TEST (2020) 29:307–339
https://doi.org/10.1007/s11749-020-00711-5

INVITED PAPER

# Inference and computation with generalized additive models and their extensions

Simon N. Wood[1]

Published online: 23 April 2020

**Abstract**
Regression models in which a response variable is related to smooth functions of some predictor variables are popular as a result of their appealing balance between flexibility and interpretability. Since the original generalized additive models of Hastie and Tibshirani (Generalized additive models. Chapman & Hall, Boca Raton, 1990) numerous model extensions have been proposed, and a variety of practically useful computational strategies have emerged. This paper provides an overview of some widely applicable frameworks for this type of modelling, emphasizing the similarities between the different approaches, and the equivalence of smoothing, Gaussian latent process models and Gaussian random effects. The focus is particularly on Bayes empirical smoother theory, fully Bayesian inference via stochastic simulation or integrated nested Laplace approximation and boosting.

**Keywords** Smoothing · Regression · Smoothing parameters · INLA · Boosting · Empirical bayes · Reduced rank

**Mathematics Subject Classification** 62J05 · 62J07 · 62J12

## 1 Introduction

Since Hastie and Tibshirani (1986, 1990) combined generalized linear models with the smoothing methods developed in the 1970s and 1980s (see especially Wahba 1990)

This invited paper is discussed in comments available at: https://doi.org/10.1007/s11749-020-00713-3, https://doi.org/10.1007/s11749-020-00714-2, https://doi.org/10.1007/s11749-020-00715-1.

**Electronic supplementary material** The online version of this article (https://doi.org/10.1007/s11749-020-00711-5) contains supplementary material, which is available to authorized users.

✉ Simon N. Wood
simon.wood@bristol.ac.uk

[1] School of Mathematics, University of Bristol, Bristol, UK

to produce the *generalized additive model*, there has been a great deal of activity extending these models and developing alternative computational approaches to their use. The original GAM was

$$y_i \sim \text{EF}(\mu_i, \phi) \text{ where } g(\mu_i) = \mathbf{A}_i \boldsymbol{\gamma} + \sum_j f_j(x_{ji}), \tag{1}$$

$y_i$ is a univariate response variable, $\text{EF}(\mu_i, \phi)$ denotes an exponential family distribution with mean $\mu_i$ and scale parameter $\phi$, $\mathbf{A}_i$ is the $i$th row of a parametric model matrix, $\boldsymbol{\gamma}$ are regression parameters, the $f_j$ are smooth functions to be estimated, and $x_j$ is a covariate (usually, but not necessarily, univariate). The original model fitting method involved estimating the $f_j$ by iterative smoothing of partial residuals w.r.t. the $x_j$: the *backfitting* algorithm. It was soon realized that models beyond the exponential family, multivariate models and models with multiple linear predictors could also be estimated, with Yee and Wild (1996) providing a pioneering reference (see Yee 2015, for an overview) and further impetus provided by Rigby and Stasinopoulos (2005) and Stasinopoulos et al. (2007, 2017).

Parallel to the backfitting developments was a recognition that the full practical benefits promised by allowing flexible dependence on covariates can only be fully realized if the degree of smoothing of the $f_j$ can be estimated as part of model fitting. The first practical methods for multiple smoothing parameter estimation were provided by Gu and Wahba (1991) and Gu (1992), but these had $O(n^3)$ computational cost ($n = \dim(\mathbf{y})$). By representing the $f_j$ using reduced rank spline smoothers, as suggested in Wahba (1980) and Parker and Rice (1985), Wood (2000) provided a much more efficient smoothing parameter estimation method. Meanwhile, Fahrmeir and Lang (2001) exploited the sparse reduced rank P-splines of Eilers and Marx (1996) for stochastic simulation-based inference with GAMs, while the reduced rank penalized spline approach of Ruppert et al. (2003) employed mixed model fitting ideas, in which smoothing parameters are treated as variance parameters. The Bayesian and mixed model approaches exploit a duality between spline smoothing and Gaussian random effects identified in Kimeldorf and Wahba (1970) and made particularly clear by Silverman (1985).

Once sound methods had been developed (and subsequently refined) for inference with GAMs, including inference about the smoothness of the component $f_j$, it was only a matter of time before these methods were also extended to wider classes of model: beyond univariate exponential family models to essentially any regular likelihood and to models in which any or all parameters of a likelihood might depend on separate sums of smooth functions of covariates (GAMLSS or ‘distributional regression’ models). See for example Belitz et al. (2015), Klein et al. (2015), Lang et al. (2014), Mayr et al. (2012), Umlauf et al. (2015), Wood et al. (2016), Wood and Fasiolo (2017). At the same time alternative computational methods were developed, most notably boosting (Schmid and Hothorn 2008) and the simulation-free approach to Bayesian inference, *integrated nested Laplace approximation* (INLA, Rue et al. 2009, 2017). The latter allows efficient inference without requiring low rank representations of smooths, thereby facilitating improved modelling of short-range correlation. There was also work on the modelling of smooth interactions and multidimensional smoothing of various sorts of spatial and spatiotemporal data and on allowing linear functionals of smooth functions in models.

