
Australian & New Zealand Journal of Statistics

Aust. N. Z. J. Stat. **60**(3), 2018, 279–300 doi: 10.1111/anzs.12241

Generalised additive mixed models analysis via `gammSlice`

Tung H. Pham[1] **and Matt P. Wand**[2,*]

University of Melbourne and University of Technology Sydney

Summary

We demonstrate the use of our R package, `gammSlice`, for Bayesian fitting and inference in generalised additive mixed model analysis. This class of models includes generalised linear mixed models and generalised additive models as special cases. Accurate Bayesian inference is achievable via sufficiently large Markov chain Monte Carlo (MCMC) samples. Slice sampling is a key component of the MCMC scheme. Comparisons with existing generalised additive mixed model software shows that `gammSlice` offers improved inferential accuracy, albeit at the cost of longer computational time.

Key words: generalised additive models; generalised linear mixed models; slice sampling; Markov chain Monte Carlo; penalised splines; R.

1. Introduction

We demonstrate use of our R package named `gammSlice` (Pham & Wand 2018) for generalised additive mixed model (GAMM) analysis within the R computing environment (R Development Core Team 2018). Generalised linear mixed models (GLMM) and generalised additive models (GAM) are handled as a special cases of GAMM. In these contexts, the term *mixed* refers to the fact that the model has both fixed effects and random effects. As the name suggests, *slice sampling* (e.g. Neal 2003), a special type of Markov chain Monte Carlo (MCMC) sampling, is used for fitting and inference within a Bayesian framework. The attraction of MCMC for GAMM analysis is the '*exactness*' of the inference. By this, we mean that numerical summaries of actual posterior density functions of parameters of interest, such as equal-tailed 95% credible sets, can be made arbitrarily accurate by drawing sufficiently large MCMC samples. More established R packages for GAMM analysis, such as Wood (2017), use the Laplace approximation-based *penalised quasi-likelihood* (PQL). However, PQL is susceptible to inaccuracies, as demonstrated by Breslow & Lin (1995) and in Section 4 of this article. The more accurate Laplace approximation used in Bates *et al.* (2015) for GLMM has not yet been extended to the GAMM case.

GAMM extends GLMM by allowing continuous predictors to have a nonparametric functional impact on the mean response. Suppose that, in a longitudinal data set, y_{ij} is the jth measurement on the ith subject of a binary response variable y for each of $1 \leqslant j \leqslant n_i$, $1 \leqslant i \leqslant m$. Let x_{1ij} and x_{2ij} be defined similarly for continuous predictors x_1 and x_2. A GLMM for this model is

*Author to whom correspondence should be addressed.
[1]School of Mathematics and Statistics, University of Melbourne, Melbourne, VIC 3000, Australia
[2]School of Mathematical and Physical Sciences, University of Technology Sydney, P.O. Box 123, Broadway, NSW 2007, Australia e-mail: matt.wand@uts.edu.au
Acknowledgement. This research was partially funded by an Australian Research Council Discovery Project. We are grateful for comments from an associate editor and two referees.

$$y_{ij}|U_i \overset{\text{ind}}{\sim} \text{Bernoulli}(\text{logit}^{-1}(U_i+\beta_1 x_{1ij}+\beta_2 x_{2ij})), \quad U_i \overset{\text{ind}}{\sim} \text{N}(0,\sigma^2) \tag{1}$$

where U_i is a random subject intercept. The notation in (1) is: $x_i \overset{\text{ind}}{\sim} \mathcal{D}_i$ means that the x_i s are independent with distributions $\mathcal{D}_i$, $x \sim$ Bernoulli(p) means that x has a Bernoulli distribution with mean p and $\text{logit}^{-1}(x) = e^x/(1+e^x)$. A GAMM for the same data is

$$y_{ij}|U_i \overset{\text{ind}}{\sim} \text{Bernoulli}(\text{logit}^{-1}(U_i+f_1(x_{1ij})+f_2(x_{2ij}))), \quad U_i \overset{\text{ind}}{\sim} \text{N}(0,\sigma^2)$$

where f_1 and f_2 are functions that are assumed to be smooth but otherwise arbitrary. The types of application for which such models are useful include those in the biomedical realm based on longitudinally followed subjects (e.g. Fitzmaurice *et al.* 2008), educational research in which there is multilevel structure due to, for example, students being in the same classroom (e.g. Goldstein 2010) and econometrics based on so-called panel data (e.g. Balgtagi 2013). There are numerous ways by which f_1 and f_2 can be estimated. In `gammSlice`, $f_1(x_1)+f_2(x_2)$ is modelled using mixed model-based penalised splines:

$$f_1(x_1)+f_2(x_2) = \beta_0+\beta_{x1}x_1+\sum_{k=1}^{K_1} u_{1k}z_{1k}(x_1)+\beta_{x2}x_2+\sum_{k=1}^{K_2} u_{2k}z_{2k}(x_2)$$
$$\text{where} \quad u_{1k} \overset{\text{ind}}{\sim} \text{N}(0,\sigma_1^2),\ 1 \leqslant k \leqslant K_1 \quad \text{and} \quad u_{2k} \overset{\text{ind}}{\sim} \text{N}(0,\sigma_2^2),\ 1 \leqslant k \leqslant K_2.$$

The subscripted zs are the orthogonalised O'Sullivan spline basis functions (Wand & Ormerod 2008). Further details on their specification are given in Sections 2 and 3.2.

The `gammSlice` package is essentially an MCMC implementation of GAMM, GLMM and GAM special cases of the *general design Bayesian generalised linear mixed models* described in Zhao *et al.* (2006). The reader is referred to that paper for details on this very wide class of models.

As of early 2018, there are more than a dozen R packages on the Comprehensive R Archive Network (`www.cran.r-project.org`) that support GLMM analysis. Many of them, e.g. MASS (Venables & Ripley 2002), `gamm4` (Wood & Scheipl 2017) and `mgcv` (Wood 2017), use Laplace approximation, but a few, e.g. `glmmBUGS` (Brown & Zhou 2018), `MCMCglmm` (Hadfield 2017), `R2BayesX` (Umlauf *et al.* 2017) and `spikeSlabGAM` (Scheipl 2011), use MCMC methods. A method based on exact maximum likelihood, via quadrature, is supported by `glmmML` (Broström & Holmberg 2018) and `lme4` (Bates *et al.* 2015). To the best of our knowledge, the GAMM extension is only supported by `gamm4`, `mgcv`, `R2BayesX` and `spikeSlabGAM`. The first two of the these packages use Laplace approximation and are quite similar in terms of GAMM analysis. The third and fourth support GAMM fitting via MCMC. Scheipl (2011) provides detailed a description of the capabilities of `spikeSlabGAM`, although this paper focusses on generalised additive models rather than GAMM. The `spikeSlabGAM` models differ from `gammSlice` and `mgcv` in terms of function estimation in that they use spike-and-slab type prior distributions on the spline coefficients. The package `R2BayesX` purports to support similar models to those supported by `gammSlice`. In early 2017, we tested some GAMM analyses for the examples of Sections 3 and 5 in Version 1.1 of `R2BayesX` but most of them led to breakdowns due the model not being supported. The landscape is ever-changing, but as of November 2017 we believe that `gammSlice` is the only R package that provides MCMC-based inference for GAMM analyses using the simple penalised spline approach that `mgcv` uses.

Section 2 contains an overview of the gammSlice package. Our demonstration of gammSlice begins in Section 3 with four illustrative examples based on simulated data. In Section 4 we illustrate the accuracy advantages of gammSlice over some existing GAMM software. Section 5 contains some applications of gammSlice. In Section 6, we briefly discuss possible extensions of gammSlice. An appendix gives some details on the sampling scheme used by gammSlice for Bayesian inference in GAMM analysis.

2. Overview of gammSlice

The centerpiece of the gammSlice package is the function gSlc(). The main argument of gSlc() is formula, which is used for specification of a GAMM corresponding to the data in the data argument.

The formula argument is analogous to that in the popular GAM and GAMM package mgcv and supports the specification of additive functions of the predictors. Numerical-valued predictors may enter the model linearly or in the form of arbitrary smooth functions of the predictors, although this latter situation is only appropriate when a predictor that is used in this way has many unique values. Smooth function specification is done via the function s(). For example, the formula

```
y~x1 + x2 + s(x3) + s(x4)
```

corresponds to a GAM or GAMM with linear predictor of the form

$$\beta_1 x_1 + \beta_2 x_2 + f_3(x_3) + f_4(x_4).$$

In gSlc(), we use the same constructor function s() that is used by gam() and gamm() in the mgcv package. The R command

```
help(s, package = "mgcv")
```

provides a full description of s() and further details, in the context of some examples, are given in Section 3. The formula argument also supports categorical predictors entering the model via the factor() constructor function. For example if x1 and x2 are numerical predictors but x3 contains information on gender via the character strings "female" and "male" then

```
y~x1 + s(x2) + factor(x3)
```

will set up an indicator variable for an observation being in the male category. The female category serves as a baseline since "female" precedes "male" in lexicographic ordering.

A random effect structure can be specified via the random argument. The family argument allows specification of the distribution family of the response variables and is currently restricted to "binomial" and "poisson". Additional preferences can be specified through the control argument and the companion function gSlc.control(). Further details can be obtained from issuing the following commands in R:

```
library(gammSlice)
help(gSlc)
help(gSlc.control)
```

The gammSlice package also has two methods functions for objects produced by gSlc(). The main methods function is summary.gSlc(), which provides summaries of

the MCMC output and Bayesian inferential summaries of model parameters. The other methods function is `plot.gSlc()`, which supports display of estimated functions. More details are available via:

```
help(summary.gSlc)
help(plot.gSlc)
```

The only other components of `gammSlice` are data-sets named `indonRespir` and `toenail`. In Section 5, these datasets are used to illustrate GAMM analyses via `gammSlice`.

