced. Section 3 presents the conceptional algorithm used to estimate numerous (complex) models along with the corresponding "Lego bricks" in Section 4. Section 5 describes computational strategies for implementation before Section 6 illustrates the concept using two complex and computationally demanding illustrations: a large climatology model for daily precipitation observations using censored heteroscedastic regression and a Cox model for continuous time with space–time interactions.

2. Model Structure

Based on data for $i = 1, \ldots, n$ observations, the models discussed in this article assume conditional independence of individual response observations given covariates. As in the classes of GAMLSS (Rigby and Stasinopoulos 2005) or distributional regression models (Klein et al. 2015b) all parameters of the response distribution can be modeled by explanatory variables such that

$$y \sim \mathcal{D}\left(h_1(\theta_1) = \eta_1,\ h_2(\theta_2) = \eta_2, \ldots,\ h_K(\theta_K) = \eta_K\right),$$

where $\mathcal{D}$ denotes a parametric distribution for the response variable y with K parameters θ_k, $k = 1, \ldots, K$, that are linked to additive predictors using known monotonic and twice differentiable functions $h_k(\cdot)$. Note that the response may also be a q-dimensional vector $\mathbf{y} = (y_1, \ldots, y_q)^\top$, for example, when $\mathcal{D}$ is a multivariate distribution (see, e.g., Klein et al. 2015a). The additive predictor for the kth parameter is given by

$$\boldsymbol{\eta}_k = \eta_k(\mathbf{X}; \boldsymbol{\beta}_k) = f_{1k}(\mathbf{X}; \boldsymbol{\beta}_{1k}) + \cdots + f_{J_k k}(\mathbf{X}; \boldsymbol{\beta}_{J_k k}), \quad (1)$$

based on $j = 1, \ldots, J_k$ unspecified (possibly nonlinear) functions $f_{jk}(\cdot)$, applied to each row of the generic data matrix $\mathbf{X}$, encompassing all available covariate information. The corresponding parameters $\boldsymbol{\beta}_k = (\boldsymbol{\beta}_{1k}, \ldots, \boldsymbol{\beta}_{J_k k})^\top$ are typically regression coefficients pertaining to model matrices $\mathbf{X}_k = (\mathbf{X}_{1k}, \ldots, \mathbf{X}_{J_k k})^\top$, whose structure only depend on the type of covariate(s) and prior assumptions about $f_{jk}(\cdot)$. For notational convenience, the vector of function evaluations (across all observations $i = 1, \ldots, n$) is also denoted by

$$\mathbf{f}_{jk} = f_{jk}(\mathbf{X}_{jk}; \boldsymbol{\beta}_{jk}) = (f_{jk}(\mathbf{x}_1; \boldsymbol{\beta}_{jk}), \ldots, f_{jk}(\mathbf{x}_n; \boldsymbol{\beta}_{jk}))^\top, \quad (2)$$

where $\mathbf{X}_{jk}$ $(n \times m_{jk})$ is the design matrix of the jth term for the kth parameter. In the case where it is derived through a basis function approach, it can be written as $\mathbf{f}_{jk} = \mathbf{X}_{jk}\boldsymbol{\beta}_{jk}$.

While functions $f_{jk}(\cdot)$ are usually based on a basis function approach, where η_k then is a typical GAM-type or so-called structured additive predictor (STAR, Fahrmeir, Kneib, and Lang 2004), in this article we relax this assumption and let $f_{jk}(\cdot)$ be an unspecified composition of covariate data and regression coefficients. A simple example for an $f_{jk}(\cdot)$ that is nonlinear in the parameters $\boldsymbol{\beta}_{jk}$ would be a Gompertz growth curve $\mathbf{f}_{jk} = \beta_1 \cdot \exp(-\exp(\beta_2 + \mathbf{X}_{jk}\beta_3))$.

Note that using basis functions the individual model components $\mathbf{X}_{jk}\boldsymbol{\beta}_{jk}$ can be further decomposed into a mixed model representation given by $\mathbf{f}_{jk} = \tilde{\mathbf{X}}_{jk}\tilde{\boldsymbol{\gamma}}_{jk} + \mathbf{U}_{jk}\tilde{\boldsymbol{\beta}}_{jk}$, where $\tilde{\boldsymbol{\gamma}}_{jk}$ represents the fixed effects parameters and $\tilde{\boldsymbol{\beta}}_{jk} \sim \mathcal{N}(\mathbf{0}, \tau_{jk}^2\mathbf{I})$ iid random effects. The matrix $\mathbf{U}_{jk}$ is derived from a spectral decomposition of the penalty matrix $\mathbf{K}_{jk}$ and $\tilde{\mathbf{X}}_{jk}$ by finding a basis of the null space of $\mathbf{K}_{jk}$ such that $\tilde{\mathbf{X}}_{jk}^\top\mathbf{K}_{jk} = \mathbf{0}$, that is, parameters $\tilde{\boldsymbol{\gamma}}_{jk}$ are not penalized (see, e.g., Ruppert, Wand, and Carroll 2003; Wood 2004; Fahrmeir et al. 2013). Such transformations can be used to estimate functions $f_{jk}(\cdot)$ using standard algorithms for random effects (see, e.g., Wood 2016).

3. A Conceptional Lego Toolbox

3.1. Response and Posterior Distribution

The main building block of regression model algorithms is the probability density function $d_y(\mathbf{y}|\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$, or for computational reasons its logarithm. Estimation typically requires to evaluate the log-likelihood function

$$\ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X}) = \sum_{i=1}^{n} \log\, d_y(y_i; \theta_{i1} = h_1^{-1}(\eta_1(\mathbf{x}_i; \boldsymbol{\beta}_1)), \ldots, \theta_{iK} = h_K^{-1}(\eta_K(\mathbf{x}_i; \boldsymbol{\beta}_K)))$$

a number of times, where the vector $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \ldots, \boldsymbol{\beta}_K^\top)^\top$ comprises all regression coefficients/parameters that should be estimated, $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_K)$ are the respective covariate matrices whose ith row is denoted by $\mathbf{x}_i$ and $\boldsymbol{\theta}_k$ are distribution parameter vectors of length n. Assigning prior distributions $p_{jk}(\cdot)$ to the individual components yields the log-posterior

$$\log\, \pi(\boldsymbol{\beta}, \boldsymbol{\tau}; \mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}) \propto \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X}) + \sum_{k=1}^{K}\sum_{j=1}^{J_k}\left[\log\, p_{jk}(\boldsymbol{\beta}_{jk}; \boldsymbol{\tau}_{jk}, \boldsymbol{\alpha}_{jk})\right], \quad (3)$$

where $\boldsymbol{\tau} = (\boldsymbol{\tau}_1^\top, \ldots, \boldsymbol{\tau}_K^\top)^\top = (\boldsymbol{\tau}_{11}^\top, \ldots, \boldsymbol{\tau}_{J_1 1}^\top, \ldots, \boldsymbol{\tau}_{1K}^\top, \ldots, \boldsymbol{\tau}_{J_K K}^\top)^\top$ is the vector of all assigned hyperparameters used within prior functions $p_{jk}(\cdot)$ and similarly $\boldsymbol{\alpha}$ is the set of all fixed prior specifications. More precisely, the rather general prior for the jth model term of the kth parameter is given by

$$p_{jk}(\boldsymbol{\beta}_{jk}; \boldsymbol{\tau}_{jk}, \boldsymbol{\alpha}_{jk}) \propto d_{\boldsymbol{\beta}_{jk}}(\boldsymbol{\beta}_{jk}|\, \boldsymbol{\tau}_{jk}; \boldsymbol{\alpha}_{\boldsymbol{\beta}_{jk}}) \cdot d_{\boldsymbol{\tau}_{jk}}(\boldsymbol{\tau}_{jk}|\, \boldsymbol{\alpha}_{\boldsymbol{\tau}_{jk}}), \quad (4)$$

with prior densities (or combinations of densities) $d_{\boldsymbol{\beta}_{jk}}(\cdot)$ and $d_{\boldsymbol{\tau}_{jk}}(\cdot)$ that depend on the type of covariate and prior assumptions about $f_{jk}(\cdot)$. In this framework, $\boldsymbol{\tau}_{jk}$ are typically variances, for example, that account for the degree of smoothness of $f_{jk}(\cdot)$ or the amount of correlation between observations. For example, using a spline representation of $f_{jk}(\cdot)$ in combination with a normal prior for $d_{\boldsymbol{\beta}_{jk}}(\cdot)$, the variances can be interpreted as the inverse smoothing parameters in a penalized regression context, that is, from a frequentist perspective (3) can be viewed as a penalized log-likelihood. In addition, the fixed prior specifications $\boldsymbol{\alpha}_{jk} = \{\boldsymbol{\alpha}_{\boldsymbol{\beta}_{jk}}, \boldsymbol{\alpha}_{\boldsymbol{\tau}_{jk}}\}$ can further control the shape of $d_{\boldsymbol{\beta}_{jk}}(\cdot)$

and $d_{\boldsymbol{\tau}_{jk}}(\cdot)$, incorporate prior beliefs about $\boldsymbol{\beta}_{jk}$, or for GAM-type models $\boldsymbol{\alpha}_{jk}$ usually holds the so-called penalty matrices, among others.

3.2. Model Fitting

Bayesian point estimates of $\boldsymbol{\beta}$ and $\boldsymbol{\tau}$ are typically obtained by either one of:

E1. Maximization of the log-posterior for posterior mode estimation.

E2. Solving high-dimensional integrals, for example, for posterior mean or median estimation.

For the possibly complex models within the BAMLSS framework, E1 and E2 are commonly solved by computer-intensive iterative algorithms, since analytical solutions are available only in a few special cases. In either case, the algorithms perform an updating of type

$$(\boldsymbol{\beta}^{(t+1)}, \boldsymbol{\tau}^{(t+1)}) = U(\boldsymbol{\beta}^{(t)}, \boldsymbol{\tau}^{(t)}; \mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}), \tag{5}$$

where function $U(\cdot)$ is an updating function, for example, for generating one Newton–Raphson step in E1 or getting the next step in an MCMC simulation in E2, among others. The updating scheme can be partitioned into separate updating equations using leapfrog or zigzag iteration (see, e.g., Smyth 1996). Now let

$$\begin{aligned}
&\left(\boldsymbol{\beta}_1^{(t+1)}, \boldsymbol{\tau}_1^{(t+1)}\right) \\
&\quad = U_1\left(\boldsymbol{\beta}_1^{(t)}, \boldsymbol{\beta}_2^{(t)}, \ldots, \boldsymbol{\beta}_K^{(t)}, \boldsymbol{\tau}_1^{(t)}, \boldsymbol{\tau}_2^{(t)}, \ldots, \boldsymbol{\tau}_K^{(t)}; \mathbf{y}, \mathbf{X}_1, \boldsymbol{\alpha}_1\right) \\
&\left(\boldsymbol{\beta}_2^{(t+1)}, \boldsymbol{\tau}_2^{(t+1)}\right) \\
&\quad = U_2\left(\boldsymbol{\beta}_1^{(t+1)}, \boldsymbol{\beta}_2^{(t)}, \ldots, \boldsymbol{\beta}_K^{(t)}, \boldsymbol{\tau}_1^{(t+1)}, \boldsymbol{\tau}_2^{(t)}, \ldots, \boldsymbol{\tau}_K^{(t)}; \mathbf{y}, \mathbf{X}_2, \boldsymbol{\alpha}_2\right) \\
&\quad \vdots \\
&\left(\boldsymbol{\beta}_K^{(t+1)}, \boldsymbol{\tau}_K^{(t+1)}\right) \\
&\quad = U_K\left(\boldsymbol{\beta}_1^{(t+1)}, \boldsymbol{\beta}_2^{(t+1)}, \ldots, \boldsymbol{\beta}_K^{(t)}, \boldsymbol{\tau}_1^{(t+1)}, \boldsymbol{\tau}_2^{(t+1)}, \ldots, \boldsymbol{\tau}_K^{(t)}; \mathbf{y}, \mathbf{X}_K, \boldsymbol{\alpha}_K\right)
\end{aligned} \tag{6}$$

be a partitioned updating scheme with updating functions $U_k(\cdot)$, that is, in each iteration updates for the kth parameter are computed while holding all other parameters fixed. Furthermore, this strategy can be applied for all terms within a parameter using updating function $U_{jk}(\cdot)$ for an individual model term

$$\begin{aligned}
\left(\boldsymbol{\beta}_{jk}^{(t+1)}, \boldsymbol{\tau}_{jk}^{(t+1)}\right) &= U_{jk}\left(\boldsymbol{\beta}_{jk}^{(t)}, \boldsymbol{\tau}_{jk}^{(t)}; \cdot\right) \\
j = 1, \ldots, J_k, \quad k &= 1, \ldots, K.
\end{aligned} \tag{7}$$

The partitioned updating system allows for having different functions $U_{jk}(\cdot)$ for different model terms, for example, in problem E1 some updating functions could be based on iteratively weighted least squares (IWLS, Gamerman 1997) and some on ordinary Newton–Raphson steps (see, e.g., Section 6.2). In problem E2 using MCMC it is common to mix between several sampling methods depending on the type of model term or distribution parameter.

Using highly modular systems like (6) and (7), it is possible to develop a generic estimation algorithm for numerous possibly very complex models, as outlined in Algorithm A1. The algorithm initializes all model parameters and predictors. Then an outer iteration loops over all distributional parameters performing an inner iteration updating all model terms of the respective parameter, that is, the algorithm uses backfitting type updating schemes. In practice, for full Bayesian inference the algorithm is applied twice, that is, first computing estimates for E1 and then using these as starting values for solving E2.