The purpose of this paper is to provide an overview (albeit scandalously skewed to my own work) of the theory and computational methods for working with these general smooth regression models, emphasizing that the different computational strategies are using essentially the same modelling framework, based on the correspondence between smoothing and latent Gaussian random field models (and indeed simple Gaussian random effects), and the fact that 'smoothing' can be induced by an appropriate choice of Gaussian prior. Similarly, most inference with such models can be viewed as Bayesian or empirical Bayesian, albeit with some results suggesting good frequentist properties, and access to some frequentist tools such as AIC and approximate p values. Much of what is discussed here is available in the R package `mgcv`, and code for the examples is supplied as supplementary material.

## 2 Statistical function estimation

The key statistical concepts for modelling with smooth functions are most easily explained in the context of a one-dimensional model for smoothing a response variable $y$ with respect to a predictor variable (covariate), $x$. Let $y_i$ be modelled as an observation of a random variable with probability (density) function $\pi(y_i|\mu_i, \boldsymbol{\theta})$ where $\mu_i$ is a location parameter (e.g. $\mathbb{E}(y_i)$) and $\boldsymbol{\theta}$ a vector of other parameters of the likelihood (e.g. the dispersion parameter of a negative binomial). The interesting part of our model states that $\mu_i$ is an unknown function of $x_i$,

$$\mu_i = f(x_i) \text{ or } g(\mu_i) = f(x_i) \text{ for } i = 1, \ldots, n, \tag{2}$$

where $g$ is an (optional) known smooth monotonic link function, useful for keeping $\mu_i$ within some pre-defined range (such as $(0, 1)$ or $(0, \infty)$). Assuming that the $y_i$ are independent, given $x_i$, the log-likelihood function for such a model is $l(f, \boldsymbol{\theta}) = \sum_i \log \pi(y_i|\mu_i, \boldsymbol{\theta})$. But without further structure $\hat{f} = \text{argmax}_f l(f, \boldsymbol{\theta})$ is not unique. Any $f$ corresponding to each $\mu_i$ maximizing $\pi(y_i|\mu_i, \boldsymbol{\theta})$ would have equal likelihood and $f$ is free to do anything in between $x_i$ values.

To obtain uniqueness of $\hat{f}$ requires more structure. Let us assume that $f$ is *smooth*. To make this precise we need a mathematical characterization of smoothness. One possibility is

A function $f$ is smoother than a function $g$ if $\int f''(x)^2 \mathrm{d}x < \int g''(x)^2 \mathrm{d}x$.

There are many alternatives to the integrated squared second derivative, or *cubic spline*, penalty, $\int f''(x)^2 \mathrm{d}x$, which all lead to essentially the same mathematical structure, so we lose nothing by sticking with this one for the moment. Notice how the penalty will be high for a very wiggly curve, but is zero if $f$ is any linear function of $x$.

We could now remove the ambiguity in $\hat{f}$ by picking the minimizer of $\int f''(x)^2 \mathrm{d}x$ among the maximizers of the log likelihood, but in most cases (with distinct $x_i$ values) the resulting model would then interpolate the $x_i$, $y_i$ data so that $\hat{\mu}_i = y_i$. Interpolation is rarely a desirable outcome of statistical modelling, since it amounts to 'fitting the noise' as well as the signal. We need stronger smoothness restrictions on $f$. These can be obtained by penalizing the log likelihood using the smoothing penalty, so that we seek

$$\hat{f}, \hat{\boldsymbol{\theta}} = \underset{f,\theta}{\operatorname{argmax}}\, l(f, \boldsymbol{\theta}) - \frac{\lambda}{2} \int f''(x)^2 \mathrm{d}x \tag{3}$$

where $\lambda$ is now a parameter controlling the balance between smoothness and model fit in estimation (the 2 is convenient later). The addition of the penalty automatically means that we select the candidate $\hat{f}$ that is smoothest in between the $x_i$ values, so the initial lack of identifiability has gone. But from the theory of inequality constrained optimization you can also see that imposing the penalty is equivalent to putting some upper bound on the value of $\int f''(x)^2 \mathrm{d}x$ allowed in the solution, with $\lambda/2$ playing the role of a Lagrange multiplier. This means that we no longer interpolate the data, and the larger $\lambda$ is, the more heavily we smooth it.

How do we get from infinite-dimensional functional optimization problem (3) to something computable? It turns out that if $\mathcal{N} \le n$ is the number of unique $x_i$, then the solution to (3) has the form

$$\hat{f}(x) = \sum_{j=1}^{\mathcal{N}} \beta_j b_j(x)$$

where the $\beta_j$ are coefficients to be chosen to maximize (3), but the basis functions, $b_j(x)$, have known fixed form, which does not depend on $\lambda$. In consequence we can express the $n$ vector of evaluated function values, $f(x_i)$, as $\mathbf{f} = \mathbf{X}\boldsymbol{\beta}$ where $X_{ij} = b_j(x_i)$, and hence the log likelihood, $l$, can be expressed as a function of the unknown coefficients $\boldsymbol{\beta}$. The basis functions are of course not unique: if $\mathbf{A}$ is any rank $\mathcal{N}$ matrix, then the functions $a_j(x) = \sum_k A_{jk} b_k(x)$ form an equally valid basis, and in fact the analytic forms of several such alternatives are known.