3. Illustrative examples using simulated data

We first give examples involving simulated data. These allow the essence of `gammSlice` to be conveyed without the complexities associated with data arising from applied studies.

3.1. Simple logistic mixed model

Our first simulated data example involves the simple Bayesian logistic mixed model

$$\begin{aligned} y_{ij}|x_{ij},\beta_0,\beta_x,U_i &\overset{\text{ind}}{\sim} \text{Bernoulli}(\text{logit}^{-1}(\beta_0+\beta_x x_{ij}+U_i)), \quad 1\leqslant i\leqslant m,\ 1\leqslant j\leqslant n,\\ x_{ij} &\overset{\text{ind}}{\sim} \text{Uniform}(0,1), \quad U_i|\sigma^2 \overset{\text{ind}}{\sim} \text{N}(0,\sigma^2)\\ \beta_0,\beta_x &\overset{\text{ind}}{\sim} \text{N}(0,\sigma^2_{\boldsymbol{\beta}}), \quad \sigma\sim\text{Half-Cauchy}(A). \end{aligned} \tag{2}$$

The notation $\sigma\sim$Half-Cauchy(A) means that σ has density function $p(\sigma)=2A/(\pi(\sigma^2+A^2)),\quad \sigma>0$ and $A>0$. The hyperparameters σ^2_β and A can be specified by the user. Their default values are discussed later in this section.

To illustrate fitting (2) in `gammSlice` we generate data with $m=100$, $n=2$, $\beta_0=0.5$, $\beta_x=1.7$ and $\sigma^2=0.8$ as follows:

```
set.seed(39402)
m <- 100
n <- 2
beta0True <- 0.5
betaxTrue <- 1.7
sigsqTrue <- 0.8
idnum <- rep(1:m, each=n)
x <- runif(m * n)
U <- rep(rnorm(m, 0, sqrt(sigsqTrue)), each = n)
mu <- 1/(1 + exp(-(beta0True + betaxTrue * x + U)))
y <- rbinom((m * n), 1, mu)
```

Fitting is then performed using:

```
fit1 <- gSlc(y~x, random = list(idnum =~1), family = "binomial")
```

To obtain a summary of the slice sample output we use:

```
summary(fit1)
```

The resulting summary is shown in Figure 1. Column 2 contains trace plots of the MCMC samples, commonly called *chains*, which are simply the values plotted in time order. The third column plots each value of the chain against its previous value to allow quick visual

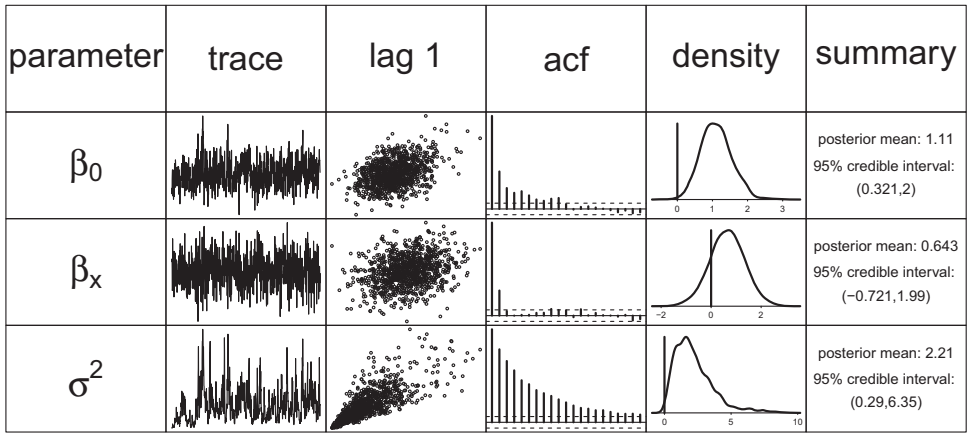

Figure 1. Summary of slice-based MCMC samples for fitting the simple logistic mixed model (2) in `gammSlice`. This plot was produced via the command `summary(fit1)` where `fit1` is the fitted object from a call to `gSlc()`. The columns are as follows: (1) parameter name, (2) trace plot of the MCMC sample, (3) plot of MCMC sample against its lag 1 sample, (4) sample autocorrelation function, (5) kernel density estimate of posterior density function, (6) numerical summaries of posterior density function.

assessment of the amount of serial correlation. Column 4 shows the sample autocorrelation functions for each chain, which is a more formal way to assess chain correlations at several lags. Then in column 5, there are kernel estimates of the posterior density functions for each parameter, followed by numerical summmaries in column 6 – each based on the relevant chain.

Columns 2–4 in Figure 1 provide indications of MCMC convergence, which is seen to be quite good: the trace plots show no trends, the lag 1 correlation is mild and the autocorrelation function spikes decay reasonably well. The last two columns show the estimated posterior density function and some basic numerical summaries.

3.1.1 Pre-transformation of input data

The default version of the `gSlc()` function in `gammSlice` applies the pre-transformation:

$$(x_{ij} - \min(x_{ij}))/(\max(x_{ij}) - \min(x_{ij}))$$

to the x_{ij} values before the slice sampling takes place. The corresponding back-transformation is applied to the MCMC output. This strategy ensures that the results are scale invariant when default hyperparameters are used. These defaults are discussed next.

3.1.2 Default settings and their modification

The `gSlc()` function has several default settings. Here we briefly describe the most important of these and how the user may modify them if desired.

The default hyperparameters in `gSlc()` are as follows:

$$\texttt{fixedEffPriorVar} = \sigma_{\boldsymbol{\beta}}^2 = 10^{10}$$
$$\text{and} \quad \texttt{sdPriorScale} = A = 10^5.$$

If, for example, one would like to have the prior distributions set to

$$\beta_0, \beta_x \overset{\text{ind}}{\sim} \text{N}(0, 10^{13}), \qquad \sigma \sim \text{Half-Cauchy}(10^3)$$

then the following command should be used:

```
fit1MyPriors <- gSlc( y~x,random = list(idnum =~1),
                      family = "binomial",
                      control = gSlc.control(fixedEffPriorVar=1e13,
                      sdPriorScale=1e3))
```

The slice sampling sample sizes in `gSlc()` have the following default values:

`nBurn` = number of burn-in iterations = 5000,
`nKept` = number of kept iterations = 5000
and `nThin` = thinning factor = 5.

To specify, say, a burn-in of size 10,000, number of kept iterations equal to 8000, and a thinning factor of 10, the call to `gSlc()` should be:

```
fit1BigMCMC <- gSlc(y~x, random = list(idnum =~1),
                    family = "binomial",
                    control = gSlc.control(nBurn = 10000,
                    nKept = 8000, nThin = 10))
```

3.1.3 Extraction of Markov chain Monte Carlo samples

The `summary` and `plot` commands allow a quick and convenient examination of the `gSlc()` fit. However, the user may wish to produce other summaries based on the MCMC output. Relevant values can be extracted via commands such as:

```
betaMCMC <- fit1$beta
sigmaSquaredMCMC <- fit1$sigmaSquared
```

3.1.4 Construction of additional summaries

The MCMC output can be used to construct summaries concerning the fitted model. For example, the odds ratio, conditional on each subject's random intercept, for x at its third quartile (Q_3) compared with x at its first quartile (Q_1) is

$$\text{OR}_x = \exp(\beta_x(Q_3 - Q_1)).$$

The following code computes the approximate posterior mean odds ratio, and corresponding 95% credible set, for OR_x:

```
Q1 <- as.numeric(quantile(x, 0.25))
Q3 <- as.numeric(quantile(x, 0.75))
betaxMCMC <- fit1beta[,1]
ORxMCMC <- exp(betaxMCMC * (Q3 - Q1))
postMeanORx <- mean(ORxMCMC)
credSetORx <- as.numeric(quantile(ORxMCMC, c(0.025, 0.975)))
```

For the data and MCMC output corresponding to Figure 1, we get the posterior mean for OR_x equal to 1.53 and corresponding 95% credible set equal to $(0.720, 2.89)$.

3.2. Poisson nonparametric regression

Next, we consider the Poisson nonparametric regression model

$$y_i \overset{\text{ind}}{\sim} \text{Poisson}\big(\exp(f(x_i))\big), \quad 1 \leqslant i \leqslant n. \tag{3}$$

The smooth function f is modelled using penalised splines:

$$f(x) = \beta_0 + \beta_x x + \sum_{k=1}^{K} u_k z_k(x) \qquad \text{where} \qquad u_k \overset{\text{ind}}{\sim} \text{N}(0, \sigma^2). \tag{4}$$

The $\{z_k(\cdot) : 1 \leqslant k \leqslant K\}$ are the orthogonalised O'Sullivan spline functions described in section 4 of Wand & Ormerod (2008). The choice of K is of relatively minor concern for penalised splines. Its default value is $\min(25, n_U/4)$ where n_U is the number of unique x_i values.