Finding good starting values is especially important for complex model terms, for example, for multi-dimensional $f_{jk}(\cdot)$ with multiple smoothing variances. Therefore, we propose to estimate $\boldsymbol{\tau}_{jk}$ using a goodness-of-fit criterion within the stepwise selection from Algorithm A2a, similar to Belitz and Lang (2008). In each updating step in A1 each $\boldsymbol{\tau}_{jk} = (\tau_{1jk}, \ldots, \tau_{L_{jk}jk})^\top$ is optimized one after the other using adaptive search intervals. Hence, the optimization problem is reduced to a one-dimensional search that is relatively fast and straightforward. The algorithm does not guarantee a global minimum, however, the solution is at least "close" and serves as the starting point for full MCMC. Optimization speed can be further increased if for a given search interval only a grid of possible values for each τ_{ljk} is used.

The MCMC updating usually either accepts or rejects samples of the parameters and smoothing variances are sampled after $\boldsymbol{\beta}_{jk}$. In Algorithm A2b the general scheme is shown. Note again the general structure of sampling Algorithm A2b, that is, the proposal functions $q_{jk}(\cdot)$ generate parameter samples $\boldsymbol{\beta}_{jk}^{(t+1)}$, $\boldsymbol{\tau}_{jk}^{(t+1)}$ using (possibly) different sampling schemes like derivative-based Metropolis–Hastings and slice sampling, see Section 4.2

4. Lego Bricks

For computing parameter updates for either E1 or E2 using flexible partitioned updating systems like (6) and (7), the following "Lego bricks" are repeatedly used in Algorithm A1:

B1. The density $d_y(\mathbf{y}|\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$ and respective log-likelihood function $\ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})$,

B2. link functions $h_k(\cdot)$,

B3. model terms $f_{jk}(\cdot)$ and corresponding prior densities $p_{jk}(\boldsymbol{\beta}_{jk}; \boldsymbol{\tau}_{jk}, \boldsymbol{\alpha}_{jk})$.

Moreover, in this section we derive the details for updating Algorithms A2a and A2b, which usually require

B4. the derivatives of inverse link functions $h_k^{-1}(\cdot)$,

B5. the first-order derivatives of the predictors $\frac{\partial \boldsymbol{\eta}_k}{\partial \boldsymbol{\beta}_{jk}}$,

B6. first-order derivatives of the log-likelihood

B6a. w.r.t. regression coefficients/parameters $\frac{\partial \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\beta}_{jk}}$,

B6b. w.r.t. predictors $\frac{\partial \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\eta}_k}$,

B7. the second-order derivatives of the log-likelihood

B7a. w.r.t. regression coefficients/parameters $\frac{\partial^2 \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\beta}_{jk} \partial \boldsymbol{\beta}_{jk}^\top}$,

B7b. w.r.t. predictors $\frac{\partial^2 \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_k^\top}$,

B8. derivatives for log-priors, for example, $\frac{\partial \log p_{jk}(\boldsymbol{\beta}_{jk}; \boldsymbol{\tau}_{jk}, \boldsymbol{\alpha}_{jk})}{\partial \boldsymbol{\beta}_{jk}}$.

Computationally, this leads to a "Lego" system and extending the toolbox can be done in different directions, for example: For a new response distribution, only building block B1, and possibly B6b and B7b are necessary, since in most cases B6a and B7a can be simplified when fragmenting with the chain rule. For a new model term, B3 and B5 are needed. And for a

Algorithm A1 Generic BAMLSS model fitting algorithm.

Input: $\mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}$.
Set: Stopping criterion ε, number of iterations T, for example, $\varepsilon = 0.0001$, $T = 1000$.
Initialize: $\boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{\tau}$, for example, $\boldsymbol{\beta} = \mathbf{0}$, $\boldsymbol{\tau} = 0.001 \cdot \mathbf{1}$, $\Delta = \varepsilon + 1$, $t = 1$.
while $(\Delta > \varepsilon)$ & $(t < T)$ **do**
  Set $\mathring{\boldsymbol{\eta}} = \boldsymbol{\eta}^{(t)}$.
  for $k = 1$ to K **do**
    for $j = 1$ to J_k **do**
      Obtain new state $(\boldsymbol{\beta}_{jk}^{(t+1)}, \boldsymbol{\tau}_{jk}^{(t+1)}) \leftarrow U_{jk}(\boldsymbol{\beta}_{jk}^{(t)}, \boldsymbol{\tau}_{jk}^{(t)}; \cdot)$ using Algorithm A2a or A2b.
      Compute $\mathbf{f}_{jk}^{(t+1)} \leftarrow f_{jk}(\mathbf{X}_{jk}, \boldsymbol{\beta}_{jk}^{(t+1)})$.
      Update $\boldsymbol{\eta}_k^{(t+1)} \leftarrow \boldsymbol{\eta}_k^{(t)} - \mathbf{f}_{jk}^{(t)} + \mathbf{f}_{jk}^{(t+1)}$.
    end for
  end for
  Compute $\Delta \leftarrow$ rel.change$(\mathring{\boldsymbol{\eta}}, \boldsymbol{\eta}^{(t+1)})$.
  Increase $t \leftarrow t + 1$.
end while
Output: Posterior mode estimates $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(t)}$, $\hat{\boldsymbol{\tau}} = \boldsymbol{\tau}^{(t)}$ for E1; or MCMC samples $\boldsymbol{\beta}^{(t)}, \boldsymbol{\tau}^{(t)}, t = 1, \ldots, T$ for E2.

Algorithm A2a Posterior mode updating $U_{jk}(\cdot)$ with smoothing variance selection.

Input: $\mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\tau}^{(t)}$.
Set: Goodness-of-fit criterion C.
  for $l = 1$ to L_{jk} **do**
    Set search interval for $\tau_{ljk}^{(t+1)}$,
    for example, $\mathcal{I}_{ljk} = [\tau_{ljk}^{(t)} \cdot 10^{-1}, \tau_{ljk}^{(t)} \cdot 10]$.
    Find $\tau_{ljk}^{(t+1)} \leftarrow \underset{\tau_{ljk}^{\star} \in \mathcal{I}_{ljk}}{\arg\min}\, C(U_{jk}(\boldsymbol{\beta}_{jk}^{(t)}, \boldsymbol{\tau}_{jk}^{\star}; \cdot))$.
  end for
  Update $\boldsymbol{\beta}_{jk}^{(t+1)} \leftarrow U_{jk}(\boldsymbol{\beta}_{jk}^{(t)}, \boldsymbol{\tau}_{jk}^{(t+1)}; \cdot)$.
Output: Updates $\boldsymbol{\beta}_{jk}^{(t+1)}, \boldsymbol{\tau}_{jk}^{(t+1)}$.

Algorithm A2b MCMC updating $U_{jk}(\cdot)$.

Input: $\mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\tau}^{(t)}$.
Set: Sampling method, for example, derivative-based MCMC (see Section 4.2.).
  Sample $\boldsymbol{\beta}_{jk}^{\star} \leftarrow q_{jk}(\cdot \mid \boldsymbol{\beta}_{jk}^{(t)})$.
  Compute acceptance probability $\alpha\left(\boldsymbol{\beta}_{jk}^{\star} \mid \boldsymbol{\beta}_{jk}^{(t)}\right)$.
  if uniform draw $\mathcal{U}(0, 1) \leq \alpha\left(\boldsymbol{\beta}_{jk}^{\star} \mid \boldsymbol{\beta}_{jk}^{(t)}\right)$ **then**
    $\boldsymbol{\beta}_{jk}^{(t+1)} \leftarrow \boldsymbol{\beta}_{jk}^{\star}$
  else
    $\boldsymbol{\beta}_{jk}^{(t+1)} \leftarrow \boldsymbol{\beta}_{jk}^{(t)}$.
  end if
  Generate $\boldsymbol{\tau}_{jk}^{(t+1)}$ analogously.
Output: Next state $\boldsymbol{\beta}_{jk}^{(t+1)}, \boldsymbol{\tau}_{jk}^{(t+1)}$.

new link function, B2 and B4. Then, the new building blocks are straightforward to combine with other previously available building blocks, moreover, most parts that are used for solving E1 can also be used for E2.

The remainder of this section is as follows. Details about commonly used prior densities in GAM-type models (building block B3) are provided in the next section. In Section 4.2, we derive the general parts that are needed for updating functions in Algorithm A2a and A2b, that is, building blocks B6a, B6b, B7a, and B7b. In Section 4.3 and 4.4, we briefly discuss model choice, Bayesian inference, and prediction.

4.1. Model Terms and Priors

In the following, we summarize commonly used specifications for priors $p_{jk}(\cdot)$ used for estimating GAM-type models (building block B3). In addition, Table 1 of the supplemental material provides a more detailed overview of model terms and prior structures.

4.1.1. Linear Effects

For simple linear effects $f_{jk}(\cdot)$, a common choice for $p_{jk}(\cdot)$ is to use a noninformative (constant) flat prior. One of the simplest informative priors is a normal prior given by

$$p_{jk}(\boldsymbol{\beta}_{jk}; \boldsymbol{\tau}_{jk}, \boldsymbol{\alpha}_{jk}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{\beta}_{jk} - \mathbf{m})^{\top}\mathbf{P}_{jk}(\boldsymbol{\tau}_{jk})(\boldsymbol{\beta}_{jk} - \mathbf{m})\right),$$

where $\boldsymbol{\tau}_{jk}$ are assumed to be fixed with $d_{\boldsymbol{\tau}_{jk}}(\cdot) = 1$ and $\boldsymbol{\alpha}_{jk} = \{\boldsymbol{\alpha}_{\boldsymbol{\beta}_{jk}} = \{\mathbf{m}\}\}$ with $\mathbf{m}$ as a prior mean for $\boldsymbol{\beta}_{jk}$. The matrix $\mathbf{P}_{jk}(\boldsymbol{\tau}_{jk})$ is a fixed prior precision matrix, for example, $\mathbf{P}_{jk} = \text{diag}(\boldsymbol{\tau}_{jk})$. In a lot of applications, a vague prior specification is used with $\mathbf{m} = \mathbf{0}$ and a large precision (see, e.g., Fahrmeir et al. 2013).

4.1.2. Nonlinear Effects

If the nonlinear functions $f_{jk}(\cdot)$ in (1) are modeled using a basis function approach, the usual choice of prior $p_{jk}(\cdot)$ is based on a multivariate normal kernel for $\boldsymbol{\beta}_{jk}$ given by

$$d_{\boldsymbol{\beta}_{jk}}(\boldsymbol{\beta}_{jk} \mid \boldsymbol{\tau}_{jk}, \boldsymbol{\alpha}_{\boldsymbol{\beta}_{jk}}) \propto |\mathbf{P}_{jk}(\boldsymbol{\tau}_{jk})|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{\beta}_{jk}^{\top}\mathbf{P}_{jk}(\boldsymbol{\tau}_{jk})\boldsymbol{\beta}_{jk}\right). \tag{8}$$

Here, the precision matrix $\mathbf{P}_{jk}(\boldsymbol{\tau}_{jk})$ is derived from prespecified so-called penalty matrices $\boldsymbol{\alpha}_{\boldsymbol{\beta}_{jk}} = \{\mathbf{K}_{1jk}, \ldots, \mathbf{K}_{Ljk}\}$, for example, for tensor product smooths the precision matrix is $\mathbf{P}_{jk}(\boldsymbol{\tau}_{jk}) = \sum_{l=1}^{L_{jk}} \frac{1}{\tau_{ljk}}\mathbf{K}_{ljk}$. Note that $\mathbf{P}_{jk}(\boldsymbol{\tau}_{jk})$ is often not of full rank and therefore $d_{\boldsymbol{\beta}_{jk}}(\cdot)$ is partially improper. The variances τ_{ljk} account for the amount of smoothness (regularization) of the function and can be interpreted as the inverse smoothing parameters in the frequentist approach. A common choice for the prior for $\boldsymbol{\tau}_{jk}$ is based on inverse gamma distributions for each $\boldsymbol{\tau}_{jk} = (\tau_{1jk}, \ldots, \tau_{L_{jk}jk})^{\top}$

$$d_{\boldsymbol{\tau}_{jk}}(\boldsymbol{\tau}_{jk} \mid \boldsymbol{\alpha}_{\boldsymbol{\tau}_{jk}}) = \prod_{l=1}^{L_{jk}} \frac{b_{ljk}^{a_{ljk}}}{\Gamma(a_{ljk})}\tau_{ljk}^{-(a_{ljk}+1)} \exp(-b_{ljk}/\tau_{ljk}), \tag{9}$$

with fixed parameters $\boldsymbol{\alpha}_{\boldsymbol{\tau}_{jk}} = \{\mathbf{a}_{jk}, \mathbf{b}_{jk}\}$. Small values for $\mathbf{a}_{jk}$ and $\mathbf{b}_{jk}$ correspond to approximate flat priors for $\log(\tau_{l\,jk})$. Setting $\mathbf{b}_{jk} = \mathbf{0}$ and $\mathbf{a}_{jk} = -\mathbf{1}$ or $\mathbf{a}_{jk} = -1/2 \cdot \mathbf{1}$ yields flat priors for $\tau_{l\,jk}$ and $\tau_{l\,jk}^{0.5}$, respectively. However, the inverse gamma prior is very sensitive to the choice of $\mathbf{a}_{jk}$ and $\mathbf{b}_{jk}$ if $\tau_{l\,jk}$ is close to zero. Therefore, Gelman (2006) proposed the half-Cauchy prior which has the desirable property that for $\tau_{l\,jk} = 0$ the density is a nonzero constant, whereas the density of the inverse gamma for $\tau_{l\,jk} \to 0$ vanishes (see also Polson and Scott 2012). Another question is the actual choice of hyperparameters. A recent suggestion reducing this issue to the choice of a scale parameter that is directly related to the functions $f_{jk}(\cdot)$ (and thus much better interpretable and accessible for the user) was given by Klein and Kneib (2016) for several different hyperpriors for $\tau_{l\,jk}$, such as resulting priors from half-Cauchy, half-normal or uniform priors for $\tau_{l\,jk}^{0.5}$ or resulting penalized complexity priors (Simpson et al. 2017), so-called scale-dependent priors.