From the known $b_j(x)$ it also follows that $\int f''(x)^2 \mathrm{d}x = \boldsymbol{\beta}^T \mathbf{S} \boldsymbol{\beta}$ where the elements of matrix $\mathbf{S}$ are fixed and known. To see this let $\mathbf{b}(x)$ and $\mathbf{b}''(x)$ denote the vectors of basis functions, and second derivates of basis functions, evaluated at $x$. So $f(x) = \boldsymbol{\beta}^T \mathbf{b}(x)$ and hence $f''(x) = \boldsymbol{\beta}^T \mathbf{b}''(x)$. It follows that $f''(x)^2 = \boldsymbol{\beta}^T \mathbf{b}''(x) \mathbf{b}''(x)^T \boldsymbol{\beta}$ and so $\int f''(x)^2 \mathrm{d}x = \boldsymbol{\beta}^T \int \mathbf{b}''(x) \mathbf{b}''(x)^T \mathrm{d}x \boldsymbol{\beta} = \boldsymbol{\beta}^T \mathbf{S} \boldsymbol{\beta}$, where $S_{ij} = \int b_i''(x) b_j''(x) \mathrm{d}x$.

So estimation problem (3) becomes the readily computable (see Sect. 5.1)

$$\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\beta},\theta}{\operatorname{argmax}}\, l(\boldsymbol{\beta}, \boldsymbol{\theta}) - \frac{\lambda}{2} \boldsymbol{\beta}^T \mathbf{S} \boldsymbol{\beta}. \tag{4}$$

## 2.1 Reduced rank representation of smooth functions

Having reduced the infinite-dimensional optimization to an $n$ (+ dim($\boldsymbol{\theta}$))-dimensional optimization is a step forward, but will generally entail $O(n^3)$ computational cost.[1]

[1] Actually there are cheaper algorithms when we have only one smooth term, but these do not apply once we have more than one smooth term in a model.

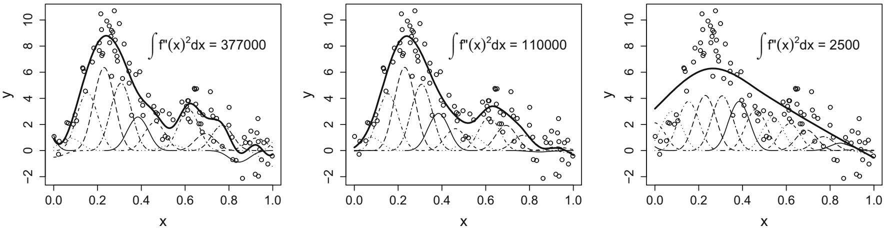

**Fig. 1** Smoothing with a reduced rank ($K = 16$) cubic spline basis. Data are open circles. The spline fit is the thick black curve, which is the sum of B-spline basis functions, $b_j(x)$, each multiplied by an estimated coefficient, $\beta_j$, illustrated as thin dashed curves. The value of the cubic spline penalty is shown for each of the 3 increasingly smooth fits

Do we really need $n$ (or $\mathcal{N}$) coefficients? To answer this we need to consider the two sources of error in estimating $f$. The first is the error entailed by approximating $f$ using a (cubic) spline basis: even if we observed $f$ without error at the $x_i$ values and just interpolated the resulting data, between the data we would have an error proportional to the 4th power of the spacing between adjacent $x_i$ values (de Boor 2001). For evenly spaced $x_i$ (or any reasonably behaved infill process generating $x_i$) this corresponds to an approximation error of $O(n^{-4})$: this is the rate that we have to expect for the estimation bias. The second error is the regular statistical estimation error, which cannot be better than $O(n^{-1/2})$—so, clearly we have considerable scope for allowing the bias to increase before it becomes significant relative to the sampling uncertainty.

To exploit this observation, we could decide to pick $K$ evenly spaced $x_i$ values from our full set of $n$ and compute the $K$ cubic spline basis functions that would have been obtained if these were all the data points we had. We can then use this reduced set of basis functions to represent $f$ as $f(x) = \sum_{j=1}^{K} \beta_j b_j(x)$ when modelling our full data set. The approximation error/bias is now $O(K^{-4})$, while the sampling error is at worst $O(\sqrt{K/n})$ (it could be of lower order depending on penalization). This suggests setting $K = n^{1/9}$ if we want to minimize the overall error and not have the bias or sampling error dominating at a worse rate asymptotically. A more careful consideration of the situation under penalization (e.g. Claeskens et al. 2009) actually suggests setting $K = O(n^{1/5})$, but this does not alter the main point, which is that from a statistical perspective we are not gaining anything useful by using $n$ basis functions (and hence coefficients), and we might as well use far fewer. If we do this, the cost of solving (4) typically drops to $O(nK^2)$. Figure 1 shows a rank 16 basis used to smooth 100 data.