The full Bayesian model, amenable to fitting in `gammSlice` and with default hyperparameters, is

$$\begin{aligned} y_i | x_i, \beta_0, \beta_x, u_1, \ldots, u_K &\overset{\text{ind}}{\sim} \text{Poisson}(\exp(\beta_0 + \beta_x x_i + \textstyle\sum_{k=1}^{K} u_k z_k(x_i))), \\ x_i &\overset{\text{ind}}{\sim} \text{Uniform}(0,1), \quad u_k | \sigma^2 \overset{\text{ind}}{\sim} \text{N}(0, \sigma^2), \\ \beta_0, \beta_x &\overset{\text{ind}}{\sim} \text{N}(0, 10^{10}), \quad \sigma \sim \text{Half-Cauchy}(10^5). \end{aligned} \tag{5}$$

To illustrate its `gammSlice` fitting, we simulate data with $n = 400$, the x_is uniformly distributed in $(0, 1)$ and

$$f(x) = \cos(4\pi x) + 2x - 1$$

as follows:

```
set.seed(53902)
n <- 400
x <- runif(n)
y <- rpois(n, exp(cos(4 * pi * x) + 2 * x - 1))
```

To fit (5), we call `gSlc()` as follows:

```
fit2 <- gSlc(y ~s(x), family = "poisson")
```

The function `s()` used here is an internal function that models f as a mixed model-based penalised spline, as described by (4).

There are no clearly identifiable or interpretable parameters in (5). For convergence assessment of such models, we recommend working with *effective degrees of freedom* (edf) measures of the smooth function components. In the current model, with just a single smooth function component, the value of the effective degrees of freedom is defined to be

$$\text{edf} = \text{tr}((\boldsymbol{C}^T \boldsymbol{W} \boldsymbol{C} + \sigma^{-2} \boldsymbol{D})^{-1} \boldsymbol{C}^T \boldsymbol{W} \boldsymbol{C}) \tag{6}$$

where

$$\boldsymbol{C} = [1 \;\; x_i \;\; z_1(x_i) \;\; \cdots \;\; z_K(x_i)]_{1 \leqslant i \leqslant n}, \quad \boldsymbol{D} = \text{diag}(0, 0, 1, \ldots, 1), \qquad \boldsymbol{W} = \text{diag}(\exp(\boldsymbol{f})).$$

and $\boldsymbol{f} = \boldsymbol{C}[\beta_0, \beta_x, u_1, \ldots, u_k]^T$, with $\text{diag}(\boldsymbol{v})$ denoting the diagonal matrix with the entries of $\boldsymbol{v}$ along the diagonal. The justification for (6) is given in Hastie & Tibshirani (1990).

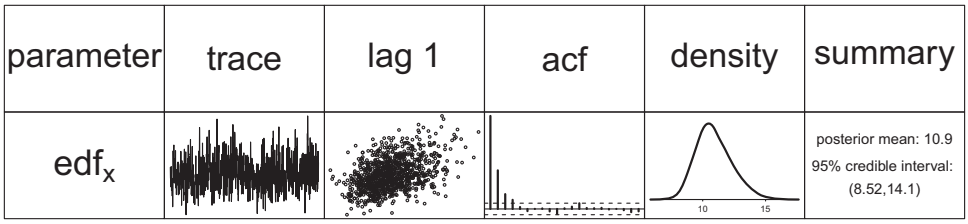

Figure 2. Summary of the MCMC output of the effective degrees of freedom for the penalised spline regression fitted in model (5). This plot was produced via the command `summary(fit2)` where `fit2` is the fitted object from a call to `gSlc()`. The columns are as described in the caption of Figure 1.

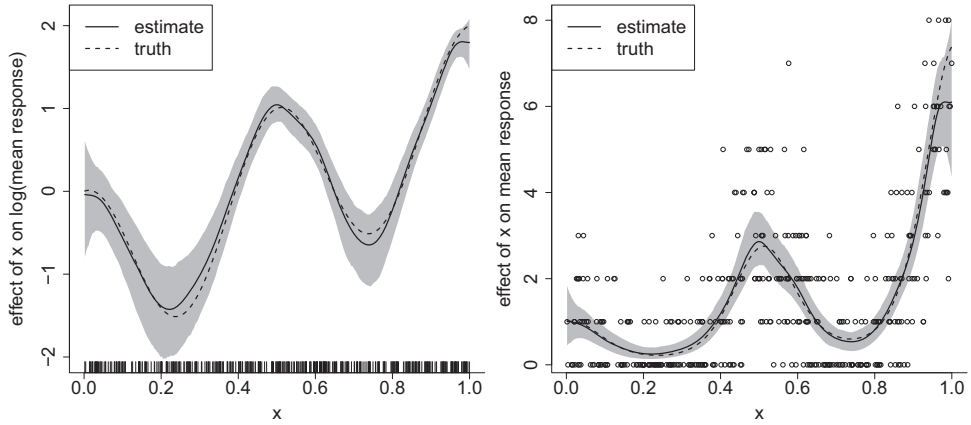

Figure 3. Left panel: Bayesian posterior mean estimate of f (solid line) and pointwise 95% credible intervals (shaded region) obtained from `gSlc()` for the Poisson nonparametric regression model (5). This is obtained via the command `plot(fit2)` where `fit2` is the fitted object from `gSlc()`. The true f from which the data were generated has been added, and is shown as a dashed line. Right panel: As for the left panel, but for $\exp(f)$ instead of f. The data are shown as circles.

MCMC convergence, via the edf measure, can be checked using:

```
summary(fit2)
```

which, for data simulated above, leads to the plot shown in Figure 2. The convergence of edf is seen to be reasonable since the global line trend in the trace plot is flat, the lag 1 correlation is mild, and the autocorrelation spikes are well below 1. The posterior distribution of edf is seen to be centred about 10.5, which indicates a high degree of nonlinearity – consistent with the mean function from which the data were generated. Visualisations of the estimate of f are shown in Figure 3, and were obtained by issuing the commands:

```
plot(fit2)
plot(fit2, responseScale = TRUE)
points(x, y)
```

Setting `responseScale=TRUE` inside `plot()` transforms the estimate of f to the scale of the data – in this case $\exp(f)$, the inverse link function composed with f. Additional ad hoc R commands were used to add the true f curves as dashed lines, and the legends. The estimate of f from `gSlc()` is seen to be quite good in this case.

3.3. Semiparametric logistic regression

Our third example is the simple semiparametric logistic model:

$$y_i \overset{\text{ind}}{\sim} \text{Bernoulli}\left(\text{logit}^{-1}(\beta_{x1}x_{1i} + f(x_{2i}))\right), \quad 1 \leqslant i \leqslant n. \tag{7}$$

A Bayesian penalised spline formulation is:

$$\begin{aligned} &y_i | x_{1i}, x_{2i}, \beta_0, \beta_{x_1}, \beta_{x_2}, u_1, \ldots, u_K \overset{\text{ind}}{\sim} \\ &\text{Bernoulli}\left(\text{logit}^{-1}(\beta_0 + \beta_{x_1}x_{1i} + \beta_{x_2}x_{2i} + \sum_{k=1}^{K} u_k z_k(x_{2i}))\right), \\ &x_{1i} \overset{\text{ind}}{\sim} \text{Bernoulli}\left(\frac{1}{2}\right), \quad x_{2i} \overset{\text{ind}}{\sim} \text{Uniform}(0,1), \\ &u_k | \sigma^2 \overset{\text{ind}}{\sim} \text{N}(0, \sigma^2), \quad \beta_0, \beta_{x_1}, \beta_{x_2} \overset{\text{ind}}{\sim} \text{N}(0, 10^{10}), \quad \sigma \sim \text{Half-Cauchy}(10^5). \end{aligned} \tag{8}$$

Consider a data set simulated with $n = 500$ and $f(x) = \sin(2\pi x)$:

```
set.seed(981127)
n <- 500
betax1True <- 0.5
x1 <- sample(c(0, 1), n, replace=TRUE)
x2 <- runif(n)
mu <- 1/(1 + exp(- (betax1True * x1 + sin(2 * pi * x2))))
y <- rbinom(n, 1, mu)
```

Model (8) is fitted, and then summarised, in `gammSlice` using:

```
fit3 <- gSlc(y ~ x1 + s(x2), family = "binomial")
summary(fit3)
```

Figure 4 shows the result of the `summary(fit3)` call. MCMC convergence for the key β_{x_1} parameter is very good, and is reasonable for the effective degrees of freedom for the $f(x_2)$ component.

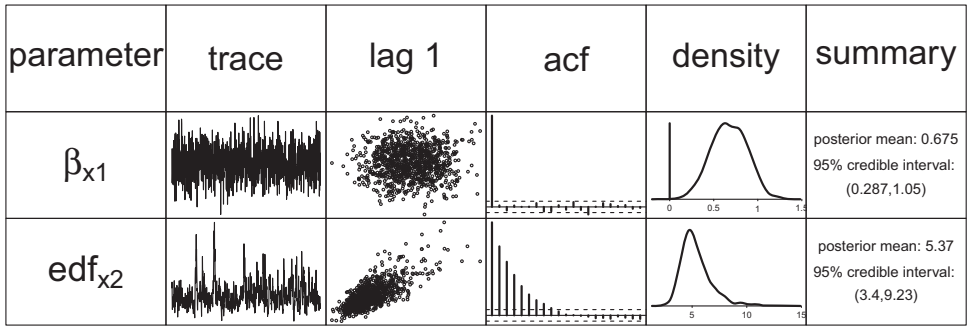

Figure 4. Summary of slice-based MCMC samples for fitting model (7), with Bayesian representation (8), in `gammSlice`. This plot was produced via the command `summary(fit3)` where `fit3` is the fitted object from a call to `gSlc()`. The columns are as described in the caption of Figure 1.