4.1.3. Multilevel Effects

In numerous applications, geographical information and spatial covariates are given at different resolutions. Whenever there is such a nested structure in the data, it is possible to model the complex (spatial) heterogeneity effects using a compound prior $\boldsymbol{\beta}_{jk} = \tilde{\boldsymbol{\eta}}_{jk}(\mathbf{x}; \tilde{\boldsymbol{\beta}}_{jk}) + \boldsymbol{\varepsilon}_{jk}$, where $\boldsymbol{\varepsilon}_{jk} \sim \mathcal{N}(\mathbf{0}, \tilde{\tau}_{jk}\mathbf{I})$ is a vector of iid Gaussian random effects and $\tilde{\boldsymbol{\eta}}_{jk}(\mathbf{x}; \tilde{\boldsymbol{\beta}}_{jk})$ represents a full predictor of nested covariates, for example, including a discrete regional spatial effect. This way, potential costly operations in updating Algorithm A2a and A2b can be avoided since the number of observations in $\tilde{\boldsymbol{\eta}}_{jk}(\mathbf{x}; \tilde{\boldsymbol{\beta}}_{jk})$ is equal to the number of coefficients in $\boldsymbol{\beta}_{jk}$, which is usually much smaller than the actual number of observations n. Moreover, the full conditionals (see also Section 4.2) for $\tilde{\boldsymbol{\beta}}_{jk}$ are Gaussian regardless of the response distribution and leads to highly efficient estimation algorithms, see Lang et al. (2014).

4.2. Model Fitting

The construction of suitable updating functions $U_{jk}(\cdot)$ for solving E1 and E2 can be carried out in many ways. (Note again that the general algorithm A1 does not restrict to a specific iterative procedure.) In the following, we describe commonly used quantities that can be used for estimation of BAMLSS. Moreover, this section highlights the "Lego" properties obtained from the gradient-based updating in E1 and E2. More precisely, we first describe posterior mode updating as used within Algorithm A2a, before we introduce several MCMC sampling schemes that can be employed in the updating Algorithm A2b.

4.2.1. Posterior Mode

The mode of the posterior distribution is the mode of the log-posterior (3) given by

$$\text{Mode}(\boldsymbol{\beta}, \boldsymbol{\tau} | \mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}) = \arg\max_{\boldsymbol{\beta}, \boldsymbol{\tau}} \log \pi(\boldsymbol{\beta}, \boldsymbol{\tau}; \mathbf{y}, \mathbf{X}, \boldsymbol{\alpha})$$

and is equivalent to the maximum likelihood estimator

$$\text{ML}(\boldsymbol{\beta} | \mathbf{y}, \mathbf{X}) = \arg\max_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})$$

when assigning flat (constant) priors to $\boldsymbol{\beta}_{jk}$ for $j = 1, \ldots, J_k$, $k = 1, \ldots, K$. For models involving shrinkage priors, for example, in GAM-type models given by (8), the posterior mode is equivalent to a penalized maximum likelihood estimator for fixed parameters $\boldsymbol{\tau}_{jk}$ and prior densities $d_{\boldsymbol{\tau}_{jk}}(\cdot) \propto$ constant. Moreover, the structure of the log-posterior (3) usually prohibits estimation of $\boldsymbol{\tau}_{jk}$ through maximization and the estimator $\hat{\boldsymbol{\tau}}_{jk}$ is commonly derived by additionally minimizing information criteria such as the Akaike (AIC) or Bayesian information criterion (BIC). See also Algorithm A2a for an adaptive stepwise approach for estimation of $\boldsymbol{\tau}_{jk}$ (see also Rigby and Stasinopoulos 2005, Appendix A.2. for a more detailed discussion on smoothing parameter estimation). In Section 4.3, we briefly discuss details on the computation of information criteria with equivalent degrees of freedom.

For developing general updating functions, we begin with describing posterior mode estimation for the case of fixed parameters $\boldsymbol{\tau}_{jk}$, because these updating functions form the basis of estimation algorithms for $\boldsymbol{\tau}_{jk}$. Estimation of $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \ldots, \boldsymbol{\beta}_K^\top)^\top$ requires solving equations $\partial(\log \pi(\boldsymbol{\beta}, \boldsymbol{\tau}; \mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}))/\partial\boldsymbol{\beta} = \mathbf{0}$. A particularly convenient updating function (5) for maximization of (3) is a Newton–Raphson type updating

$$\boldsymbol{\beta}^{(t+1)} = U(\boldsymbol{\beta}^{(t)}; \cdot) = \boldsymbol{\beta}^{(t)} - \mathbf{H}\left(\boldsymbol{\beta}^{(t)}\right)^{-1} \mathbf{s}\left(\boldsymbol{\beta}^{(t)}\right) \tag{10}$$

with the following score vector $\mathbf{s}(\boldsymbol{\beta})$ and Hessian matrix $\mathbf{H}_{ks}(\boldsymbol{\beta})$ $(k, s = 1, \ldots, K)$

$$\begin{aligned}
\mathbf{s}(\boldsymbol{\beta}) &= \frac{\partial \log \pi(\boldsymbol{\beta}, \boldsymbol{\tau}; \mathbf{y}, \mathbf{X}, \boldsymbol{\alpha})}{\partial \boldsymbol{\beta}} = \frac{\partial \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\beta}} \\
&\quad + \sum_{k=1}^{K} \sum_{j=1}^{J_k} \left[\frac{\partial \log p_{jk}(\boldsymbol{\beta}_{jk}; \boldsymbol{\tau}_{jk}, \boldsymbol{\alpha}_{jk})}{\partial \boldsymbol{\beta}} \right], \\
\mathbf{H}_{ks}(\boldsymbol{\beta}) &= \frac{\partial \mathbf{s}(\boldsymbol{\beta}_k)}{\partial \boldsymbol{\beta}_s^\top} = \frac{\partial^2 \log \pi(\boldsymbol{\beta}, \boldsymbol{\tau}; \mathbf{y}, \mathbf{X}, \boldsymbol{\alpha})}{\partial \boldsymbol{\beta}_k \partial \boldsymbol{\beta}_s^\top}.
\end{aligned}$$

By chain rule, the part of the score vector involving the derivatives of the log-likelihood for the kth parameter can be further decomposed to

$$\frac{\partial \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\beta}_k} = \frac{\partial \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\eta}_k} \frac{\partial \boldsymbol{\eta}_k}{\partial \boldsymbol{\beta}_k} = \frac{\partial \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\theta}_k} \frac{\partial \boldsymbol{\theta}_k}{\partial \boldsymbol{\eta}_k} \frac{\partial \boldsymbol{\eta}_k}{\partial \boldsymbol{\beta}_k},$$

including the derivatives of: the log-likelihood with respect to $\boldsymbol{\eta}_k$ and $\boldsymbol{\theta}_k$ (building block B6a), the inverse link functions (B4), the predictor $\boldsymbol{\eta}_k$ with respect to coefficients $\boldsymbol{\beta}_k$ (B5). Similarly, the components of $\mathbf{H}_{ks}$ including $\ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})$ (B7a) can be written as

$$\begin{aligned}
\mathbf{J}_{ks}(\boldsymbol{\beta}) &= \frac{\partial^2 \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\beta}_k \partial \boldsymbol{\beta}_s^\top} \\
&= \left(\frac{\partial \boldsymbol{\eta}_s}{\partial \boldsymbol{\beta}_s}\right)^\top \frac{\partial^2 \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_s^\top} \frac{\partial \boldsymbol{\eta}_k}{\partial \boldsymbol{\beta}_k} \underbrace{+ \frac{\partial \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\eta}_k} \frac{\partial^2 \boldsymbol{\eta}_k}{\partial^2 \boldsymbol{\beta}_k}}_{\text{if } k=s},
\end{aligned} \tag{11}$$

yielding a decomposition of building blocks B7b and B5. The second term on the right-hand side cancels out if all functions (2) can be written as a matrix product of a design matrix and coefficients, for example, when using a basis function

approach. Within the first term, the second derivatives of the log-likelihood involving the predictors can be written as

$$\frac{\partial^2 \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_s^\top} = \frac{\partial \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\theta}_k} \frac{\partial^2 \boldsymbol{\theta}_k}{\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_s^\top} + \frac{\partial^2 \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})}{\partial \boldsymbol{\theta}_k \partial \boldsymbol{\theta}_s^\top} \frac{\partial \boldsymbol{\theta}_k}{\partial \boldsymbol{\eta}_k} \frac{\partial \boldsymbol{\theta}_s}{\partial \boldsymbol{\eta}_s} \tag{12}$$

involving the second derivatives of the link functions.

Using a k-partitioned updating scheme as in (6), updating functions $U_k(\cdot)$ are given by

$$\boldsymbol{\beta}_k^{(t+1)} = U_k(\boldsymbol{\beta}_k^{(t)}, \cdot) = \boldsymbol{\beta}_k^{(t)} - \mathbf{H}_{kk}\left(\boldsymbol{\beta}_k^{(t)}\right)^{-1} \mathbf{s}\left(\boldsymbol{\beta}_k^{(t)}\right). \tag{13}$$

Assuming model terms (2) that can be written as a matrix product of a design matrix and coefficients, the Hessian matrix in (13) is given by

$$\mathbf{H}_{kk}\left(\boldsymbol{\beta}_k^{(t)}\right) = \begin{pmatrix} \mathbf{X}_{1k}^\top \mathbf{W}_{kk} \mathbf{X}_{1k} + \mathbf{G}_{1k}(\boldsymbol{\tau}_{1k}) & \cdots & \mathbf{X}_{1k}^\top \mathbf{W}_{kk} \mathbf{X}_{J_k k} \\ \vdots & \ddots & \vdots \\ \mathbf{X}_{J_k k}^\top \mathbf{W}_{kk} \mathbf{X}_{1k} & \cdots & \mathbf{X}_{J_k k}^\top \mathbf{W}_{kk} \mathbf{X}_{J_k k} + \mathbf{G}_{J_k k}(\boldsymbol{\tau}_{J_k k}) \end{pmatrix}^{(t)},$$

with diagonal weight matrix $\mathbf{W}_{kk} = -\text{diag}(\partial^2 \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X}) / \partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_k^\top)$ and matrices forming building block B8

$$\mathbf{G}_{jk}(\boldsymbol{\tau}_{jk}) = \frac{\partial^2 \log\, p_{jk}(\boldsymbol{\beta}_{jk}; \boldsymbol{\tau}_{jk}, \boldsymbol{\alpha}_{jk})}{\partial \boldsymbol{\beta}_{jk} \partial \boldsymbol{\beta}_{jk}^\top}. \tag{14}$$

Here, we want to emphasize that the influence of these prior derivatives matrices is usually controlled by $\boldsymbol{\tau}_{jk}$, however, note once again that the $\boldsymbol{\tau}_{jk}$ are held fixed for the moment and usually estimation cannot be done with maximization of the log-posterior (see also Section 4.3). Typically, using a linear basis function representation of functions $f_{jk}(\cdot)$ and priors based on multivariate normal kernels (8) matrices $\mathbf{G}_{jk}(\boldsymbol{\tau}_{jk})$ are a simple product of smoothing variances and penalty matrices, for example, with only one smoothing variance building block B8 becomes $\mathbf{G}_{jk}(\tau_{jk}) = \tau_{jk}^{-1} \mathbf{K}_{jk}$ with corresponding penalty matrix $\mathbf{K}_{jk}$.