### 2.1.1 Eigen-based rank reduction

Rather than simply picking $K$ 'nice' values of $x_i$ from which to compute basis functions, we could seek the $K$ basis functions that are 'best' in some sense. This idea leads to reduced rank eigen bases. One general possibility is to form the full basis and then to form the QR decomposition $\mathbf{X} = \mathbf{QR}$ followed by the eigen decomposition $\mathbf{UDU}^T = \mathbf{R}^{-1}\mathbf{SR}^{-T}$. The reparameterization $\tilde{\boldsymbol{\beta}} = \mathbf{U}^T\mathbf{R}\boldsymbol{\beta}$ corresponds to setting the penalty matrix $\mathbf{S}$ to the diagonal eigenvalue matrix $\mathbf{D}$ and the basis function matrix $\mathbf{X}$ to $\mathbf{QU}$. The columns of $\mathbf{QU}$ are now interpretable as the evaluated basis functions

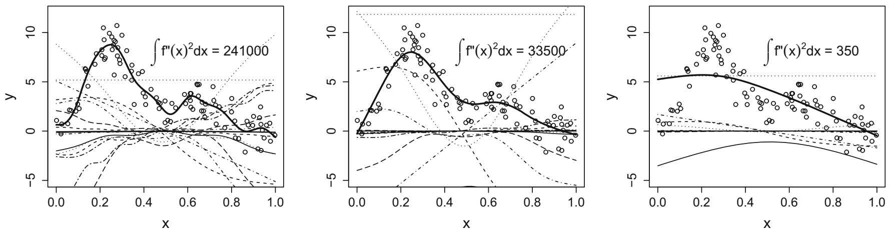

**Fig. 2** As Fig. 1, but using a rank 16 eigen basis. Notice how the scaled basis functions ($\beta_j b_j(x)$—thin curves) now have an ordering from smooth to wiggly and are no longer compactly supported translations of each other. Also notice how all basis functions are involved in the wiggly fit on the left, but as the penalization increases (so that the value of the penalty decreases), the more wiggly basis functions are shrunk towards zero

under reparameterization, and if the diagonal elements of $\mathbf{D}$ are arranged in order of decreasing magnitude, then the columns of $\mathbf{QU}$ will be arranged in order of decreasing wiggliness, since they represent decreasingly heavily penalized components of $f$. Hence to obtain a reduced rank basis, we can simply retain the final $K$ columns of $\mathbf{QU}$ and rows and columns of $\mathbf{D}$, which is equivalent to setting all but the last $K$ elements of $\tilde{\boldsymbol{\beta}}$ to zero (see, e.g. Wood 2017a, section 5.4.2 for a fuller discussion). While simple and general, the disadvantage of this approach is that it has an $O(n^3)$ set-up cost for the matrix decompositions. However, for some choices of basis function, an almost equivalent optimal approximation can be based solely on a truncated eigen decomposition of the $\mathbf{S}$ matrix, which can be computed at $O(n^2K)$ computational cost using Lanczos methods (Wood 2003). When $n$ is large these eigen approximations are usually combined with $x_i$ selection: for example, a size $n_r$ random sample of the original $x_i$ values is selected, and a spline basis is computed for this which is then used as the basis for obtaining a rank $K$ basis by eigen methods. The idea is that $n \gg n_r \gg K$. Figure 2 illustrates such a basis.

### 2.1.2 P-splines and all that

The idea of rank reduced smoothing goes back at least as far as Wahba (1980) and Parker and Rice (1985) and in the GAM context is discussed in Hastie and Tibshirani (1990), but it was given renewed impetus by Eilers and Marx (1996) and Ruppert et al. (2003) who provided alternative (but closely related) spline like reduced rank smoothers that had the advantage of being very easy to set up. In the Eilers and Marx (1996) case they also had the singular advantage of providing sparse bases and penalties,[2] facilitating computational efficiency in the context of Bayesian computation. The Eilers and Marx (1996) idea is to use a 'B-spline basis' (e.g. de Boor 2001) such as that illustrated in Fig. 1, but replace the associated derivative-based penalty with a difference penalty on the model coefficients, such as $\sum_j(\beta_{j+1} - 2\beta_j + \beta_{j-1})^2$ (one is free to choose the order of difference in the penalty). The simplicity of implementing this approach has led to a wide range of applications (see Eilers et al. 2015). Actually,

[2] That is bases and penalties yielding model matrices and penalty matrices with a high proportion of zero entries.

sparse penalties and the ability to freely choose the penalty order are also readily available when using derivative-based penalties (Wood 2017b) although implementation requires slightly more code. The Ruppert et al. (2003) approach used the truncated power basis for splines, with a simple ridge penalty on the coefficients of the truncated basis functions. The advantage of this is that it makes for very easy fitting using standard mixed modelling software.

### 2.2 Further inference about smooth functions

How can we estimate the smoothing parameter, $\lambda$, or make inferences about $f$ beyond simple point estimation? A Bayesian view of the smoothing penalty helps. It only makes sense to penalize a particular definition of smoothness if we believe that correspondingly smooth functions are somehow more probable than wiggly ones. A Bayesian prior formalizes this:

$$\pi(\boldsymbol{\beta}|\lambda) \propto \exp(-\boldsymbol{\beta}^T \mathbf{S}_\lambda \boldsymbol{\beta}/2),$$

which is immediately recognizable as an improper Gaussian prior on $\boldsymbol{\beta}$ with mean $\mathbf{0}$ and precision matrix $\mathbf{S}_\lambda$: here $\mathbf{S}_\lambda = \lambda \mathbf{S}$, but it will be generalized later. The prior is improper because $\mathbf{S}_\lambda$ is rank deficient by the dimension of the penalty null space (the dimension of the space of functions with zero penalty: 2 for the cubic spline penalty). Combining this prior with our model likelihood, the objective function in (4) is immediately recognizable as the log joint density of $\mathbf{y}$ and $\boldsymbol{\beta}$ (to within an additive constant). Hence $\hat{\boldsymbol{\beta}}$ is the posterior mode, or MAP estimate.