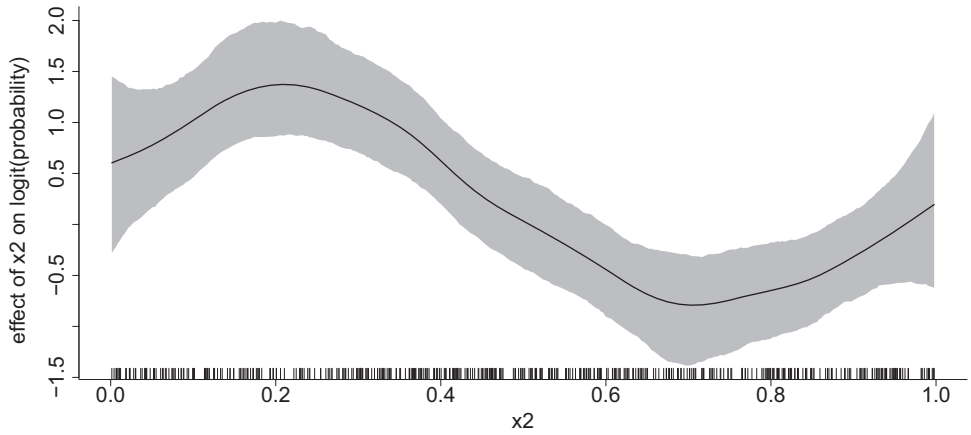

Figure 5. Bayesian posterior mean estimate of f (solid line) and pointwise 95% credible intervals (shaded region) obtained from `gSlc()`. This plot was produced via the command `plot(fit3)` where `fit3` is the fitted object from a call to `gSlc()`.

Lastly, the estimate of f can be viewed using:

```
plot(fit3)
```

This leads to Figure 5, which captures the sinusoidal nature of the true f function. Also note that `gammSlice` convention is to plot the fit for the predictor of interest with all other predictors set to their sample means. For Figure 5, the implication is that we are plotting the Bayes estimate of $\beta_{x1}\bar{x}_1 + f(x_2)$ against x_2, where $\bar{x}_1$ is the average of the x_{1i}s.

3.4. Poisson additive mixed model

The last example of simulated data is a Poisson additive mixed model

$$y_{ij} \overset{\text{ind}}{\sim} \text{Poisson}\Big(\exp\big(U_i + f_1(x_{1ij}) + f_2(x_{2ij})\big)\Big) \quad U_i \overset{\text{ind}}{\sim} \text{N}(0, \sigma^2). \tag{9}$$

The corresponding Bayesian penalised spline is as follows:

$$\begin{gathered} y_{ij}|x_{1ij}, x_{2ij}, \beta_0, \beta_{x_1}, u_{11}, \ldots, u_{1K_1}, u_{21}, \ldots, u_{2K_2} \overset{\text{ind}}{\sim} \\ \text{Poisson}\Big(\exp\Big(\beta_0 + U_i + \beta_{x_1}x_{1i} + \textstyle\sum_{k=1}^{K_1} u_{1k}z_{1k}(x_{1ij}) + \beta_{x_2}x_{2ij} + \sum_{k=1}^{K_2} u_{2k}z_{2k}(x_{2ij})\Big)\Big), \\ x_{1ij} \overset{\text{ind}}{\sim} \text{Uniform}(0,1), \quad x_{2ij} \overset{\text{ind}}{\sim} \text{Uniform}(0,1), \\ U_i \overset{\text{ind}}{\sim} \text{N}(0, \sigma^2), \quad u_{1k}|\sigma^2_{x_1} \overset{\text{ind}}{\sim} \text{N}(0, \sigma^2_{x_1}), \quad u_{2k}|\sigma^2_{x_2} \overset{\text{ind}}{\sim} \text{N}(0, \sigma^2_{x_2}), \\ \beta_0, \beta_{x_1}, \beta_{x_2} \overset{\text{ind}}{\sim} \text{N}(0, 10^{10}), \quad \sigma \sim \text{Half-Cauchy}(10^5), \\ \sigma_{x_1} \sim \text{Half-Cauchy}(10^5), \quad \sigma_{x_2} \sim \text{Half-Cauchy}(10^5) \end{gathered} \tag{10}$$

where

$$\{z_{1k}(\cdot) : 1 \leqslant k \leqslant K_1\} \quad \text{and} \quad \{z_{2k}(\cdot) : 1 \leqslant k \leqslant K_2\}$$

are basis functions over the range of the x_{1i} s and x_{2i} s, respectively.

Consider data generated from (9) with $m = 100, n = 10$, x_{1i} and x_{2i} independently and uniformly distributed on $(0, 1)$, $\sigma^2 = 1$ and the mean functions

$$f_1(x) = \cos(4\pi x) + 2x, \qquad f_2(x) = \sin(2\pi x^2):$$

```
set.seed(2966703)
m <- 100
n <- 10
x1 <- runif(m * n)
x2 <- runif(m * n)
idnum <- rep(1:m, each = n)
sigsqTrue <- 1
U <- rep(rnorm(m, 0, sqrt(sigsqTrue)), each = n)
mu <- exp(U + cos(4 * pi * x1) + 2 * x1 + sin(2 * pi * x2^2))
y <- rpois(m * n, mu)
```

To fit the model to the data set, we use:

```
fit4 <- gSlc(y~s(x1) + s(x2), random = list(idnum = ~1),
family = "poisson")
```

As before, summary and functional fit plots are obtained as follows:

```
summary(fit4)
plot(fit4)
```

Figure 6 shows the result of `summary(fit4)`. It indicates good convergence with the default MCMC sample sizes. The 95% credible set for σ^2 is $(0.535, 0.975)$, which is close to including the value of 1 from which the data were generated.

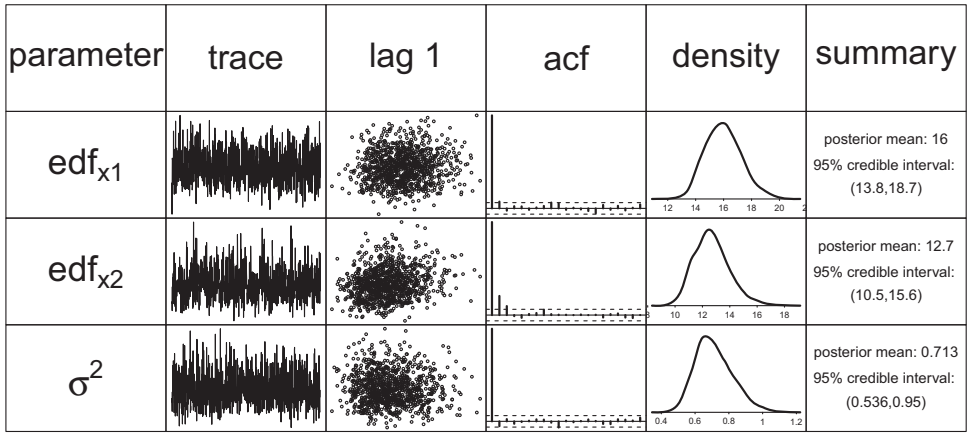

Figure 6. Summary of slice-based MCMC samples for fitting the Poisson additive model (9), with Bayesian representation (10), in `gammSlice`. This plot was produced via the command `summary(fit4)` where `fit4` is the fitted object from a call to `gSlc()`. The columns are as described in the caption of Figure 1.

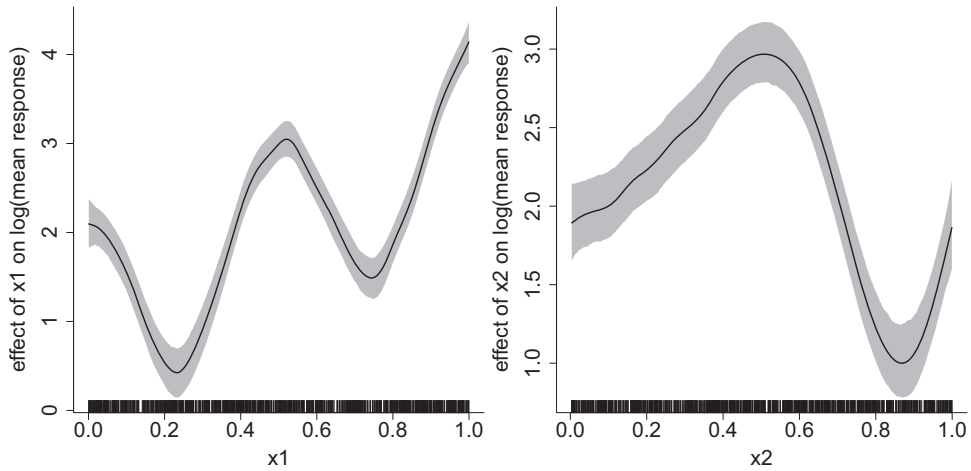

Figure 7. Bayesian posterior mean estimates of f_1 and f_2 (solid lines) and pointwise 95% credible intervals (shaded regions) obtained from `gSlc()` for the Poisson additive mixed model (9), with Bayesian representation (10). This plot was produced via the command `plot(fit4)` where `fit4` is the fitted object from a call to `gSlc()`.

The high edf values reflect the nonlinearity of the fitted functional components. The impression of these nonlinearity effects is reinforced by Figure 7.

4. Accuracy assessment

A small-scale simulation study was run to assess the accuracy of `gammSlice`, and to see how it compares with the faster Laplace approximation based approaches.