Similarly, the score vector is

$$\mathbf{s}\left(\boldsymbol{\beta}_k^{(t)}\right) = \left(\mathbf{X}_{1k}^\top \mathbf{u}_k^{(t)} - \mathbf{G}_{1k}(\boldsymbol{\tau}_{1k}) \boldsymbol{\beta}_{1k}^{(t)}, \ \ldots, \ \mathbf{X}_{J_k k}^\top \mathbf{u}_k^{(t)} - \mathbf{G}_{J_k k}(\boldsymbol{\tau}_{J_k k}) \boldsymbol{\beta}_{J_k k}^{(t)}\right)^\top$$

and derivatives $\mathbf{u}_k = \partial \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X}) / \partial \boldsymbol{\eta}_k$. Focusing on the jth row of (13) leads to single model term updating functions $U_{jk}(\cdot)$ as presented in algorithm (7). The updates are based on an iteratively weighted least-square scheme given by

$$\begin{aligned} \boldsymbol{\beta}_{jk}^{(t+1)} &= U_{jk}\left(\boldsymbol{\beta}_{jk}^{(t)}; \cdot\right) \\ &= \left(\mathbf{X}_{jk}^\top \mathbf{W}_{kk} \mathbf{X}_{jk} + \mathbf{G}_{jk}(\boldsymbol{\tau}_{jk})\right)^{-1} \mathbf{X}_{jk}^\top \mathbf{W}_{kk} \left(\mathbf{z}_k - \boldsymbol{\eta}_{k,-j}^{(t+1)}\right) \end{aligned} \tag{15}$$

with working observations $\mathbf{z}_k = \boldsymbol{\eta}_k^{(t)} + \mathbf{W}_{kk}^{-1} \mathbf{u}_k^{(t)}$ (in the supplemental material Section 3, the detailed derivations are presented), that is, the algorithm only requires building blocks B6b, B7b, and B8. Hence, this leads to a backfitting algorithm and cycling through (15) for terms $j = 1, \ldots, J_k$ and parameters $k = 1, \ldots, K$ approximates a single Newton–Raphson step in (10), since cross derivatives are not incorporated in the updating scheme. Note that this yields the ingredients of the *RS*-algorithm developed by Rigby and Stasinopoulos (2005), Appendix B.2. The updating scheme (15) can be further generalized to $\boldsymbol{\beta}_{jk}^{(t+1)} = U_{jk}(\boldsymbol{\beta}_{jk}^{(t)}, \mathbf{z}_k - \boldsymbol{\eta}_{k,-j}^{(t+1)}; \cdot)$, that is, theoretically any updating function applied on the "partial residuals" $\mathbf{z}_k - \boldsymbol{\eta}_{k,-j}^{(t+1)}$ can be used. Note also that this is equivalent to updating function

$$\boldsymbol{\beta}_{jk}^{(t+1)} = U_{jk}(\boldsymbol{\beta}_{jk}^{(t)}; \cdot) = \boldsymbol{\beta}_{jk}^{(t)} - \left[\mathbf{J}_{kk}\left(\boldsymbol{\beta}_{jk}^{(t)}\right) + \mathbf{G}_{jk}(\boldsymbol{\tau}_{jk})\right]^{-1} \mathbf{s}\left(\boldsymbol{\beta}_{jk}^{(t)}\right), \tag{16}$$

where matrix $\mathbf{J}_{kk}(\cdot)$ is the derivative matrix given in (11), involving building blocks B6a, B7a, and B8 (see also the supplemental material Section 3 the detailed derivations).

For optimization, different strategies of the backfitting algorithm (15) can be applied. One alternative is a complete inner backfitting algorithm for each parameter k, that is, the backfitting procedure updates $\boldsymbol{\beta}_{jk}$, for $j = 1, \ldots, J_k$ until convergence, afterward updates for parameters for the next k are calculated again by a complete inner backfitting algorithm, and so forth (see also Rigby and Stasinopoulos 2005).

Note that for numerical reasons, it is oftentimes better to replace the Hessian by the expected Fisher information with weights $\mathbf{W}_{kk} = -\text{diag}(E(\partial^2 \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X}) / \partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_k^\top))$, see Klein, Kneib, and Lang (2015). Moreover, to achieve convergence, algorithms for posterior mode usually initialize the parameter vectors $\boldsymbol{\theta}_k$. Then, after one complete inner backfitting iteration, the algorithm can proceed in a full zigzag fashion or again with inner iterations. For all updating schemes, it might also be appropriate to vary the updating step length of parameter updates (half-stepping), possibly in each iteration.

4.2.2. Posterior Mean

The mean of the posterior distribution is

$$E(\boldsymbol{\beta}, \boldsymbol{\tau} | \mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}) = \int \int (\boldsymbol{\beta}, \boldsymbol{\tau})^\top \pi(\boldsymbol{\beta}, \boldsymbol{\tau}; \mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}) d(\boldsymbol{\beta}, \boldsymbol{\tau}).$$

Clearly, the problem in deriving the expectation, and other quantities like the posterior median, relies on the computation of usually high-dimensional integrals, which can be rarely solved analytically and thus need to be approximated by numerical techniques.

MCMC simulation is commonly used in such situations as it provides an extensible framework that can adapt to almost any type of problem. In the following, we summarize sampling techniques that are especially well-suited within the BAMLSS framework, that is, techniques that can be used for a highly modular and extensible system. In this context, we describe sampling functions for the updating scheme presented in (7), that is, the functions $U_{jk}(\cdot)$ now generate the next step in a Markov chain.

Note that for some models, there exist full conditionals that can be derived in closed form from the log-posterior (3). However, we especially focus on situations where this is not generally the case. MCMC samples for the regression coefficients $\boldsymbol{\beta}_{jk}$ can be derived by each of the following methods:

- *Derivative-based Metropolis–Hastings*: Probably the most important algorithm, because of its generality and ease

of implementation, is random-walk Metropolis. The sampler proceeds by drawing a candidate $\boldsymbol{\beta}_{jk}^{\star}$ from a symmetric jumping distribution $q(\boldsymbol{\beta}_{jk}^{\star}|\,\boldsymbol{\beta}_{jk}^{(t)})$ which is commonly a normal distribution $\mathcal{N}(\boldsymbol{\beta}_{jk}^{(t)}, \boldsymbol{\Sigma}_{jk})$ centered at the current iterate. However, in complex settings this sampling scheme is usually not efficient and tuning the covariance matrix $\boldsymbol{\Sigma}_{jk}$ in an adaptive phase is difficult and does not necessarily result in iid behavior of the Markov chain. Therefore, a commonly used alternative for the covariance matrix of the jumping distribution is to use the local curvature information $\boldsymbol{\Sigma}_{jk} = -(\partial^2 \pi(\boldsymbol{\vartheta}; \mathbf{y}, \mathbf{X})/\partial \boldsymbol{\beta}_{jk} \boldsymbol{\beta}_{jk}^{\top})^{-1}$, or its expectation, computed at the posterior mode estimate $\hat{\boldsymbol{\beta}}_{jk}$, requiring building blocks B7a and B8. However, fixing $\boldsymbol{\Sigma}_{jk}$ during MCMC simulation might still lead to undesired behavior of the Markov chain especially when parameter samples move into regions with low probability mass of the posterior distribution. A solution with good mixing properties is to construct approximate full conditionals $\pi(\boldsymbol{\beta}_{jk}|\cdot)$ that are based on a second-order Taylor series expansion of the log-posterior centered at the last state (Gamerman 1997; Klein, Kneib, and Lang 2015). The resulting proposal density $q(\boldsymbol{\beta}_{jk}^{\star}|\boldsymbol{\beta}_{jk}^{(t)})$ is again normal (see supplements Section 4) with precision matrix and mean given by

$$\begin{aligned}\left(\boldsymbol{\Sigma}_{jk}^{(t)}\right)^{-1} &= -\mathbf{H}_{kk}\left(\boldsymbol{\beta}_{jk}^{(t)}\right)\\ \boldsymbol{\mu}_{jk}^{(t)} &= \boldsymbol{\beta}_{jk}^{(t)} - \left[\mathbf{J}_{kk}\left(\boldsymbol{\beta}_{jk}^{(t)}\right) + \mathbf{G}_{jk}(\boldsymbol{\tau}_{jk})\right]^{-1}\mathbf{s}\left(\boldsymbol{\beta}_{jk}^{(t)}\right),\end{aligned}$$

which is equivalent to the updating function given in (16) and can again be build using blocks B7a and B8. Hence, the mean is simply one Newton or Fisher scoring iteration toward the posterior mode at the current step.
Again, assuming a basis function approach for $f_{jk}(\cdot)$ the precision and mean are

$$\begin{aligned}\left(\boldsymbol{\Sigma}_{jk}^{(t)}\right)^{-1} &= \mathbf{X}_{jk}^{\top}\mathbf{W}_{kk}\mathbf{X}_{jk} + \mathbf{G}_{jk}(\boldsymbol{\tau}_{jk})\\ \boldsymbol{\mu}_{jk}^{(t)} &= \boldsymbol{\Sigma}_{jk}^{(t)}\mathbf{X}_{jk}^{\top}\mathbf{W}_{kk}\left(\mathbf{z}_k - \boldsymbol{\eta}_{k,-j}^{(t)}\right)\end{aligned}$$

with weights $\mathbf{W}_{kk} = -\text{diag}(\partial^2 \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})/\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_k^{\top})$, or the corresponding expectation, as in posterior mode updating using building blocks B7b and B8, with working observations $\mathbf{z}_k = \boldsymbol{\eta}_k^{(t)} + \mathbf{W}_{kk}^{-1}\mathbf{u}_k^{(t)}$ (see also the supplemental material Section 4 for a detailed derivation). Note again, the computation of the mean is equivalent to a full Newton step as given in updating function (16), or Fisher scoring when using $-E(\partial^2 \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})/\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_k^{\top})$, in each iteration of the MCMC sampler using iteratively weighted least squares (IWLS). If the computation of the weights is expensive, one simple strategy is to update $\mathbf{W}_{kk}$ only after samples of all parameters of $\boldsymbol{\theta}_k$ are drawn.

- *Slice sampling*: Slice sampling (Neal 2003) is a gradient-free MCMC sampling scheme that produces samples with 100% acceptance rate. Therefore, and because of the simplicity of the algorithm, slice sampling is especially useful for automated general purpose MCMC implementations that allow for sampling from many distributions. The basic slice sampling algorithm samples univariate directly under the plot of the log-posterior (3). Updates for the ith parameter in $\boldsymbol{\beta}_{jk}$ are generated by:
  1. Sample $h \sim \mathcal{U}(0, \pi(\beta_{ijk}^{(t)}|\cdot))$.
  2. Sample $\beta_{ijk}^{(t+1)} \sim \mathcal{U}(S)$ from the horizontal slice $S = \{\beta_{ijk} : h < \pi(\beta_{ijk}|\cdot)\}$.

The full conditional $\pi(\boldsymbol{\tau}_{jk}|\cdot)$ for smoothing variances is commonly constructed using priors for $\boldsymbol{\tau}_{jk}$ that lead to known distributions, that is, simple Gibbs sampling is possible. For example, this is the case when using a basis function approach and only one smoothing variance τ_{jk} is assigned. Then, by using an inverse gamma prior (9) for τ_{jk} in combination with the normal prior (8) for $\boldsymbol{\beta}_{jk}$ the full-conditional $\pi(\tau_{jk}|\cdot)$ is again an inverse gamma distribution with $\tilde{a}_{jk} = \frac{1}{2}rk(\mathbf{K}_{jk}) + a_{jk}$, $\tilde{b}_{jk} = \frac{1}{2}(\boldsymbol{\beta}_{jk}^{\star})^{\top}\mathbf{K}_{jk}\boldsymbol{\beta}_{jk}^{\star} + b_{jk}$ and matrix $\mathbf{K}_{jk}$ is again a penalty matrix that depends on the type of model term. As mentioned in Section 4.1, other priors than the inverse gamma might be desirable. Then, Metropolis–Hastings steps also for the variances can be constructed, see Klein and Kneib (2016) for details. If a simple Gibbs sampling step cannot be derived, for example, for multidimensional tensor product splines, another strategy is to use slice sampling, since the number of smoothing variances is usually not very large, the computational burden does most of the times not exceed possible benefits.

4.3. Model Choice

4.3.1. Diagnostics

Quantile residuals defined as $\hat{r}_i = \Phi^{-1}(\mathcal{F}(y_i|\,\hat{\boldsymbol{\theta}}_i))$ with the inverse cumulative distribution function (CDF) of a standard normal distribution Φ^{-1} and $\mathcal{F}(\cdot)$ the CDF of the modeled distribution $\mathcal{D}(\cdot)$ with estimated parameters $\hat{\boldsymbol{\theta}}_i = (\hat{\theta}_{i1}, \ldots, \hat{\theta}_{iK})^{\top}$ plugged in, should at least approximately be standard normally distributed if the correct model has been specified. Resulting residuals can be assessed by quantile-quantile plots or probability integral transforms which consider $u_i = \mathcal{F}(y_i|\,\hat{\boldsymbol{\theta}}_i)$. If the estimated model is a good approximation to the true data-generating process, the u_i will then approximately follow a uniform distribution on $[0, 1]$. Graphically, histograms of the u_i can be used for this purpose.

4.3.2. Smoothing Variances with Posterior Mode

As noted in Section 4.2, depending on the structure of the priors (4) parameters $\boldsymbol{\tau}_{jk}$ cannot be estimated by maximization of the log-posterior (3), for example, this is the case for GAM-type models using normal priors (8) where $\boldsymbol{\tau}_{jk}$ represents smoothing variances.

Then, goodness-of-fit criteria like the Akaike information criterion (AIC), or the corrected AIC, as well as the Bayesian information criterion (BIC), among others, are commonly used for selecting the smoothing variances. Estimation of model complexity is based on the so-called equivalent degrees of freedom (EDF), calculated by $\text{edf}_{jk}(\boldsymbol{\tau}_{jk}) := \text{trace}[\mathbf{J}_{kk}(\boldsymbol{\beta}_{jk})(\mathbf{J}_{kk}(\boldsymbol{\beta}_{jk}) + \mathbf{G}_{jk}(\boldsymbol{\tau}_{jk}))^{-1}]$, where $\mathbf{J}_{kk}(\cdot)$ is the derivative matrix given in (11) and matrix $\mathbf{G}_{jk}(\boldsymbol{\tau}_{jk})$ is the prior derivative matrix as given in (14). The total degrees of freedom used to fit the model are then estimated by $\sum_k \sum_j \text{edf}_{jk}(\boldsymbol{\tau}_{jk})$. Note that the definition of EDF here is slightly more general and is usually defined as the trace of the

smoother matrix (see, e.g., Hastie and Tibshirani 1990) and can be applied even for more complex likelihood structures.