Given a prior and likelihood we can apply Bayes theorem to get a posterior distribution $\pi(\boldsymbol{\beta}|\mathbf{y}, \lambda)$. This only has closed form when the likelihood is Gaussian, but a simple Taylor expansion about $\hat{\boldsymbol{\beta}}$ shows that for arbitrary $\lambda$, a Fisher regular likelihood with suitably bounded second and third derivatives and $K = o(n^{1/3})$ (see e.g. Wood et al. 2016, §B.4)

$$\boldsymbol{\beta}|\mathbf{y}, \lambda \sim N(\hat{\boldsymbol{\beta}}, (\hat{\mathcal{I}} + \mathbf{S}_\lambda)^{-1}), \tag{5}$$

in the $n \to \infty$ limit, where $\hat{\mathcal{I}}$ is the observed information matrix (Hessian of the negative log likelihood) at $\hat{\boldsymbol{\beta}}$. This result is particularly useful, since it requires only quantities that we would anyway have to compute in order to maximize (4) by Newton's method. Of course if the model is such that $K$ growing at less than $n^{1/3}$ is not a tenable assumption, then we would have to use a higher-order approximation or MCMC for inference about $\boldsymbol{\beta}$ (see Sect. 5.2).

(5) is useful for computing credible intervals for any function of $\boldsymbol{\beta}$. For nonlinear functions we simulate replicate $\boldsymbol{\beta}$ vectors from (5) and compute the corresponding function of each replicate. For linear functions, such as the smooth itself, no simulation is necessary, because such functions have a directly computable Gaussian distribution. For example, confidence intervals for $f(x)$ can be computed and have remarkably good frequentist coverage properties, provided we consider average coverage, over the range of observed $x$ values (the intervals may over or undercover pointwise at

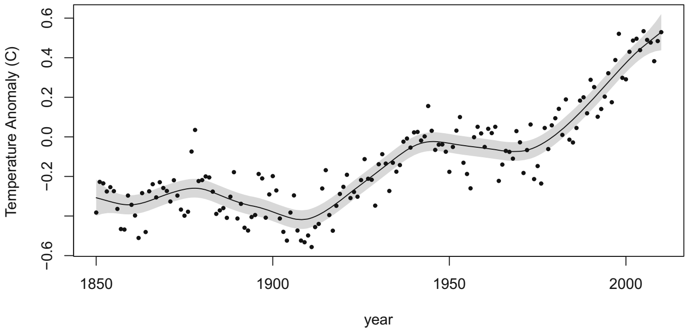

**Fig. 3** Global annual mean temperature anomalies plotted against year and smoothed with a rank 50 spline basis using a cubic spline penalty, with smoothing parameter estimated by marginal likelihood (REML) maximization. The grey band is a 95% credible interval computed using (5). The smooth function estimate has 11.7 effective degrees of freedom

particular $x$ values, but not when averaged over all $x$ values). Nychka (1988) provides explanation of why this occurs (or see Wood 2017a, section 6.10.1) and why the interval performance is rather robust to the choice of smoothing parameter value.

The smoothness prior view also facilitates the empirical Bayes approach of estimating $\lambda$ (finding its MAP estimate under a flat prior) as the maximizer of the marginal likelihood

$$\pi(\mathbf{y}|\lambda) = \int \pi(\mathbf{y}|\boldsymbol{\beta}, \lambda)\pi(\boldsymbol{\beta}|\lambda)d\boldsymbol{\beta}.$$

Notice how the marginal likelihood can be interpreted as the average likelihood of random draws from the prior $\pi(\boldsymbol{\beta}|\lambda)$. We are choosing $\lambda$ so that random draws from the prior have the right level of smoothness to get close enough to the data to have reasonably high likelihood. Except for the Gaussian likelihood case the integral is intractable. But since $\pi(\mathbf{y}|\lambda) = \pi(\mathbf{y}|\boldsymbol{\beta}, \lambda)\pi(\boldsymbol{\beta}|\lambda)/\pi(\boldsymbol{\beta}|\mathbf{y}, \lambda)$, then when (5) is valid we can use the approximation

$$\pi(\mathbf{y}|\lambda) \simeq \pi_L(\mathbf{y}|\lambda) = \frac{\pi(\mathbf{y}|\hat{\boldsymbol{\beta}}, \lambda)\pi(\hat{\boldsymbol{\beta}}|\lambda)}{\pi_G(\hat{\boldsymbol{\beta}}|\mathbf{y}, \lambda)} \tag{6}$$

where $\pi_G(\boldsymbol{\beta}|\mathbf{y}, \lambda)$ denotes the p.d.f. of Gaussian approximation (5). In fact this approximation is identically a first-order Laplace approximation (see e.g. Wood 2015, section 5.3.1) to the marginal likelihood integral (a proper prior can also be placed on $\log \lambda$ if needed).