We generated 100 samples according to the GLMM

$$y_{ij} \overset{\text{ind}}{\sim} \text{Bernoulli}(\text{logit}^{-1}(\beta_0 + U_i)), \quad 1 \leqslant i \leqslant 250, \quad 1 \leqslant j \leqslant 2, \quad U_i \overset{\text{ind}}{\sim} \text{N}(0, \sigma^2) \tag{11}$$

with 'true values'

$$\beta_0 = 2 \quad \text{and} \quad \sigma = 1.$$

For each sample we obtained point estimates and 95% confidence intervals for β_0 and σ using the functions `glmmPQL()` and `intervals()` from the R package MASS (Venables & Ripley 2002) and the functions `glmer()` and `confint()` from the R package `lme4` (Bates *et al.* 2015). As its name suggests, `glmmPQL()` uses penalised quasi-likelihood (PQL) (e.g. Breslow & Clayton 1993) for approximate inference. However, `glmer()` uses a different version of Laplace approximation which is described in the vignettes of Bates *et al.* (2015). MCMC-based Bayes estimates and 95% credible sets were also obtained for each sample using `gSlc()`, and the results compared. Such a comparison could be criticised from a philosophical standpoint due to a mixture of Bayesian and frequentist inferential summaries being compared. However, since the Bayesian inference is based on diffuse priors, there is at least an informal sense in which the results are comparable.

Figure 8 provides a visual summary of the output from the simulation study. The PQL results indicate a pronounced positive bias for both β_0 and σ. This is especially the case for

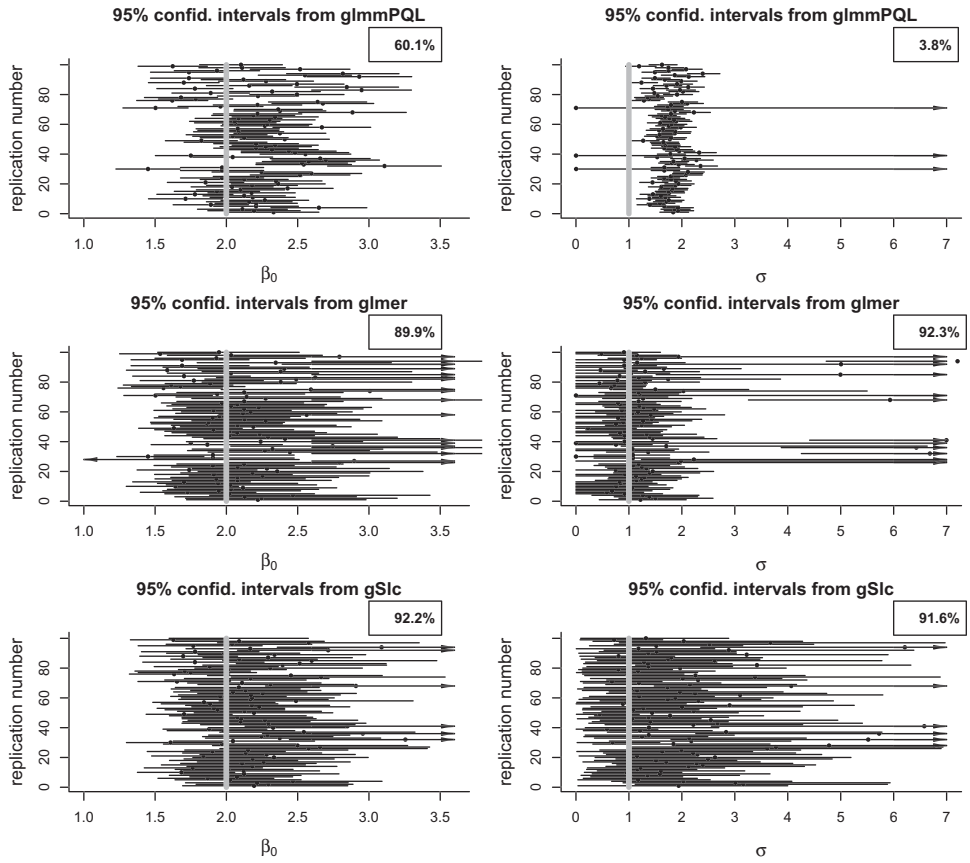

Figure 8. Point and interval estimates obtained from `glmmPQL()`, `glmer()` and `gSlc()` for β_0 (left panels) and σ (right panels) for 100 replications from (11). The true values of β_0 and σ are shown as thick grey vertical lines. For each parameter and each method, empirical coverage percentage of the 95% confidence/credible intervals based on 1000 replications is shown in the top right-hand corner of the relevant plot.

σ, where the empirical coverage of the advertised 95% approximate confidence intervals is seen to be abysmal. The Bayes estimates and 95% credible sets from `gammSlice` appear to have better coverage, although the Bayes estimates exhibit a tendency to overestimate the targets.

To better address the question of empirical coverage matching the nominal advertised coverage of 95%, we ran 900 more replications of the study that produced Figure 8 and kept track of which intervals included the true values out of the total of 1000 replications. The corresponding percentages are also displayed in Figure 8.

The empirical coverage values reaffirm the fact that the approximate confidence intervals produced by `glmmPQL()` can have very poor coverage. On the other hand, the empirical coverage of the credible sets produced by `glmer()` and `gSlc()` are seen to be quite close to the nominal level of 95%. Also note that the pronounced bias apparent in the PQL results is corrected by the better Laplace approximation used by `glmer()`.

Ideally similar accuracy comparisons would be performed for the GAMM extension, where smooth functions are present. However, the current releases of `mgcv` and `gamm4` do not fully support confidence intervals. Despite their limitations, these results indicate that, if accuracy is of utmost importance, `gammSlice` is a much better alternative to PQL-based inference.

5. Application to data arising from medical studies

We now illustrate the use of `gammSlice` for the analysis of actual data-sets, that have arisen from medical studies. Each data-set has appeared before in the GLMM literature, which allows some comparison with previously published results.

5.1. Toenail data

The *toenail data* arose from a longitudinal study reported in De Backer *et al.* (1998). The study is concerned with treatment of *onycholysis*, the separation of the toenail from its normal attachment to the nail bed. The response measurements are

$$\texttt{onycholysis}_{ij} = \begin{cases} 0 & \text{onycholysis absent or mild in patient i at visit } j \\ 1 & \text{onycholysis moderate to severe in patient i at visit } j \end{cases}$$

where $1 \leqslant i \leqslant 294$ and $1 \leqslant j \leqslant n_i$ where $n_i \in \{1,\ldots,7\}$ is the number of visits made by the ith patient.

Corresponding predictor variables are as follows:

$$\texttt{terb}_i = \begin{cases} 0 & \text{ith patient treated with } \textit{intraconazole} \\ 1 & \text{ith patient treated with } \textit{terbinafine} \end{cases}$$

and $\texttt{months}_{ij}$, the number of months since start of the treatment when the ith patient made his/her jth visit. Figure 9 shows $\texttt{onycholysis}_{ij}$ versus $\texttt{months}_{ij}$, with greyscale coding according to the treatment.

We considered the logistic mixed model

$$\begin{aligned} &\texttt{onycholysis}_{ij} | \beta_{\text{terb}}, \beta_{\text{months}}, \beta_{\text{terbXmonths}}, U_i \overset{\text{ind}}{\sim} \\ &\text{Bernoulli}\Big(\text{logit}^{-1}\big(\beta_0 + U_i + \beta_{\text{terb}}(\texttt{terb}_i) + \beta_{\text{months}}(\texttt{months}_{ij}) + \\ &\qquad + \beta_{\text{terbXmonths}}(\texttt{terb}_i \times \texttt{months}_{ij})\big)\Big), \qquad U_i \overset{\text{ind}}{\sim} \text{N}(0, \sigma^2). \end{aligned} \tag{12}$$

Slice sampling-based fitting of (12) and summarisation can be achieved via:

```
data("toenail")
terbXmonths <- toenail[,"terb"] * toenail[,"months"]
toenailPlus <- cbind(toenail, terbXmonths)
fitTN <- gSlc(onycholysis ~ terb + months + terbXmonths,
                            random = list(idnum = ~ 1),
                            data = toenailPlus,family = "binomial")
summary(fitTN)
```

The resulting summary plot is shown in Figure 10. This summary shows very good MCMC convergence for all key variables. The time effect, quantified by β_{months} is seen to be significant with a 95% credible set of $(-0.489, -0.313)$. This indicates a significant