Instead of global optimization of smoothing variances, a fast strategy is the adaptive stepwise selection approach presented in Algorithm A2a.

4.3.3. Variable Selection with Posterior Mean

The deviance information criterion (DIC) can be used for model choice and variable selection in Bayesian inference. It is easily be computed from the MCMC output without requiring additional computational effort. If $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(T)}$ is an MCMC sample from the posterior for the complete parameter vector $\boldsymbol{\beta}$, the DIC is given by $\overline{D(\boldsymbol{\beta})} + \text{pd} = 2\overline{D(\boldsymbol{\beta})} - D(\overline{\boldsymbol{\beta}}) = \frac{2}{T}\sum D(\boldsymbol{\beta}^{(t)}) - D(\frac{1}{T}\sum \boldsymbol{\beta}^{(t)})$ where $D(\boldsymbol{\beta}) = -2 \cdot \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})$ is the model deviance and $\text{pd} = \overline{D(\boldsymbol{\beta})} - D(\overline{\boldsymbol{\beta}})$ is an effective parameter count.

4.4. Inference and Prediction

Under suitable regularity conditions inference for parameters $\boldsymbol{\beta}_{jk}$ can be based on the asymptotic normality of the posterior distribution $\boldsymbol{\beta}_{jk} \,|\, \mathbf{y} \overset{a}{\sim} \mathcal{N}(\hat{\boldsymbol{\beta}}_{jk}, \mathbf{H}(\hat{\boldsymbol{\beta}}_{jk})^{-1})$, with $\hat{\boldsymbol{\beta}}_{jk}$ as the posterior mode estimate. However, this approach is problematic since it does not take into account the uncertainty of estimated smoothing parameters. From a computational perspective, it can be difficult to derive the full Hessian information, because this might involve complex cross derivatives of the parameters and there are cases where standard numerical techniques cannot be applied (see Section 6.2).

Instead, applying fully Bayesian inference is relatively easy by direct computation of desired statistics from posterior samples. Computational costs are relatively low, since only samples for parameters $\boldsymbol{\beta}_{jk}$ and $\boldsymbol{\tau}^2_{jk}$ need to be saved (in practice about 2000–3000 are sufficient) from which inference of any combination of terms is straightforward, too.

The posterior predictive distribution is approximated similarly. Random samples for response observations given new covariate values $\mathbf{x}^\star$ are computed by drawing samples from the response distribution $y^\star \sim \mathcal{D}(h_1(\theta_1) = \eta_1(\mathbf{x}^\star; \boldsymbol{\beta}_1^{(t)}), \ldots, h_K(\theta_K) = \eta_K(\mathbf{x}^\star; \boldsymbol{\beta}_K^{(t)})$ for each posterior sample $\boldsymbol{\beta}_k^{(t)}; k = 1, \ldots, K, t = 1, \ldots, T$.

5. Strategies for Implementation

An implementation of the conceptional framework proposed in the previous sections is provided in the R package `bamlss` (Umlauf et al. 2017). In this section, we outline the strategies that have been guiding this implementation but technical and R-specific details are kept brief. Instead we focus on how the flexible conceptual framework with its "Lego bricks" can be turned into an extensible and modular computational framework that readily allows to construct estimation algorithms as well as interfaces to existing software packages such as JAGS (Plummer 2003) or BayesX (Belitz et al. 2017).

To provide a common toolbox that allows to play with the Lego bricks introduced in the previous sections, a general BAMLSS software system can be set up as shown in Figure 1. This proceeds in the following steps:

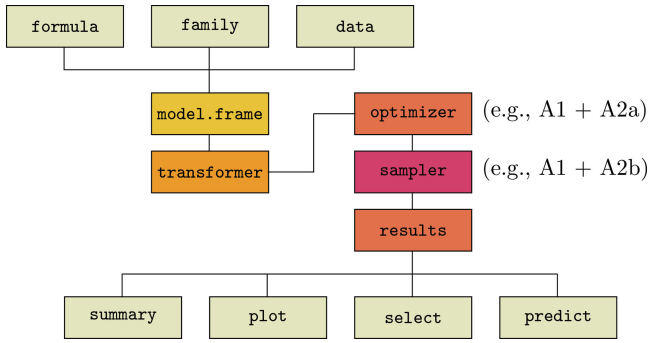

Figure 1. Flexible functional BAMLSS architecture. Each building block in the middle part can be exchanged by the user. Usually only the `optimizer` and `sampler` functions are adapted for implementing new models.

1. A unified model description where a `formula` specifies how to set up the predictors from the `data` and the `family` provides information about the Lego bricks B1–B8.
2. A generic method for setting up model terms and a `model.frame` along with the corresponding prior structures. A `transformer` can optionally set up modified terms, for example, using mixed model representation for smooth terms (see Section 2).
3. Support for modular and exchangeable updating functions or complete model fitting engines to optionally implement either E1 or E2. First, an (optional) `optimizer` function can be run, for example, for computing posterior mode estimates (E1) using Algorithm A1 and A2a. Second, a `sampler` is employed for full Bayesian inference with MCMC using Algorithm A1 in combination with A2b, which uses the posterior mode estimates from the `optimizer` as staring values. An additional step can be used for preparing the `results`.
4. Standard post-modeling extractor functions to create sampling statistics, visualizations, predictions, etc.

The items above are then essentially just collected in the main model fitting function called `bamlss()`. The most important arguments are

```
bamlss(formula, family = "gaussian",
  data = NULL, weights = NULL,
  subset = NULL, offset =
  NULL, na.action = na.omit,
  transform = NULL, optimizer = NULL,
  sampler = NULL, results = NULL,
  start = NULL, ...)
```

where the first two lines basically represent the standard model frame specifications (see Chambers and Hastie 1992).

The `formula` combines the classic Wilkinson and Rogers (1973) symbolic description—used in most standard R regression functions (Chambers and Hastie 1992)—with the infrastructure for smooth model terms like `s()`, `te()`, `ti()`, etc.—based on recommended R package mgcv—and handling multiple additive predictors—using the extended formula processing of Zeileis and Croissant (2010). Thus, a formula can be as simple as in a typical linear regression model with a response variable `y` and two regressors: `y ~ x1 + x2`. But it can also encompass smooth terms in further covariates: `y ~ x1 + x2 + s(x3) + s(x4, x5)`. Or it may even

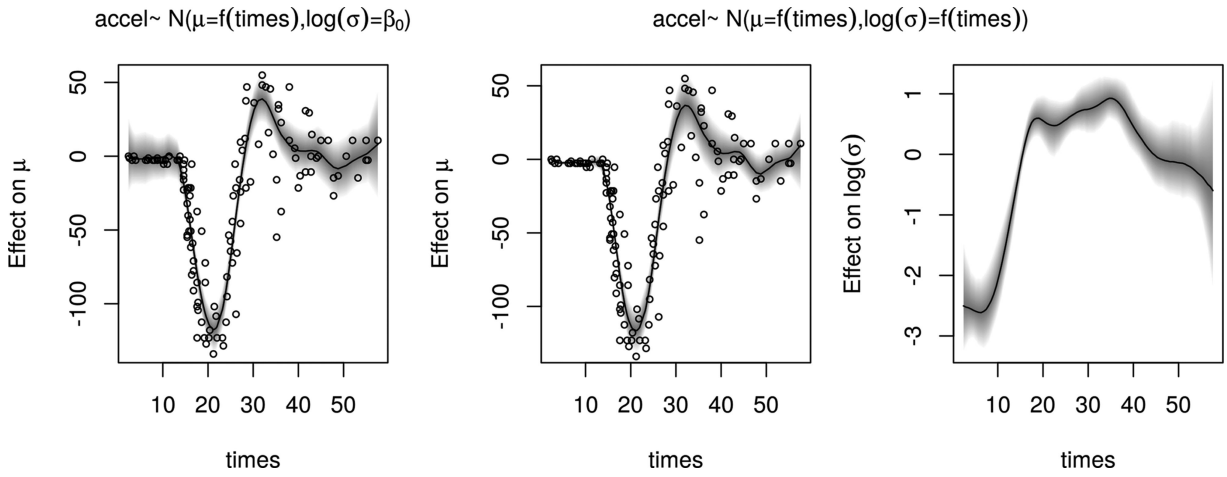

Figure 2. Estimated effects of the Gaussian location-scale model using the simulated motorcycle accident data together with 95% credible bands computed from the empirical quantiles of the MCMC samples (gray-shaded areas). The left panel shows the estimated function for parameter μ using a constant scale parameter $\log(\sigma) = \beta_0$. The middle panel shows the estimated function for parameter μ using $\log(\sigma) = f(\texttt{times})$. The right panel shows the corresponding estimated effect on $\log(\sigma)$.

have different additive predictors for different model parameters: `list(y ~ x1 + x2 + s(x3) + s(x4), sigma ~ x1 + x2 + s(x3))`, for example, in a normal model with $y \sim \mathcal{N}(\mu = \eta_\mu, \log(\sigma) = \eta_\sigma)$.

Similarly to other flexible model fitting functions, users can specify their own `family` objects to plug in different Lego bricks for B1–B8. Family objects from the gamlss suite are readily supported.

Estimation is performed by an `optimizer` and/or `sampler` function, which can be provided by the user. The default optimizer function implements the IWLS backfitting algorithm (15) with automatic smoothing variance selection, see also Algorithm A2a. The default sampler function implements derivative-based MCMC using IWLS proposals, smoothing variances are sampled using slice sampling, see also Section 4. For writing new optimizer and sampler functions, only a simple general format of function arguments and return values must be adhered to. More technical details are deferred to the documentation manual of package bamlss.

To give a brief insight how to use bamlss, we illustrate the basic steps on a small textbook example using the well-known simulated motorcycle accident data (Silverman 1985). The data contain measurements of the head acceleration (in g, variable `accel`) in a simulated motorcycle accident, recorded in milliseconds after impact (variable `times`). To estimate a Gaussian location-scale model with $\texttt{accel} \sim \mathcal{N}(\mu = f(\texttt{times}), \log(\sigma) = f(\texttt{times}))$, we use the following model formula:

```
R> f <- list(accel ~ s(times, k = 20),
+ sigma ~ s(times, k = 20))
```

where `s()` is the smooth term constructor from mgcv. The model is then fitted by

```
R> data("mcycle", package = "MASS")
R> b <- bamlss(f, data = mcycle,
+ family = "gaussian")
```

The returned object is of class `"bamlss"` for which generic extractor functions like `summary()`, `plot()`, `predict()`, etc., are provided. For example, the posterior mean function based on MCMC samples for parameter μ can be computed by

```
R> p <- predict(b, model = "mu",
+ FUN = mean)
```

where argument `FUN` can be any function, for example, a function computing credible intervals from the empirical quantiles of the MCMC samples. In Figure 2, the estimated function for a Gaussian model with constant scale $\log(\sigma) = \beta_0$ is shown in the left panel. Although the mean function seems to modeled appropriately, the plot shows that the 95% credible bands are too wide in the beginning and the end of the experiment. The middle panel shows the estimated function of the heteroscedastic model from above. Here, the credible bands follow much better the distribution of the data driven by the estimated variance function shown in the right panel. Uncertainty is low in the beginning and goes up until about 35 milliseconds and slightly decreases afterward.

6. Illustrations

6.1. Censored Heteroscedastic Precipitation Climatology

Climatology models are one important component of the meteorological tool set. The accurate and complete knowledge of precipitation climatologies is especially relevant for problems involving agriculture, risk assessment, water management, tourism, etc. One particular challenge of such models is the prediction at high temporal and spatial resolutions, especially in areas without measurement. This is usually accounted for by simple averaging/smoothing at a coarse temporal scale (e.g., monthly aggregations) combined with a second step using spatial interpolation methods like kriging (Krige 1951). However, such approaches may not work well enough at a daily resolution where precipitation data are skewed and exhibit high density at zero observations. To address these issues, Stauffer et al. (2017) had recently suggested an additive regression model for daily precipitation observations based on a censored normal response distribution and various smooth spatio-temporal effects.

Following the model of Stauffer et al. (2017) for the province of Tyrol in Austria, we take their approach a step further and establish a daily precipitation climatology for all of Austria using a large and freely available homogenized data

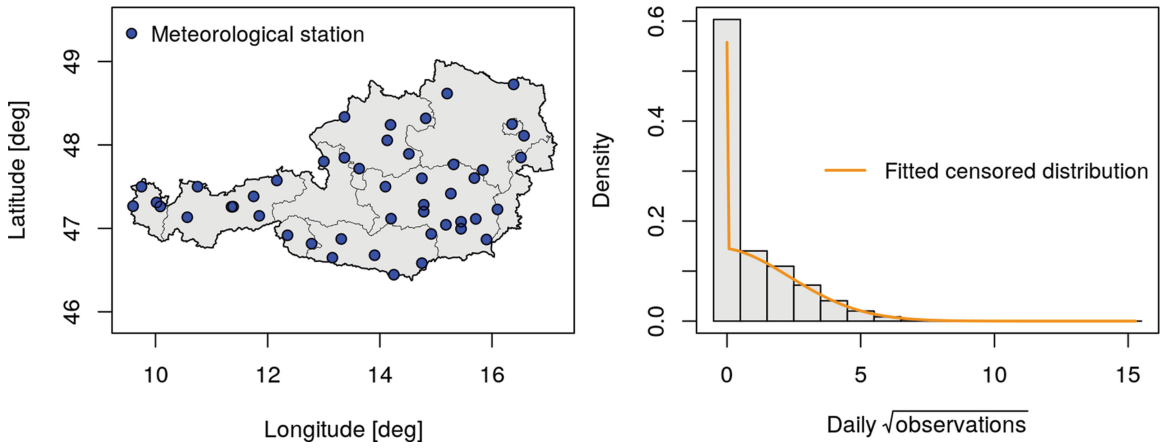

Figure 3. Distribution of available meteorological stations, left panel, and daily square root transformed precipitation values, right panel.

source. The data are taken from the HOMSTART project (*http://www.zamg.ac.at/cms/de/forschung/klima/datensaetze/homstart/*) conducted at the Zentralanstalt für Meteorologie und Geodynamik (ZAMG) and funded by the Austrian Climate Research Programme (ACRP, Nemec et al. 2011, 2013). Homogenization was successfully carried out for daily precipitation time series within 1948–2009 from a rather dense net of 57 meteorological stations (see the left panel of Figure 3). Umlauf et al. (2012) previously investigated the data based on a much simpler ordered probit model to answer the question whether it rains more frequently on weekends than during work days (it does not). Here, we reanalyze the data using a much more complex additive regression model with a normal response left-censored at zero. To make positive observations more "normal," a commonly used square-root transformation has been applied prior to regression modeling (see the right panel of Figure 3).