In summary, having obtained a basis and chosen a penalty for $f$, we can estimate the smoothing parameter, $\lambda$ to maximize (6), while the model coefficient estimates/posterior modes given $\lambda$ are obtained from (4). Bayesian credible intervals for $\boldsymbol{\beta}$ and hence $f$ can be obtained using (5). Figure 3 shows a reduced rank spline computed in this way to smooth the global temperature series. Section 5 gives computational details alongside fully Bayesian alternatives. Notice how general the inferential machinery is here. The Gaussian smoothing prior gives our smooth model the structure of a Gaussian random effect/field or a latent Gaussian process model or a Gaussian process regression model: these are different terms for essentially the same thing.

The Bayesian view of spline smoothing dates back to Kimeldorf and Wahba (1970), with marginal likelihood being used for $\lambda$ estimation in Anderssen and Bloomfield (1974), but the real impetus came with Wahba (1983), Wahba (1985) and Silverman (1985). More recent linkage of mixed models and smoothing is really a rediscovery of the same ideas.

### 2.3 Other aspects of inference

Marginal likelihood is not the only approach to estimating $\lambda$. Cross-validation chooses $\lambda$ to maximize the average model probability of each $y_i$, when that $y_i$ was omitted from the fit, i.e.

$$l^{\text{cv}}(\lambda) = \sum_i \log \pi(y_i|\hat{\mu}^{[-i]}, \lambda)$$

is maximized, where $\hat{\mu}^{[-i]}$ is the estimate of $\mu_i$ when $y_i$ is omitted from the fitting data. Computationally efficient (and invariant) approximations to $l_{cv}$ give rise to generalized cross-validation (GCV Craven and Wahba 1979) and generalized approximate cross-validation (e.g. Gu 1992; Wood 2008). Another approach is to obtain an AIC (Akaike 1973) like criterion, by developing an estimate of the KL divergence of the model from the true model, accounting for penalization, and choosing $\lambda$ to minimize this. Attempting to find a computable approximation to $l^{\text{cv}}$ or an appropriate AIC lead to essentially the same criterion to minimize

$$\text{AIC} = -2l(\hat{\boldsymbol{\beta}}) + 2\tau \text{ where } \tau = \text{trace}\{(\hat{\mathcal{I}} + \mathbf{S}_\lambda)^{-1}\hat{\mathcal{I}}\}.$$

Since $\tau$ takes the role of the number of parameters in AIC, it is natural to interpret it as the *effective degrees of freedom* of the model. It is easy to see that the maximum value of $\tau$ is $K$ when $\lambda = 0$ and with slightly more effort that as $\lambda \to \infty$, $\tau \to 2$ (the dimension of the null space of $\mathbf{S}$). In between it take intermediate values. A more detailed consideration of the eigen approximation considered in Sect. 2.1 suggests that in general there is always[3] a reduced rank eigen basis of dimension close to $\tau$ that will yield un-penalized estimates having very similar statistical behaviour to the penalized estimates with EDF, $\tau$. So in that sense the characterization is reasonable. Another characterization of $\tau$ is as the number of coefficients multiplied by their average shrinkage as a result of penalization. Anticipating Sect. 3.1, if we sum up the elements of $\text{diag}\{(\hat{\mathcal{I}} + \mathbf{S}_\lambda)^{-1}\hat{\mathcal{I}}\}$ corresponding to one smooth, then we obtain its term-specific effective degrees of freedom. The smooth in Fig. 3 has $\tau = 11.7$.

AIC can be used for model selection in the usual way, but for optimal model selection behaviour it is necessary to correct $\tau$ for $\lambda$ estimation uncertainty (see Greven and

[3] Note that while this applies to smooth function estimates, it does not apply to general Gaussian random effects where there is no covariate ordering the observations.