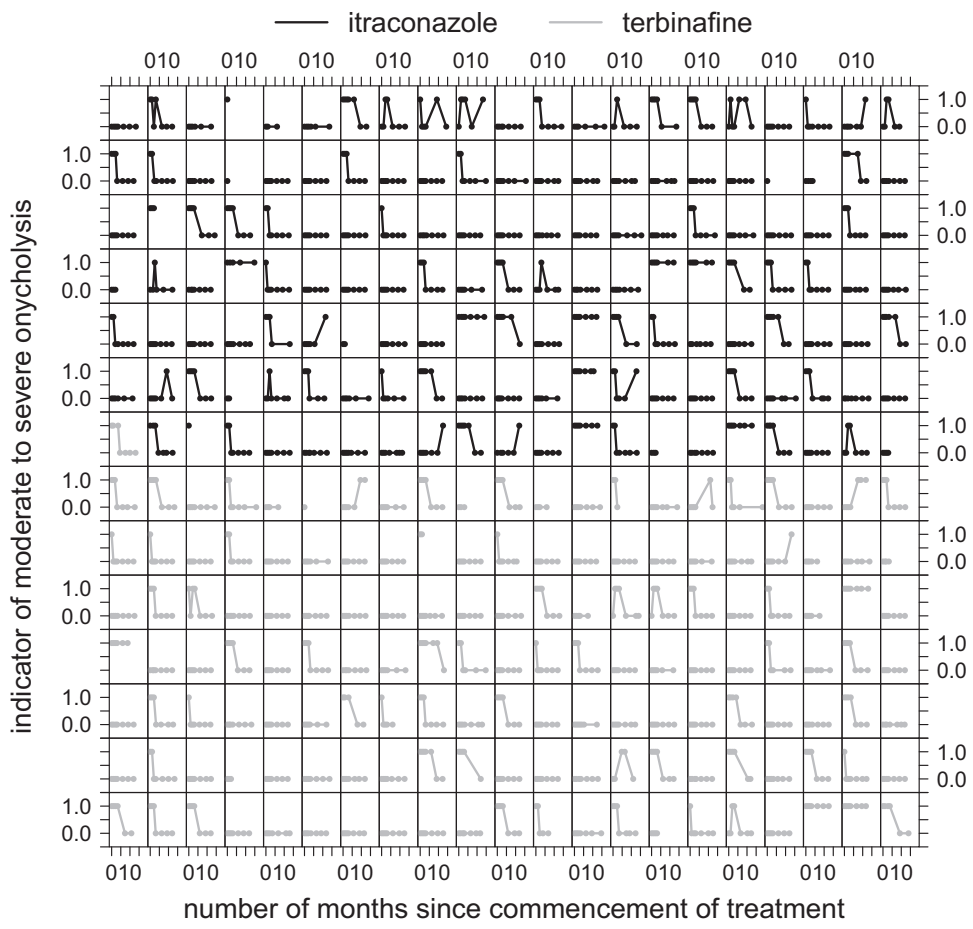

Figure 9. Plot of the toenail data, obtained using the R package `lattice` (Sarkar 2008).

decrease in prevalence of onycholysis over the course of the study. Similarly, $\beta_{\text{terbXmonths}}$ has a 95% credible set of $(-0.278, -0.00379)$. Hence there is evidence that terbinafine has a moderating effect over time in terms of reducing prevalence of onycholysis. The 95% credible set for σ^2 of $(12.4, 26)$ constitutes evidence of within-patient correlation.

5.1.1 Illustrative odds ratio summary

The odds ratio, conditional on each subject's random intercept, for

$$\texttt{terb} = 1 \quad \text{and} \quad \texttt{months} = \text{median of months values} = Q_2$$

versus

$$\texttt{terb} = 0 \quad \text{and} \quad \texttt{months} = 0$$

is $\text{OR}_{\text{terb}} = \exp(\beta_{\text{terb}} Q_2)$. Via calculations analogous to those given in Section 3.1.4, the posterior mean for OR_{terb} is found to be 0.767 and the 95% credible set to be $(1.87 \times 10^{-7}, 0.670)$. These results are consistent with the statistically significant interaction effect shown in Figures 10 and 11.

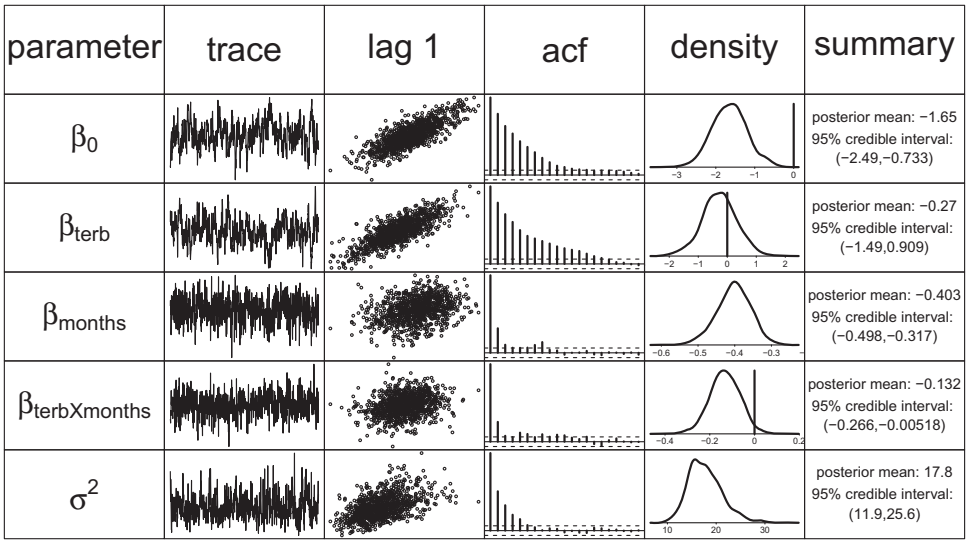

Figure 10. Summary of slice-based MCMC samples for fitting the logistic mixed model (12) fitted to the toenail data in `gammSlice`. This plot was produced via the command `summary(fitTN)` where `fitTN` is the fitted object from a call to `gSlc()`. The columns are as described in the caption of Figure 1.

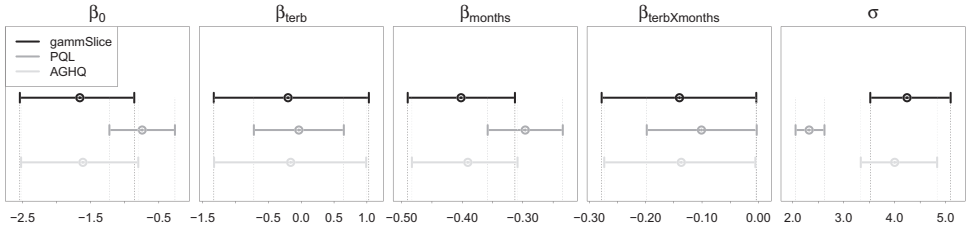

Figure 11. Comparison of point and interval estimates for each of the model parameters, based on `gammSlice`, PQL and adaptive Gauss-Hermite quadrature (AGHQ) fitting of (12) to the toenail data. For `gammSlice`, the point estimates correspond to posterior means and the interval estimates are 95% credible sets based on the MCMC output. For PQL, the point estimates are penalised maximum likelihood estimates and the interval estimates are approximate 95% confidence intervals based on PQL approximation. The PQL results were obtained via the function `glmmPQL()` from the R package `MASS`. The AGHQ results were obtained via the function `glmer()` from the R package `lme4`.

5.1.2 Comparison with penalised quasi-likelihood and exact maximum likelihood

Figure 11 provides a comparison of Bayes estimates and 95% credible sets from `gammSlice` with their penalised quasi-likelihood and exact likelihood counterparts. The penalised quasi-likelihood (PQL) fits were obtained using the function `glmmPQL()` from the R package MASS (Venables & Ripley 2002). The exact likelihood fits were obtained via the function `glmer()` from the R package `lme4` (Bates *et al.* 2015), which uses adaptive Gauss-Hermite quadrature (AGHQ) to achieve exactness. The potential philosophical criticisms of comparing frequentist and Bayesian inferential summaries, mentioned in Section 4, apply here as well. The diffuseness of the priors means that at least informal comparison is justifiable.

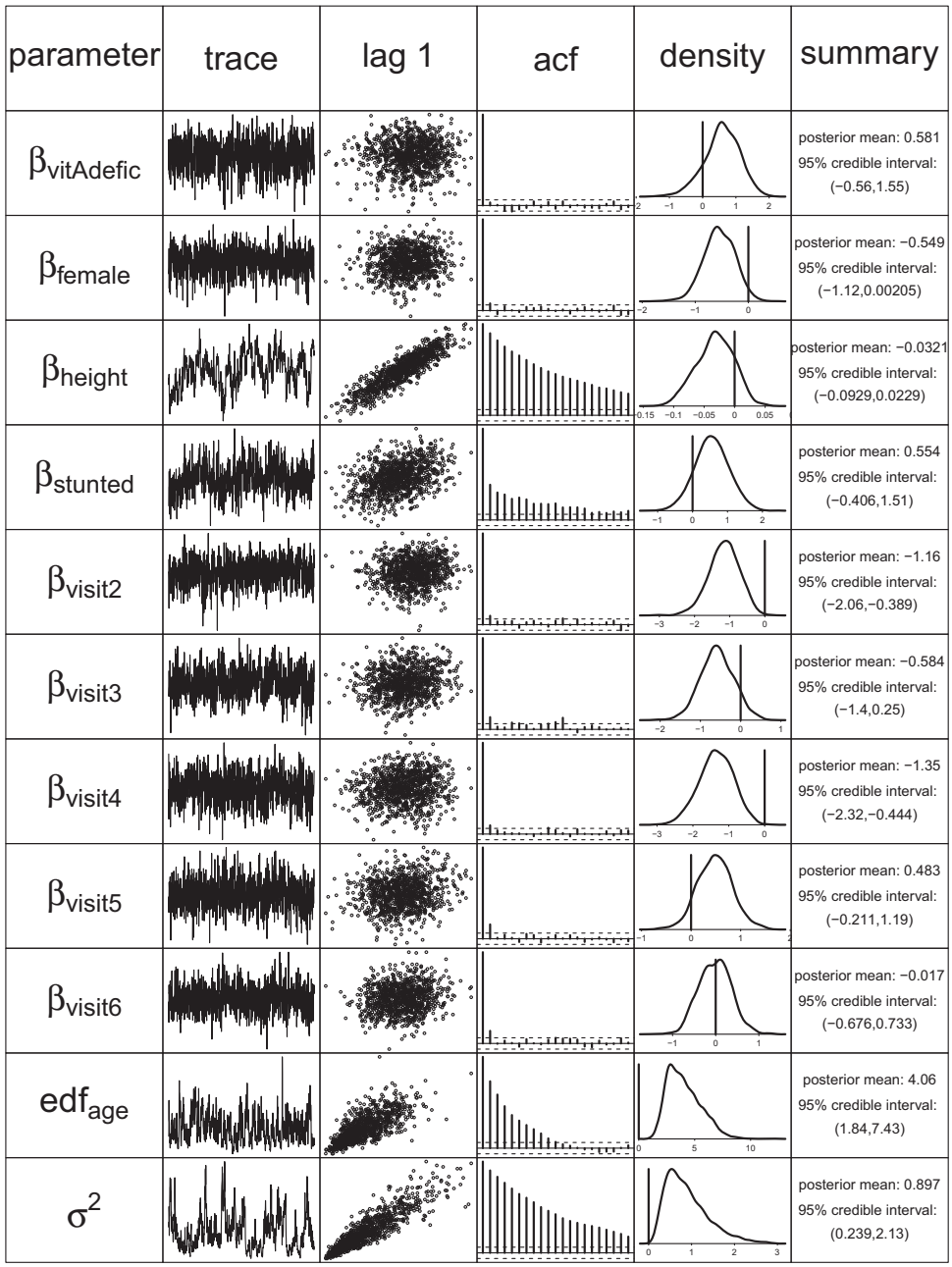

Figure 12. Summary of slice-based MCMC samples for fitting the logistic additive mixed model (13) to the toenail data in `gammSlice`. This plot was produced via the command `summary(fitIR)` where `fitIR` is the fitted object from a call to `gSlc()`. The columns are as described in the caption of Figure 1.