Specifically, the censored normal model with latent Gaussian variable $y^\star$ and observed response y, the square root of daily precipitation observations, is given by

$$y^\star \sim \mathcal{N}(\mu, \sigma^2), \;\; \mu = \eta_\mu, \;\; \log(\sigma) = \eta_\sigma, \;\; y = \max(0, y^\star).$$

Because precipitation in the Alps is driven by the season and local characteristics, for example, differing altitude form north to south, we use the following predictor for μ and σ:

$$\eta = \beta_0 + f_1(\texttt{alt}) + f_2(\texttt{day}) + f_3(\texttt{lon}, \texttt{lat}) + f_4(\texttt{day}, \texttt{lon}, \texttt{lat}),$$

here function f_1 is an altitude effect, f_2 is the cyclic seasonal variation, f_3 is a spatially correlated effect of longitude and latitude coordinates, and f_4 is a spatially varying seasonal effect. Hence, the overall seasonal effect is constructed by the main effect f_2 and the interaction effect f_4, where the deviations from the main effect are modeled to sum to zero for each day of the year, that is, this can be viewed as a functional ANOVA decomposition.

For full Bayesian estimation with Algorithm A1, A2a, and A2b, we construct updating functions $U_{jk}(\cdot)$ based on IWLS structures. Hence, as shown in Section 4 this only requires the following "Lego bricks" to be implemented (the detailed expressions are provided in the supplemental material Section 2):

B1. The density function of a Gaussian distribution left censored at zero.

B6b. Score vectors $\mathbf{u}_k = \partial \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})/\partial \boldsymbol{\eta}_k$.

B7b. The diagonal elements of the weight matrix $\mathbf{W}_{kk} = -\text{diag}(\partial^2 \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})/\partial \boldsymbol{\eta}_k \partial \boldsymbol{\eta}_k^\top)$.

The first and second derivative functions have been implemented in the bamlss family `cnorm_bamlss()`.

Since the HOMSTART dataset has over 1.2 million observations, the full storage of the resulting design matrices would lead to excessive demands concerning both computer storage as well as CPU power. To prevent computational problems associated with very large datasets like HOMSTART, we make use of the fact that the number of unique regressor observations is much smaller, for example, only 365 for the day-of-year effect. This is much smaller than the total number of observations of the dataset and duplicated rows in the corresponding design matrix can be avoided within the model fitting algorithms. Therefore, we implemented updating functions $U_{jk}(\cdot)$ that support shrinkage of the design matrices based on unique covariate observations, using the highly efficient algorithm of Lang et al. (2014). This essentially employs a reduced form of the diagonal weight matrix $\mathbf{W}_{kk}$ in the IWLS algorithm and computes the reduced partial residual vector from $\mathbf{z}_k - \boldsymbol{\eta}_{k,-j}^{(t)}$ separately. For usage within bamlss, see also the documentation of estimation engines `bfit()` and `GMCMC()` and the corresponding updating functions `bfit_iwls()` and `GMCMC_iwls()`.

With a total of 4000 iterations of the MCMC sampler, on a Linux system with 8 Intel i7-2600 3.40 GHz processors running the model takes approximately 17 hr. For each core, the first 2000 samples are withdrawn and only every 10th sample is saved.

The plots of the estimated effects are shown in Figure 4. The top row illustrates the spatial variation of the seasonal effect (solid lines) together with the mean effect (dashed lines) for parameters μ and σ. The estimates indicate that during June to August precipitation is highest in the mean effect for μ. However, there is some clear spatial variation, especially differences between the regions north and south of the Alps. This is highlighted by the red, gray, and blue lines and show that the southern stations have a clear annual peak while for the northern stations the semiannual pattern is more pronounced. Similarly, the seasonal effect for parameter σ has considerable variation between north and south. The uncertainty peak is shifting from the middle of summer to fall when going from north to south.

The second row of Figure 4 shows the resulting spatial trends. The spatial effect for parameter μ indicates that regions with

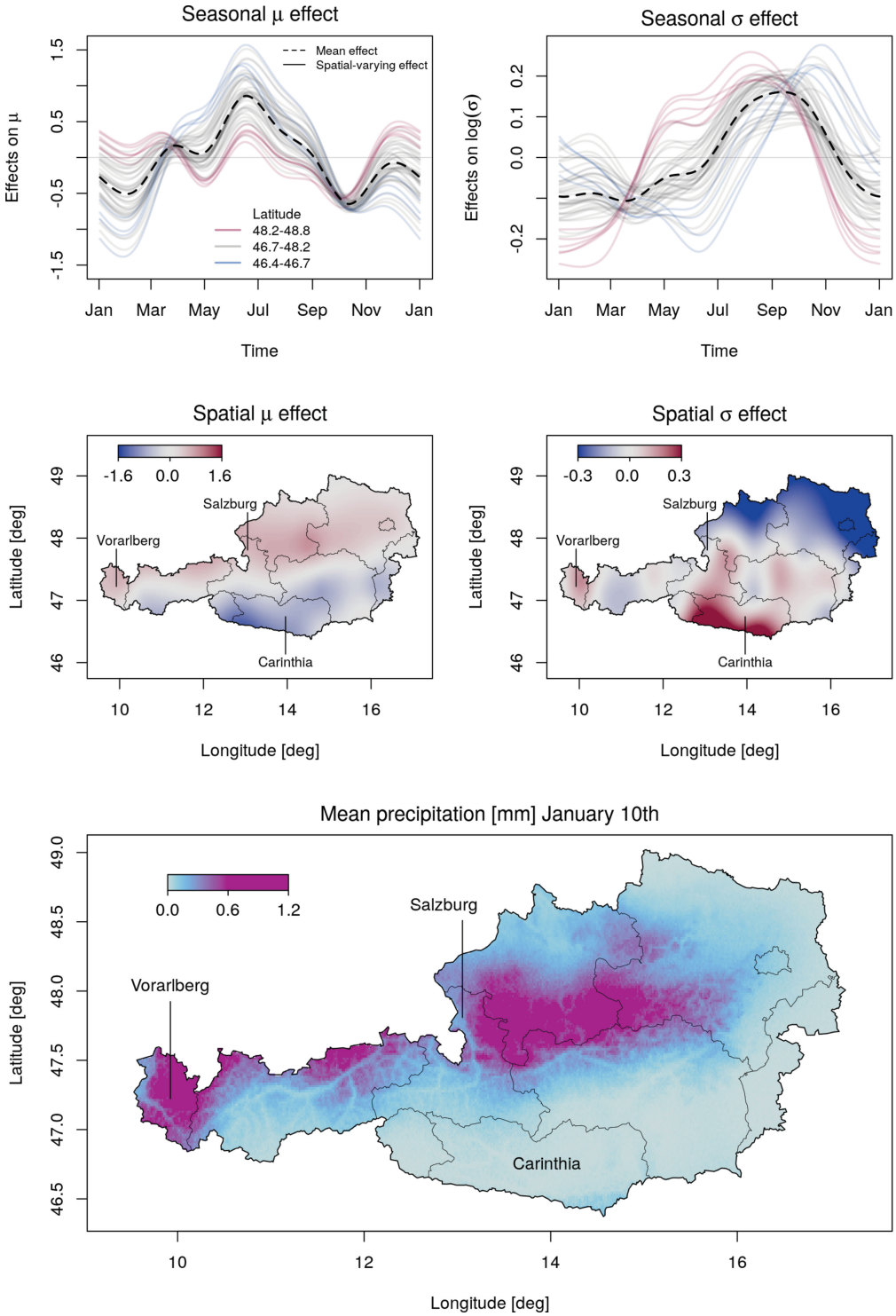

Figure 4. Estimated effects of the precipitation model showing the spatial variation of the seasonal effects together with the spatial main effects, 1st and 2nd row, predicted average precipitation of the censored mean computed using sampling from the fitted distribution for January 10, bottom row.

positive effect accumulate in the north-west part of Austria. The overall importance of the spatial effect is somewhat smaller compared to the seasonal effects, which is highlighted by using the same range for y-axes in the first row and the color legends in the second row. The spatial effect for parameter σ shows that model uncertainty is the highest within the southern regions (especially the province of Carinthia) and in the most western province (Vorarlberg).

The bottom plot in Figure 4 is an example of the resulting precipitation climatology for January 10. The predicted average precipitation is quite low all over Austria, ranging from 0 to 1.1mm. The map indicates that more precipitation can be expected in the northern parts of the Alps, especially in the west (Vorarlberg) and in the center (Salzburg). The effect of elevation is also visible since the valleys exhibit less precipitation than the alpine regions, however, the effect is not as pronounced as, for example, the seasonal effect(s), most probably because the variation of elevation of the meteorological stations used in this dataset is relatively small.

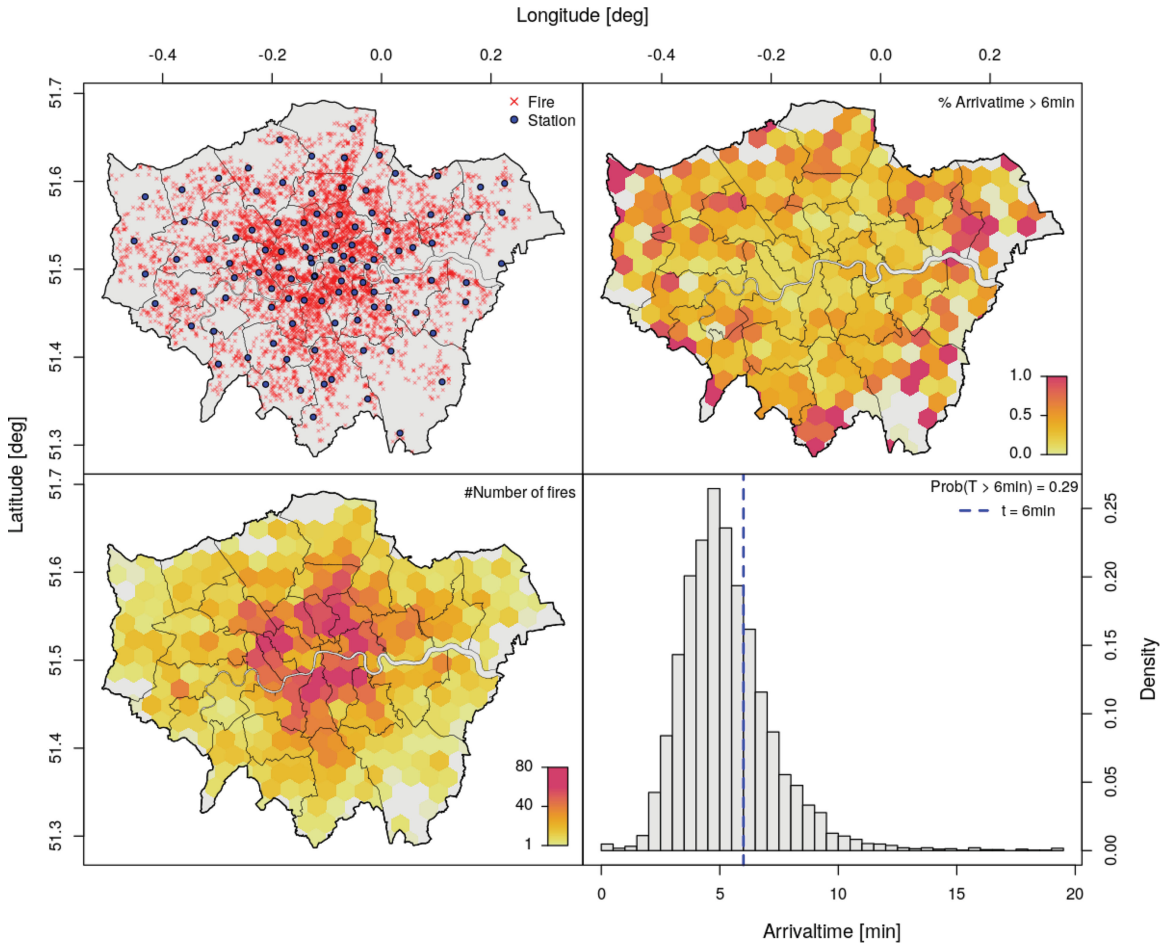

Figure 5. Distribution of dwelling fires, fire stations, and arrival times in London, 2015. The upper right panel shows that the percentage rate of arrival times > 6 min is higher close to the borders of London, besides, most fires occur in the city center illustrated in the lower left panel.