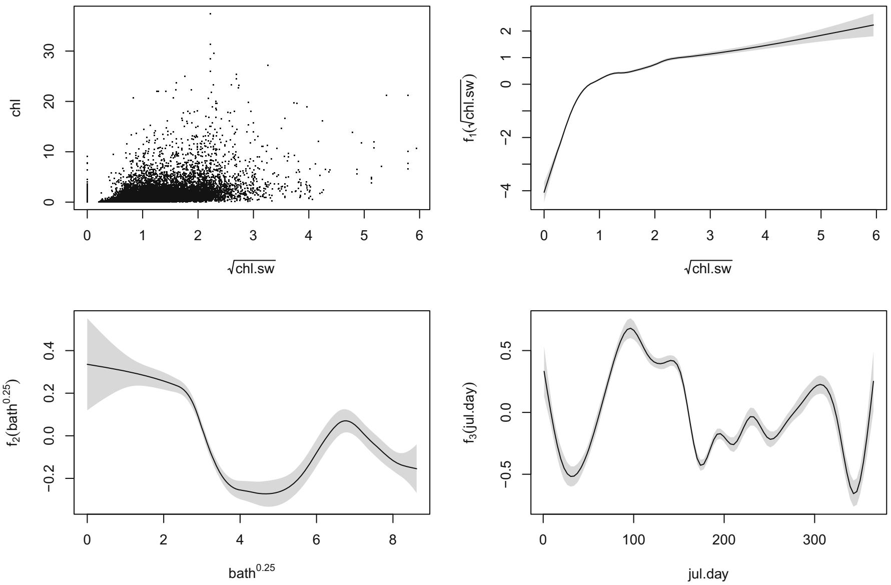

**Fig. 4** Additive model example. Top right: relationship between directly measured and remote-sensed oceanic chlorophyll. Remaining panels: estimates for the model $\texttt{chl} \sim$ Tweedie with log mean given by $\beta_0 + f_1(\sqrt{\texttt{chl.sw}}) + f_2(\texttt{bath}^{1/4}) + f_3(\texttt{jul.dat}) + \mathbb{I}(\texttt{chl.sw} = 0)\beta_1$, i.e. by a smooth additive dependence on remote-sensed chlorophyll, sea depth and day of year plus a parameter for when the satellite reading is zero. Transformations avoid excessive leverage. The relationship between direct and satellite measurements is strongly seasonally modulated and varies sharply between continental shelf and oceanic sea bed depths: see Clarke et al. (2006)

Kneib 2010; Wood et al. 2016). Another approach to model selection is to develop p values for the hypothesis $f(x) = 0$. It is again necessary to carefully account for penalization in order to obtain reasonable approximations, but this is also possible (see Wood 2013a, b).

## 3 Smooth regression in general

A wide array of models fit within the basic framework of a Fisher regular likelihood and basis expansions with quadratic smoothing penalties/Gaussian smoothing priors.

### 3.1 Extension I: additive in several smooth functions

An immediate extension of one-dimensional smoothing model (2) is to leave the distributional assumptions for $y_i$ unchanged, but to allow the location parameter to depend additively on several smooth functions of predictors, $x_j$, and possibly on some parametric effects, so that

$$g(\mu_i) = \eta_i \text{ where } \eta_i = \mathbf{A}_i \boldsymbol{\gamma} + \sum_j f_j(x_{ji}), \tag{7}$$

$\mathbf{A}_i$ is the $i$th row of a parametric model matrix with parameters $\boldsymbol{\gamma}$ and $\eta_i$ is known as the *linear predictor* of the model. This extension (essential the generalized additive model of Hastie and Tibshirani 1986, 1990) is easily accommodated by replacing each smooth function $f_j$ with a basis expansion, and associating a smoothing penalty with it, exactly as in the single smooth case. Figure 4 shows an example model calibrating satellite chlorophyll measurements.

The only new issue that we now need to deal with is identifiability: the $f_j$ are only identifiable to within an intercept term. To remove the ambiguity requires a linear constraint on each $f_j$. To obtain minimum width confidence intervals for the constrained smooth functions, sum-to-zero constraints are generally used.[4] For a single smooth, $f(x)$, with basis matrix, $\mathbf{X}$, and coefficients, $\boldsymbol{\beta}$, the constraint is $\sum_{i=1}^{n} f(x_i) = 0$ or equivalently $\mathbf{1}^T\mathbf{X}\boldsymbol{\beta} = 0$. As an identifiability constraint it can be imposed either by reparameterizing to absorb the constraint into $\mathbf{X}$ and $\mathbf{S}$ or by adding an extra quadratic penalty to the penalized likelihood during fitting: $\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{1}\mathbf{1}^T\mathbf{X}\boldsymbol{\beta}$ (since the penalty is merely removing the lack of identifiability there is no 'smoothing parameter' associated with it). Absorbing the constraint requires some routine book-keeping when subsequently predicting from the model (see e.g. Wood 2017a, §5.4.1).[5]

Everything else follows as in the one-dimensional case. The $f_j$ are replaced by their basis expansions, so that for the whole model we end up with $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$ where $\mathbf{X}$ contains $\mathbf{A}$ and the evaluated basis functions for each $f_j$ in successive blocks of columns. Similarly $\boldsymbol{\beta}$ contains $\boldsymbol{\gamma}$ and the coefficients for the different $f_j$ terms ($\boldsymbol{\theta}$ still denotes extra likelihood parameters). The fitting problem then becomes

$$\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\beta},\theta}{\operatorname{argmax}}\, l(\boldsymbol{\beta}, \boldsymbol{\theta}) - \frac{1}{2}\sum_j \lambda_j \boldsymbol{\beta}^T\mathbf{S}_j\boldsymbol{\beta}, \tag{8}$$

which only differs from the one-dimensional case in having a penalty/precision matrix made up of a sum of terms: i.e. $\mathbf{S}_\lambda = \sum_j \lambda_j \mathbf{S}_j$ (here the individual $\mathbf{S}_j$ are zero except for a nonzero block on the diagonal, corresponding to the coefficients for $f_j$). Hence inference proceeds exactly as in the one-dimensional case: the move to several smoothing parameters may complicate computation, but introduces nothing new to the statistical framework, beyond the fact that when interpreting smooth terms we have to bear in mind the sum-to-zero identifiability constraints.