We see from Figure 11 that `gammSlice` and AGHQ give answers that are quite close to one another, providing further evidence of the accuracy of the slice sampling approach. PQL is shown to be quite inaccurate for some of the parameters, such as β_{months} and σ.

5.2. Indonesian Children's Health Study data

The Indonesian Children's Health Study data, obtained from a cohort of 275 Indonesian children, originates from Sommer (1982) and is also is described in Diggle *et al.* (2002). The response variable is respiratory infection (0=absent, 1=present) whilst potential predictors are age, indicator of vitamin A deficiency, sex, height, indicator of being stunted and the number of clinic visits for each child. Here, we work with indicators for the number of visits, and five such indicators are required. Let $\boldsymbol{x}_{ij}$ be the 9×1 vector containing values of these predictors where $1 \leqslant i \leqslant 275$ indexes child and $1 \leqslant j \leqslant n_i$ indexes the repeated measures for each child. An additive mixed model of interest is:

$$\begin{aligned} \text{logit}(P(\texttt{respiratory infection}_{ij} = 1)) &= U_i + \boldsymbol{\beta}^T \boldsymbol{x}_{ij} + f(\texttt{age}_{ij}), \\ U_i &\overset{\text{ind}}{\sim} \text{N}(0, \sigma^2). \end{aligned} \tag{13}$$

This model is fitted in `gammSlice` via:

```
data("indonRespir")
fitIR <- gSlc(respirInfec ~s(age) + vitAdefic + female + height
                          + stunted + visit2 + visit3 + visit4 +
                          visit5 + visit6,
                          random = list(idnum =~1),family
                          = "binomial",
                          data = indonRespir)
```

To see the summary we type:

```
summary(fitIR)
```

We get the summary shown in Figure 12. This is similar to figure 1 of Zhao *et al.* (2006) and shows that a diet rich in vitamin A and regular clinical visits tend to lower the risk of respiratory infection.

A plot showing the probability of respiratory infection as a function of age is shown in Figure 13, and is obtained using:

```
plot(fitIR, responseScale = TRUE)
```

Figure 13 indicates the interesting non-monotonic age effect that was also observed by, for example, Lin & Carroll (2001) and Zhao *et al.* (2006). A simpler model, such as a GLMM without the penalised spline component, would miss this non-linear age effect. Hence, the GAMM provides a sounder and more insightful analysis.

6. Extensions

As with any package of this type, numerous extensions of `gammSlice` could be contemplated. One of the more obvious of these extensions is the provision of additional distributional families for the response variable. Examples are gamma, negative binomial and generalised extreme value distributions. Another type of extension is to allow for the GAM and GAMM to have bivariate functions of pairs of continuous predictors included

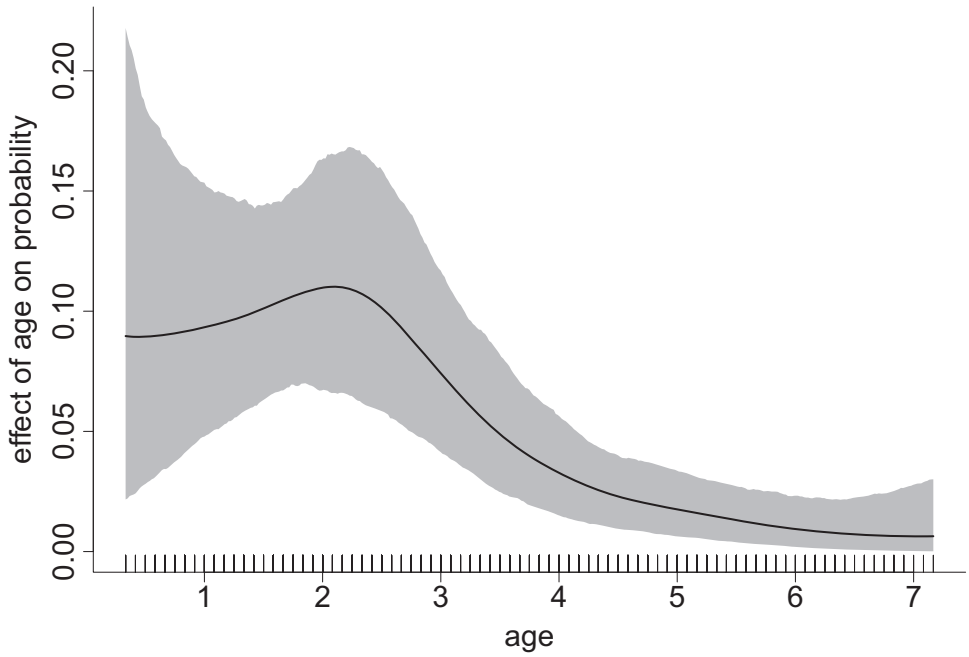

Figure 13. Bayesian posterior mean estimate of $f(age)$ (solid line) and pointwise 95% credible intervals (shaded region) obtained. This plot was produced via the command `plot(fitIR,responseScale=TRUE)` where `fitIR` is the fitted object from a call to `gSlc()`.

in the model. These are sometimes referred to a *geoadditive* models, and are included in the general design GLMM framework described in Zhao *et al.* (2006). The final extension that we would like to mention is one that would allow for vector random effect structures – appropriate for models having both random intercept and slopes. All of these extensions are relatively straightforward with respect to the `gammSlice` infrastructure. Time and resource constraints have placed the usual limitations on the current release of `gammSlice`.

Appendix I
Sampling Scheme Details

As mentioned in Section 1, a detailed mathematical description of the Bayesian GAMM supported by `gammSlice` is given in Zhao *et al.* (2006). However, the slice sampling scheme used by `gammSlice` is not discussed there. Hence we give some details here.

Models currently supported by `gammSlice` have the general form

$$p(\boldsymbol{y}|\boldsymbol{\beta},\boldsymbol{u}) \propto \exp(\boldsymbol{y}^T(\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{Z}\boldsymbol{u}) - \mathbf{1}^T b(\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{Z}\boldsymbol{u})),$$

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_L \end{bmatrix} \Bigg| \sigma_1^2,\ldots,\sigma_L^2 \sim N\left(\boldsymbol{0}, \begin{bmatrix} \sigma_1^2\boldsymbol{I}_{q_1} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_2^2\boldsymbol{I}_{q_2} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \sigma_L^2\boldsymbol{I}_{q_L} \end{bmatrix}\right), \tag{14}$$

$$\boldsymbol{\beta} \sim \mathrm{N}(\boldsymbol{0}, \sigma_{\boldsymbol{\beta}}^2\boldsymbol{I}), \quad \sigma_\ell \overset{\text{ind}}{\sim} \text{Half-Cauchy}(A_\ell),\ 1 \leqslant \ell \leqslant L.$$

The notation in (14) is as follows: $\boldsymbol{\beta}$ is the vector of all fixed effects parameters and $\boldsymbol{X}$ is the fixed effects design matrix, $\boldsymbol{u}$ is the vector of all random effects parameters and $\boldsymbol{Z}$ is the random effects design matrix, $\boldsymbol{I}_d$ denotes the $d \times d$ identity matrix, q_ℓ is the length of $\boldsymbol{u}_\ell$ and

$$b(x) = \begin{cases} \log(1+e^x), & \text{if the } y_i \text{ are Bernoulli} \\ e^x, & \text{if the } y_i \text{ are Poisson.} \end{cases}$$

Note that the $\boldsymbol{Z}$ matrix may contain both indicators of group membership and spline basis functions. For example, in the Poisson additive mixed model treated in Section 3.4 we have $\boldsymbol{Z}$ being the horizontal concatenation of $\boldsymbol{I}_m \otimes \mathbf{1}_n$, where $\mathbf{1}_n$ is the $n \times 1$ vector of ones, and the two spline basis design matrices containing the $z_{1k}(x_{1ij})$ and $z_{2k}(x_{2ij})$ values respectively.