6.2. Complex Space–Time Interactions in a Cox Model

This analysis is based on the article of Taylor (2017) and contributes to the developed model by inclusion of complex space–time interactions using the BAMLSS framework.

The *London Fire Brigade* (LFB, *http://www.london-fire.gov.uk/*) is one of the largest in the world. Each year, the LFB is called thousands of times, in most cases due to dwelling fires. To prevent further damage or fatal casualties, a short arrival time is important, that is, the time it takes until a fire engine arrives at the scene after an emergency call has been received. The LFB's annual performance target is an average fire engine arrival time of 6 min at maximum. Clearly, this mostly depends on the distance between the site and the responsible fire station but it may also depend on the number of fire stations in the area because fire engines may already be in use at another nearby fire scenery. Therefore, Taylor (2017) analyzed the effect of fire station closures in 2014 using a parametric proportional hazards model to identify regions of possible concern about the number of available fire stations. To contribute to the topic, we apply an extended complex Cox model to the 2015 dwelling fire response time data and illustrate how the generic BAMLSS framework can be used to set up new estimation algorithms for this type of model.

The distribution of the 5838 dwelling fires in 2015 together with the fire stations is shown in Figure 5 (a precompiled version of the data is available in the bamlss package). The top left panel indicates that both, fire stations and fire events, are spread all over London with a higher density in the city center which is brought out more clearly by the heatmap in the bottom left panel. The panels on the right-hand side pertain to the arrival time and show that overall about 30% of these were greater than 6 min (bottom right) with most of these occurring at the borders of London (top right).

Taylor (2017) analyzed the response times within a survival context where the hazard of an event (fire engine arriving) at time t with a relative risk model of the form

$$\lambda(t) = \exp(\eta(t)) = \exp(\eta_\lambda(t) + \eta_\gamma),$$

that is, a model for the instantaneous arrival rate conditional on the engine not having arrived before time t. Here, the hazard function is assumed to depend on a time-varying predictor $\eta_\lambda(t)$ and a time-constant predictor η_γ. In most survival models, the time-varying part $\eta_\lambda(t)$ represents the so-called baseline hazard and is a univariate function of time t. Compared to Taylor (2017), we set up a similar model but with the extended time-constant predictor

$$\eta_\gamma = \beta_0 + f_1(\texttt{fsintens}) + f_2(\texttt{daytime}) + f_3(\texttt{lon}, \texttt{lat}) + f_4(\texttt{daytime}, \texttt{lon}, \texttt{lat}),$$

where β_0 is an intercept and function f_1 is the effect of fire station intensity (`fsintens`, computed with a kernel density estimate of all fire stations in London). Thus, this variable is a proxy for the distance to the next fire station(s), especially suited for situations when the responsible fire station already send out all fire engines such that help needs to arrive from another station. Function f_2 accounts for the effect that it is more difficult for a fire engine to arrive at the scene in rush hours, that is, the risk of waiting longer than 6 min is expected to depend on the time of the day, variable `daytime`. To treat the question of structured spatially driven hazards, a spatial effect f_3 of longitude and latitude coordinates is included in the model. Moreover, we also

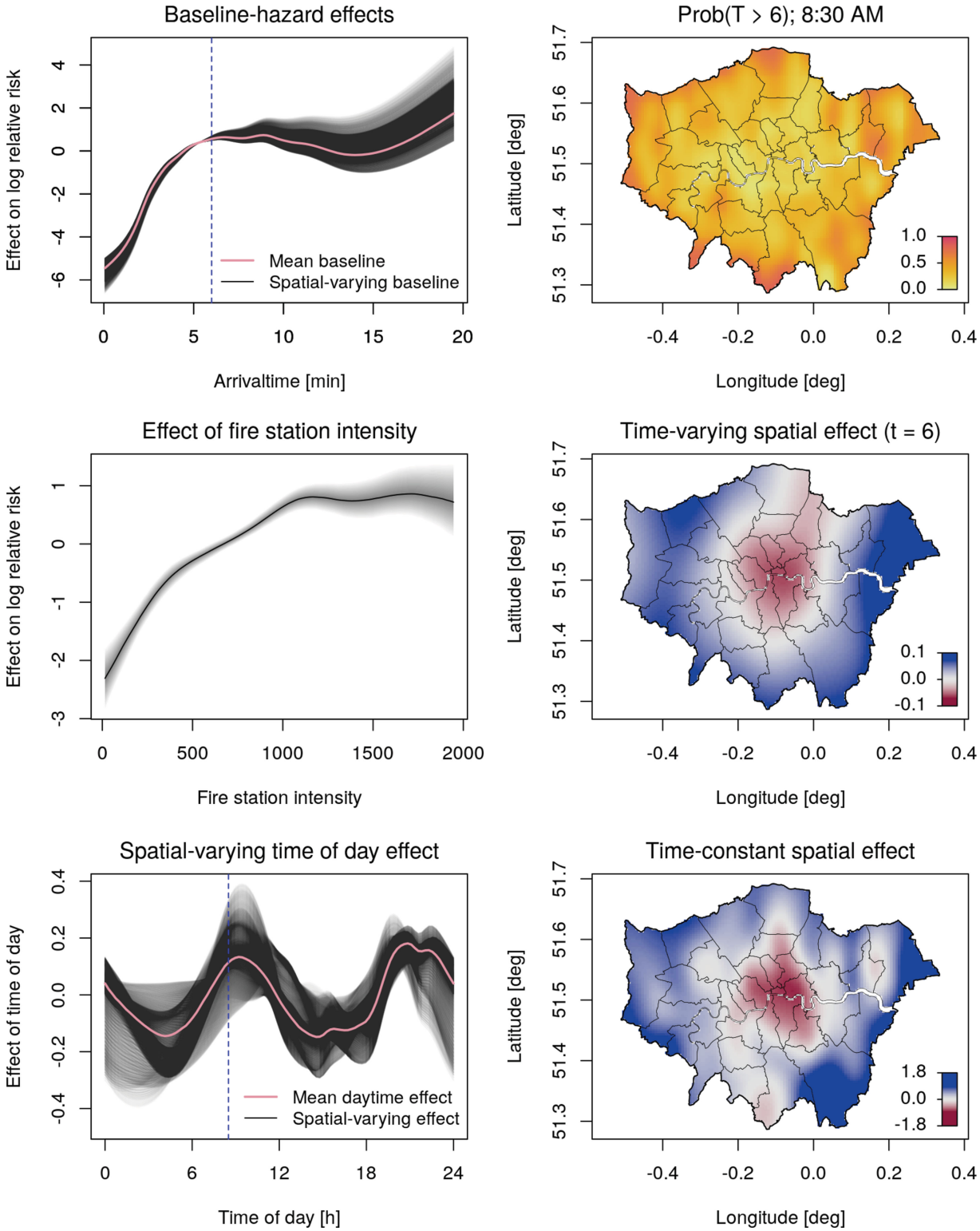

Figure 6. Estimated effects of the fire emergency response times survival model. Top left panel shows the mean baseline effect, red line, together with the spatially varying effects, black lines. The 6 min target waiting time is represented by the blue dashed vertical line. The upper right panel shows the estimated probability of waiting longer than 6 min until the first engine arrives at 8:30 am. The space-time varying effect is illustrated at 6 min waiting time in the second row, right panel. The time of day effect again shows the mean effect as red lines and spatial deviations by black lines.

treat the daytime effect in a spatially correlated context, function f_4. For example, we assume that rush hour peaks may have local hot spots that can be captured by this three-dimensional effect. Again, all functions $f_1, \ldots, f_4$ are assumed to be possibly nonlinear and are modeled using penalized splines.

Moreover, we also relax the time-varying predictor to $\eta_\lambda(t) = f_0(t) + \sum_{j=1}^{J_\lambda} f_j(t, \mathbf{x})$. Here, the baseline hazard is represented by $f_0(t)$ and all functions $f_j(t, \mathbf{x})$ are time-varying possibly nonlinear functions of covariates. Hence, our model is a complex Cox-type additive model as introduced by Kneib and Fahrmeir (2007). To further investigate if there is a space-time varying effect, that is, if the shape of the baseline hazard is dependent on the location we use the following time-varying additive predictor

$$\eta_\lambda(\texttt{arrivaltime}) = f_0(\texttt{arrivaltime}) + f_1(\texttt{arrivaltime}, \texttt{lon}, \texttt{lat}),$$

where f_0 is the baseline hazard for variable arrivaltime, the waiting time until the first fire engine arrives after the received emergency call. Function $f_1(\texttt{arrivaltime}, \texttt{lon}, \texttt{lat})$ is a space-time varying effect modeling the deviations from the baseline which can capture whether the risk of waiting longer than 6 min is driven by other factors that are not available in this analysis. Both functions are modeled using penalized splines.

The probability that the engine will arrive on the scene after time t is described by the survival function $S(t) = \text{Prob}(T > t) = \exp(-\int_0^t \lambda(u)du)$, which is of prime interest in this analysis. For full Bayesian inference, the following "Lego bricks" need to be implemented for updating functions $U_{jk}(\cdot)$ using algorithms A1, A2a, and A2b (the detailed expressions are provided in the supplemental material Section 3):

- B1. The log-likelihood function of the continuous time Cox model.
- B6a. For derivative-based estimation using Algorithm A2a and for MCMC simulation with Algorithm A2b, the score vectors and Hessian need to be computed.
- B7a. The elements of the Hessian w.r.t. $\boldsymbol{\beta}_\lambda$. Note that these cannot be computed by fragmenting with the chain rule to obtain building block B7b and IWLS updating functions, see the supplemental material Section 3 for details.
- B6b & B7b. However, constructing updating functions for the time-constant part η_γ again yields an IWLS updating scheme based on building block B7b.

As a result, applying the generic algorithm presented in Algorithm A1 to this type of problem, two specific difficulties need to be considered. First, the updating functions $U_{jk}(\cdot)$ for the time-varying predictor $\eta_\lambda(t)$ are different from the time-constant updating functions for η_γ. Second, a specific hurdle of the continuous-time Cox model is the computation of the integrals, because these do not have a closed-form solution and need to be approximated numerically, for example, by the trapezoidal rule or Gaussian quadrature (Hofner 2008; Waldmann et al. 2017). Moreover, it is inefficient to compute the integrals anew for every updating step, since for the time-constant part the integrals given in $\mathbf{P}$ do not change anymore.

To reduce computing time, we account for the idiosyncrasy of the Cox model and implement an optimizer function cox.mode() for posterior mode estimation as well as the sampler function cox.mcmc() for MCMC simulation (which are part of the corresponding bamlss family object cox_bamlss()). On a Linux system with 8 Intel i7-2600 3.40 GHz processors estimation takes approximately 1.2 days. Note that function cox.mode() also applies an automated procedure for smoothing variances selection using information criteria, see also Algorithm A2a.

The estimated effects are shown in Figure 6. The upper left panel shows that the average "risk" that a fire engine arrives increases steeply until the target time of 6 min. The space-time varying effect is relatively small compared to the overall size of the effect, especially until the 6 min target time it seems that the location does not have a great influence on the relative risk. Only for waiting times above ∼15 min, the space-time varying effect is more pronounced. The effect for fire station intensity is quite large and bounded, that is, there is a natural limit for the benefit from opening new fire stations in the area. The effect of the time of the day then indicates that in the morning hours around 4–5 am, as well as in the afternoon around 2–4 pm, the risk of waiting longer for the fire engine to arrive is only slightly increasing. In addition, the spatial deviations from the mean time of day effect are modest, similar in magnitude as the spatial varying baseline effects. The largest deviation seems to be at around 10 am. In Figure 7, the spatial varying effect is illustrated on nine time points. The maps indicate possible hot-spots of this effect, however, as mentioned above the overall effect size from −0.4 to 0.4 is not very large (see also Figure 6 bottom left) such that differences in risk probabilities are almost negligible. In contrast, the time-constant spatial effect clearly shows that the average risk of increased waiting times are higher in the city center and some smaller area in southern London. However, the estimated probabilities of waiting longer than 6 min around the center show moderate variation, while the borders of London indicate higher probabilities as well as in the western parts, most probably because of the lower fire station density in these areas. In summary, next to the baseline effect, the most important effects on the log risk are the fire station intensity and the time-constant spatial effect which have an absolute range of about 4 on the log-scale.

To conclude, the proposed model including complex model terms beyond "classical" structures, like space–time interactions in both the time-constant and the time-varying part, is a considerable extension of this type of model and can gain more insight into potential risk factors that are probably not obvious. The presented modular framework facilitates the development of such complex algorithms essentially, for example, Köhler et al. (2017) develop flexible joint models for longitudinal and time-to-event data using the modular BAMLSS framework.