In the context of model (14), MCMC sampling schemes benefit from the result

$$\sigma \sim \text{Half-Cauchy}(A) \quad \text{if and only if} \quad \begin{cases} \sigma^2 | a \sim \text{Inverse-Gamma}(1/2, 1/a), \\ a \sim \text{Inverse-Gamma}(1/2, 1/A^2), \end{cases} \tag{15}$$

where $x \sim \text{Inverse-Gamma}(\kappa, \lambda)$ means that x has an inverse gamma distribution with shape parameter κ and rate parameter λ. The inverse gamma density function is

$$p(x) = \lambda^\kappa \Gamma(\kappa)^{-1} x^{-\kappa-1} \exp(-\lambda/x), \quad x > 0.$$

Hence we replace $\sigma_\ell \overset{\text{ind}}{\sim} \text{Half-Cauchy}(A_\ell)$ in (15) by

$$\sigma_\ell^2 | a_\ell \overset{\text{ind}}{\sim} \text{Inverse-Gamma}(1/2, 1/a_\ell), \quad a_\ell \overset{\text{ind}}{\sim} \text{Inverse-Gamma}(1/2, 1/A_\ell^2).$$

Let

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{u} \end{bmatrix}, \quad \boldsymbol{C} = [\boldsymbol{X} \ \boldsymbol{Z}],$$

and let v_j denote the jth entry of $\boldsymbol{v}$ and $\boldsymbol{v}_{-j}$ denote the vector obtained from $\boldsymbol{v}$ when v_j is omitted. Similarly, define $\boldsymbol{c}_j$ to be the jth column of $\boldsymbol{C}$ and $\boldsymbol{C}_{-j}$ to be the $\boldsymbol{C}$ matrix with the jth column removed. Also, write

$$t | s_1, s_2, \boldsymbol{s}_3, \boldsymbol{s}_4 \sim \mathcal{H}_b(s_1, s_2, \boldsymbol{s}_3, \boldsymbol{s}_4)$$

to denote that the random variable t, conditional on $(s_1, s_2, \boldsymbol{s}_3, \boldsymbol{s}_4)$, has density function

$$p_b(t | s_1, s_2, \boldsymbol{s}_3, \boldsymbol{s}_4) \propto \exp\left(s_1 t - \frac{1}{2}(t^2/s_2) - \mathbf{1}^T b(t\boldsymbol{s}_3 + s_4)\right), \quad -\infty < t < \infty. \tag{16}$$

Let $\mathbf{1}_d$ denote the $d \times 1$ vector of ones, p be the number of columns in $\boldsymbol{X}$ and $q = \sum_{\ell=1} q_\ell$ be the number of columns in $\boldsymbol{Z}$. Finally, let B and G be positive integers that specify the size of the burn-in and the number of samples to be retained for inference. The Gibbs sampling scheme algorithm is:

Initialise $\boldsymbol{v}^{[0]} \in \mathbb{R}^P$, $(\sigma_\ell^2)^{[0]} > 0$, $1 \leqslant \ell \leqslant L$.
  For $g = 1,\ldots,B+G$:
    $\boldsymbol{v}^{[g]} \leftarrow [\sigma_{\boldsymbol{\beta}}^2 \mathbf{1}_p^T, (\sigma_1^2)^{[g-1]}\mathbf{1}_{q_1}^T, \ldots, (\sigma_L^2)^{[g-1]}\mathbf{1}_{q_L}^T]^T$
    For $j = 1,\ldots,p+q$:
      $v_j^{[g]} \sim \mathcal{H}_b((\boldsymbol{C}^T\boldsymbol{y})_j, (\boldsymbol{v}^{[g-1]})_j, \boldsymbol{c}_j, \boldsymbol{C}_{-j}\boldsymbol{v}_{-j}^{[g-1]})$,
    For $\ell = 1,\ldots,L$:
      $a_\ell^{[g]} \sim \text{Inverse-Gamma}\left(1, 1/(\sigma_\ell^2)^{[g-1]} + A_\ell^{-2}\right)$,
      $(\sigma_\ell^2)^{[g]} \sim \text{Inverse-Gamma}\left(\frac{1}{2}(q_\ell+1), \frac{1}{2}\|\boldsymbol{u}_\ell^{[g]}\| + 1/a_\ell^{[g]}\right)$
Return $\boldsymbol{v}^{[g]}$ and $(\sigma_\ell^2)^{[g]} > 0$, $1 \leqslant \ell \leqslant L$ for $B+1 \leqslant g \leqslant B+G$.

Assuming that B is large enough for an acceptable level of convergence, the retained samples can be treated as draws from the respective posterior distributions, and used for approximate Bayesian inference.

It remains to describe sampling from $\mathcal{H}_b(s_1, s_2, \boldsymbol{s}_3, \boldsymbol{s}_4)$ for general arguments $s_1, s_2, \boldsymbol{s}_3, \boldsymbol{s}_4$. Since the corresponding density (16) has a simple closed form expression up to the normalising factor it follows that slice sampling can be used obtain the required draws. Neal (2003) describes easy-to-implement procedures and his 'stepping out' method is the default used in `gammSlice`.

This simplicity of slice sampling makes the implementation quite straightforward and robust. However, naïve implementations can be inefficient due to the sparse forms that routinely arise in the $\boldsymbol{Z}$ matrix. In `gammSlice` we have been careful to avoid superfluous calculations arising from these sparse forms.

References

BALGTAGI, B.H. (2013). *Econometric Analysis of Panel Data*. Chichester, U.K.: John Wiley & Sons, 5th edn.

BATES, D., MÄCHLER, M., BOLKER, B. & WALKER, S. (2015). Fitting linear mixed-effects models using `lme4`. *Journal of Statistical Software* **67**, 1–48.

BRESLOW, N.E. & CLAYTON, D. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* **88**, 9–25.

BRESLOW, N.E. & LIN, X. (1995). Bias correction in generalised linear mixed models with a single component of dispersion. *Biometrika* **82**, 81–91.

BROSTRÖM, G. & HOLMBERG, H. (2018). `glmmML`: Binary and Poisson regression for clustered data, fixed and random effects with bootstrapping. R package version 1.0.3. Available from URL: http://cran.r-project.org/package=glmmML.

BROWN, P. & ZHOU, L. (2018). `glmmBUGS`: Write BUGS model files for hierarchical and spatial models, arranges unbalanced data in ragged arrays, and creates starting values. R package version 2.4.2. Available from URL: http://cran.r-project.org/package=glmmBUGS.

DE BACKER, M., DE VROEY, C., LESAFFRE, E., SCHEYS, I. & DE KEYSER, P. (1998). Twelve weeks of continuous oral therapy for toenail onychomycosis caused by dermatophytes: a double-blind comparative trial of terbinafine 250 mg/day versus itraconazole 200 mg/day. *Journal of the American Academy of Dermatology* **38**, S57–S63.

DIGGLE, P.J., HEAGERTY, P., LIANG, K. & ZEGER, S. (2002). *Analysis of Longitudinal Data*, 2nd edn. Oxford: Oxford University Press.

FITZMAURICE, G., DAVIDIAN, M., VERBEKE, G. & MOLENBERGHS G. (editors) (2008). *Longitudinal Data Analysis*. Boca Raton, FL: Chapman & Hall/CRC.

GOLDSTEIN (2010). *Multilevel Statistical Models*, 4th edn. Chichester, UK: John Wiley & Sons.

HADFIELD, J. (2017). `MCMCglmm`: MCMC generalised linear mixed models. R package version 2.25. Available from URL: http://cran.r-project.org/package=MCMCglmm.

HASTIE, T.J. & TIBSHIRANI, R.J. (1990). *Generalized Additive Models*. London: Chapman & Hall.

LIN, X. & CARROLL, R.J. (2001). Semiparametric regression for clustered data. *Biometrika* **88**, 1179–1185.

NEAL, R. (2003). Slice sampling (with discussion). *The Annals of Statistics* **31**, 705–767.

PHAM, T.H. & WAND, M.P. (2018). `gammSlice`: Generalized additive mixed model analysis via slice sampling. R package version 2.0-1. Available from URL: http://cran.r-project.org/package=gammSlice.

R DEVELOPMENT CORE TEAM (2018). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. Available from URL: http://www.R-project.org/.

SARKAR, D. (2008). *Lattice: Multivariate Data Visualisation with R*. New York: Springer. Available from URL: http://lmdvr.r-forge.r-project.org. ISBN 978-0-387-75968-5.

SCHEIPL, F. (2011). `spikeSlabGAM`: Bayesian variable selection, model choice and regularization for generalized additive mixed models in R. *Journal of Statistical Software* **43**, 1–24. Available from URL: http://www.jstatsoft.org/v43/i14/.

SOMMER, A. (1982). *Nutritional Blindness*. New York: Oxford University Press.

UMLAUF, N., KNEIB, T., LANG, S. & ZEILEIS, A. (2017). `R2BayesX`: An R interface to estimate structured additive regression models with BayesX. R package version 1.1. Available from URL: http://cran.r-project.org/package=R2BayesX.

VENABLES, W.N. & RIPLEY, B.D. (2002). *Modern Applied Statistics with S*, 4th edn. New York: Springer. Available from URL: http://www.stats.ox.ac.uk/pub/MASS4. ISBN 0-387-95457-0.

WAND, M.P. & ORMEROD, J.T. (2008). On semiparametric regression with O'Sullivan penalised splines. *Australian and New Zealand Journal of Statistics* **50**, 179–198.

WOOD, S. (2017). *Generalized Additive Models: An Introduction with R*, 2nd edn. Boca Raton, FL: Chapman and Hall/CRC.

WOOD, S.N. & SCHEIPL, F. (2017). `gamm4`: Generalized additive mixed models using `mgcv` and `lme4`. R package version 0.2. Available from URL: http://cran.r-project.org/package=gamm4.

ZHAO, Y., STAUDENMAYER, J., COULL, B.A. & WAND, M.P. (2006). General design Bayesian generalized linear mixed models. *Statistical Science* **21**, 35–51.