7. Summary and Discussion

This article combines frequently used algorithms for the estimation of additive Bayesian models in a flexible framework for distributional regression, also termed Bayesian additive models for location, scale, and shape (BAMLSS), and beyond. We highlight the similarities between optimization and sampling concepts and coalesce these in a generic toolbox of modular "Lego bricks." Two case studies illustrate how the framework can be leveraged to establish complex and difficult-to-estimate models

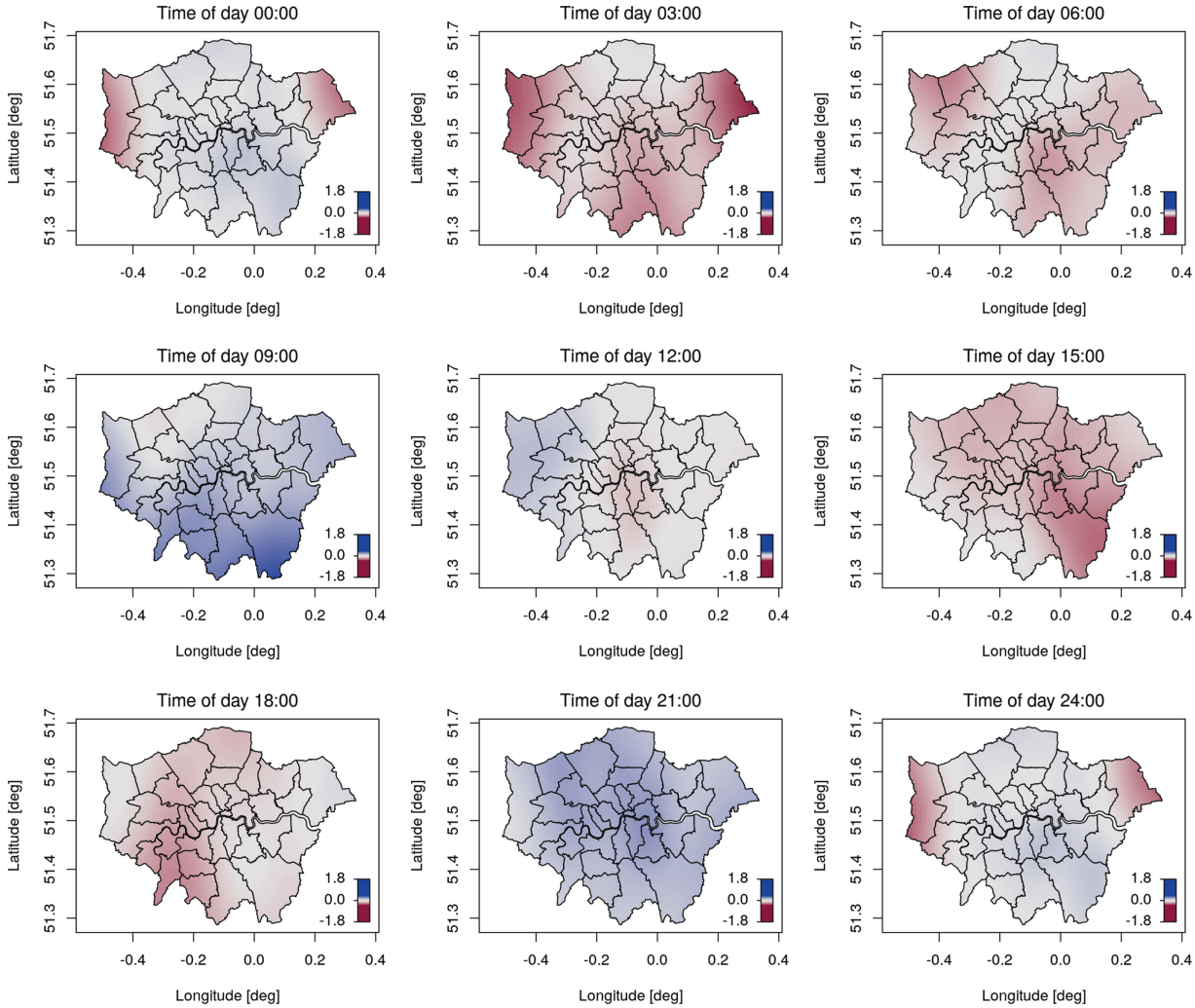

Figure 7. Estimated spatial varying time-of-the-day effect. The figure shows that in the early morning hours, as well as in the afternoon, the effect on the probability that an engine arrives are negative in certain areas, however, relatively small compared to the other estimated effects.

based on the accompanying implementation in the R package bamlss (Umlauf et al. 2017). One drawback of our framework when using an MCMC engine with highly complex predictor structures in the distribution parameters and big datasets is a possibly long computing time. To overcome this problem, one may either resort to parallel computing facilities—if available—or approximative, faster methods like integrated nested Laplace approximations (Rue, Martino, and Chopin 2009). The latter however, is currently restricted to distributions with only one predictor and hence not usable for instance a heteroscedastic Gaussian model.

Interesting extensions in the future include efficient matrix transformations for additional effect types based on Gaussian processes (Paciorek 2007) and the broadening of the available model classes in bamlss such as state-space models (Carter and Kohn 1994).

Supplementary Materials

Algorithmic details: Detailed information about algorithmic derivations and "Lego bricks" used in the main manuscript are provided in the accompanying online supplementary material `.pdf`.

R package: R package bamlss, including the datasets for reproducing the illustrations, is available at *http://CRAN.R-project.org/package=bamlss*.

Code: The R code for reproducing the models is provided in the script `models.R`, for reproducing all figures in the script `figures.R`.

References

Belitz, C., Brezger, A., Kneib, T., Lang, S., and Umlauf, N. (2017), *BayesX – Software for Bayesian Inference in Structured Additive Regression Models, Version 3.0.2*, available at *http://BayesX.org*. [612,619]

Belitz, C., and Lang, S. (2008), "Simultaneous Selection of Variables and Smoothing Parameters in Structured Additive Regression Models," *Computational Statistics & Data Analysis*, 53, 61–81. [614]

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017), "Stan: A Probabilistic Programming Language," *Journal of Statistical Software*, 76(1), 1–32. [612]

Carter, C. K., and Kohn, R. (1994), "On Gibbs Sampling for State Space Models," *Biometrika*, 81, 541–553. [626]

Chambers, J. M., and Hastie, T. J. (eds.) (1992), *Statistical Models in S*, London: Chapman & Hall. [619]

Fahrmeir, L., Kneib, T., and Lang, S. (2004), "Penalized Structured Additive Regression for Space Time Data: A Bayesian Perspective," *Statistica Sinica*, 14, 731–761. [612,613]

Fahrmeir, L., Kneib, T., Lang, S., and Marx, B. (2013), *Regression – Models, Methods and Applications*, Berlin: Springer-Verlag. [612,613,615]

Gamerman, D. (1997), "Sampling From the Posterior Distribution in Generalized Linear Mixed Models," *Statistics and Computing*, 7, 57–68. [614]

Gelman, A. (2006), "Prior Distributions for Variance Parameters in Hierarchical Models," (Comment on Article by Browne and Draper), *Bayesian Analysis*, 1, 515–534. [616]

Hastie, T., and Tibshirani, R. (1990), *Generalized Additive Models*, Boca Raton, FL: Chapman & Hall/CRC. [612,619]

Hofner, B. (2008), "Variable Selection and Model Choice in Survival Models with Time-Varying Effects," Ph.D. dissertation, Institut für Statistik. [625]

Klein, N., and Kneib, T. (2016), "Scale-Dependent Priors for Variance Parameters in Structured Additive Distributional Regression," *Bayesian Analysis*, 11, 1071–1106. [616,618]

Klein, N., Kneib, T., Klasen, S., and Lang, S. (2015a), "Bayesian Structured Additive Distributional Regression for Multivariate Responses," *Journal of the Royal Statistical Society, Series C*, 64, 569–591. [613]

Klein, N., Kneib, T., and Lang, S. (2015), "Bayesian Generalized Additive Models for Location, Scale and Shape for Zero-Inflated and Overdispersed Count Data," *Journal of the American Statistical Association*, 110, 405–419. [612,617]

Klein, N., Kneib, T., Lang, S., and Sohn, A. (2015b), "Bayesian Structured Additive Distributional Regression with an Application to Regional Income Inequality in Germany," *Annals of Applied Statistics*, 9, 1024–1052. [613]

Kneib, T., and Fahrmeir, L. (2007), "A Mixed Model Approach for Geoadditive Hazard Regression," *Scandinavian Journal of Statistics*, 34, 207–228. [625]

Köhler, M., Umlauf, N., Beyerlein, A., Winkler, C., Ziegler, A.-G., and Greven, S. (2017), "Flexible Bayesian Additive Joint Models With an Application to Type 1 Diabetes Research," *Biometrical Journal*, 59, 1144–1165. [625]

Krige, D. G. (1951), "A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand," *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52, 119–139. [620]

Lang, S., Umlauf, N., Wechselberger, P., Harttgen, K., and Kneib, T. (2014), "Multilevel Structured Additive Regression," *Statistics and Computing*, 24, 223–238. [616,621]

Lunn, D. J., Spiegelhalter, D., Thomas, A., and Best, N. (2009), "The BUGS Project: Evolution, Critique and Future Directions," *Statistics in Medicine*, 28, 3049–3082. [612]

Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000), "WinBUGS – A Bayesian Modelling Framework: Concepts, Structure, and Extensibility," *Statistics and Computing*, 10, 325–337. [612]

Neal, R. M. (2003), "Slice Sampling," *The Annals of Statistics*, 31(3), 705–767. [618]

Nemec, J., Chimani, B., Gruber, C., and Auer, I. (2011), "Ein Neuer Datensatz Homogenisierter Tagesdaten," *ÖGM Bulletin*, 2011, 19–20. [621]

Nemec, J., Gruber, C., Chimani, B., and Auer, I. (2013), "Trends in Extreme Temperature Indices in Austria Based on a New Homogenised Dataset," *International Journal of Climatology*, 33, 1538–1550. [621]

Paciorek, C. (2007), "Bayesian Smoothing With Gaussian Processes Using Fourier Basis Functions in the spectralGP Package," *Journal of Statistical Software*, 19, 1–38. [626]

Plummer, M. (2003), "JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling," in *Proceedings of the 3rd International Workshop on Distributed Statistical Computing, Vienna, Austria*, eds, K. Hornik, F. Leisch, and A. Zeileis, available at *http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/*. [612,619]

Polson, N. G., and Scott, J. G. (2012), "On the Half-Cauchy Prior for a Global Scale Parameter," *Bayesian Analysis*, 7, 887–902. [616]

R Core Team (2016), *R: A Language and Environment for Statistical Computing*, Vienna, Austria: R Foundation for Statistical Computing. [612]

Rigby, R. A., and Stasinopoulos, D. M. (2005), "Generalized Additive Models for Location, Scale and Shape," *Journal of the Royal Statistical Society, Series C*, 54, 507–554. [612,613,616,617]

Rue, H., Martino, S., and Chopin, N. (2009), "Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations," *Journal of the Royal Statistical Society*, Series B, 71, 319–392. [626]

Ruppert, D., Wand, M. P., and Carrol, R. J. (2003), *Semiparametric Regression*, New York: Cambridge University Press. [612,613]

Silverman, B. W. (1985), "Some Aspects of the Spline Smoothing Approach to Non-Parametric Regression Curve Fitting," *Journal of the Royal Statistical Society*, Series B, 47, 1–52. [620]

Simpson, D., Rue, H., Riebler, A., Martins, T. G., and Sørbye, S. H. (2017), "Penalising Model Component Complexity: A Principled, Practical Approach to Constructing Priors," *Statistical Science*, 32, 1–28. [616]

Smyth, G. (1996), "Partitioned Algorithms for Maximum Likelihood and Other Non-Linear Estimation," *Statistics and Computing*, 6, 201–216. [614]

Stauffer, R., Mayr, G. J., Messner, J. W., Umlauf, N., and Zeileis, A. (2017), "Spatio-Temporal Precipitation Climatology over Complex Terrain Using a Censored Additive Regression Model," *International Journal of Climatology*, 37, 3264–3275. [620]

Taylor, B. M. (in press), "Spatial Modelling of Emergency Service Response Times," *Journal of the Royal Statistical Society*, Series A, 180, 433–453. [623]

Umlauf, N., Adler, D., Kneib, T., Lang, S., and Zeileis, A. (2015), "Structured Additive Regression Models: An R Interface to BayesX," *Journal of Statistical Software*, 63(1), 1–46. [612]

Umlauf, N., Klein, N., Zeileis, A., and Köhler, M. (2017), *bamlss: Bayesian Additive Models for Location Scale and Shape (and Beyond), R Package Version 1.0-0*, available at *https://CRAN.R-project.org/package=bamlss*. [613,619,626]

Umlauf, N., Mayr, G., Messner, J., and Zeileis, A. (2012), "Why Does It Always Rain on Me? A Spatio-Temporal Analysis of Precipitation in Austria," *Austrian Journal of Statistics*, 41, 81–92. [621]

Waldmann, E., Taylor-Robinson, D., Klein, N., Kneib, T., Pressler, T., Schmid, M., and Mayr, A. (2017), "Boosting Joint Models for Longitudinal and Time-to-Event Data," *Biometrical Journal*, 59, 1104–1121. [625]

Wilkinson, G. N., and Rogers, C. E. (1973), "Symbolic Description of Factorial Models for Analysis of Variance," *Journal of the Royal Statistical Society, Series C*, 22, 392–399. [619]

Wood, S. N. (2004), "Stable and Efficient Multiple Smoothing Parameter Estimation for Generalized Additive Models," *Journal of the American Statistical Association*, 99, 673–686. [613]

——— (2016), "Just Another Gibbs Additive Modeler: Interfacing JAGS and mgcv," *Journal of Statistical Software*, 75(7), 1–15. [613]

Zeileis, A., and Croissant, Y. (2010), "Extended Model Formulas in R: Multiple Parts and Multiple Responses," *Journal of Statistical Software*, 34(1), 1–13. [619]