### 3.2 Extension II: beyond one-dimensional splines

The preceding general framework applies equally well when some components are smooth functions of several variables, and when the quadratically penalized basis expansions represent something other than a spline. For example, any Gaussian random effect can be represented as some model matrix columns and a quadratic penalty/Gaussian prior. Hence such terms can be added to a linear predictor just like any smooth (giving *generalized additive mixed models* in the exponential family case).

[4] the literature contains many examples of using other slightly simpler to implement constraints, often accompanied by tremendously wide confidence intervals on the smooth effects

[5] See Rue and Held (2005, §2.3.3) for how to maintain basis sparsity with a sum to zero constraint

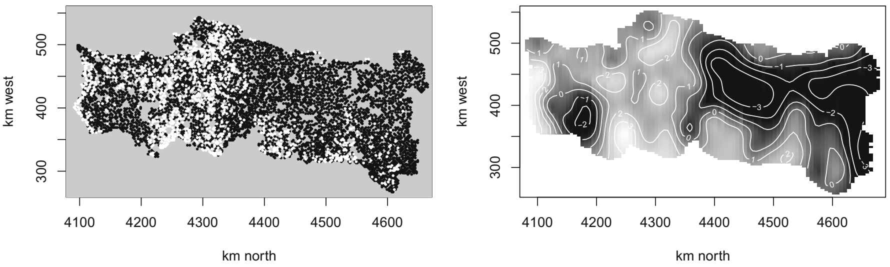

**Fig. 5** Isotropic spline example. Left: presence (white) or absence (black) of Crested Lark in survey quadrats in Portugal. Right: logit of the probability of the presence of Crested Lark in a quadrat, from a logistic regression model in which logit of probability is represented by a reduced rank Duchon spline of spatial location

The only extension that this might involve is that in general the precision matrix might not be of the form $\sum_j \lambda_j \mathbf{S}_j$ that fits directly into the framework discussed so far, and we might have to consider a nonlinear parameterization of the precision matrix. In a similar vein, Gaussian Markov random fields can also be represented as a set of (sparse) model matrix columns and a sparse precision matrix and the same is true for a variety of Gaussian process regression models.

The generalization of spline smoothing to several dimensions is closely bound up with how we define smoothness, with the best known generalization being the *thin plate spline* functional of Duchon (1977). Consider a smooth function of two covariates, $f(x, z)$. Letting subscripts denote differentiation w.r.t. a variable, the thin plate spline smoothing penalty is

$$\int f_{xx}(x, z)^2 + 2f_{xz}(x, z)^2 + f_{zz}(x, z)^2 \mathrm{d}x\mathrm{d}z.$$

As in the one-dimensional case, the optimizer of the penalized likelihood maximization problem with this penalty turns out to have a finite-dimensional representation in terms of $n$ (or $\mathcal{N}$) known basis functions. We can also generalize to more than 2 covariates, although there is a technical nuisance that the spline only exists if the order of differentiation in the penalty increases with the number of covariates (rapidly leading to inconveniently large penalty null space dimensions). Duchon's original paper actually eliminated this nuisance with a modification of the penalty: the resulting splines are as straightforward to compute with as the thin plate splines and are generally preferable in higher dimensions. As in the single covariate case, the use of $n$ basis functions is excessive from a statistical perspective, and expensive computationally, so rank reduction is used. Since the selection of a 'nice' set of $K$ covariate points becomes awkward beyond one dimension, the eigen approximations are particularly appealing. Figure 5 shows a Duchon spline, with first derivative penalization, modelling probability of presence of Crested Lark in Portugal, as a function of spatial location, in a logistic regression.

An obvious feature of the thin plate and other Duchon splines is isotropy. The penalty is invariant to rotations of the covariate space: we are choosing to treat smoothness in all directions equally. This is often not appropriate. Consider smoothing with respect to a distance and time. The thin plate spline treats squared second derivative with respect to time and squared second derivative with respect to distance 'equally', but simply changing the measurement units for time and distance we will change these quantities—their relative magnitude is arbitrary.

For smooths of covariates with no natural relative scaling, a nonisotropic construction is preferable. This can be achieved by applying the usual notion of a statistical interaction to smooth functions: that is, the effect of one covariate is itself modified by another covariate. In a parametric model the coefficients describing the relationship between the response and a covariate vary with another covariate. This translates directly to the smooth function case. For example, consider the basis expansion for a term $f(z) = \sum_j \alpha_j a_j(z)$, where the $\alpha_j$ are coefficients and the $a_j(z)$ are known basis functions. Suppose that we want a smooth interaction between $z$ and another covariate $x$. All that is needed is for the coefficients, $\alpha_j$, to become smooth functions of $x$, which we can do using a second basis expansion $\alpha_j(x) = \sum_k \beta_{jk} b_k(x)$. Substituting back in to the original expansion yields $f(x, z) = \sum_{j,k} \beta_{jk} a_j(z) b_k(x)$.

Now consider how the model matrix columns for the interaction relate to those of the corresponding main effect. Let **A** and **B** denote the marginal model matrices with elements $A_{ij} = a_j(z_i)$ and $B_{ij} = b_j(x_i)$. Then the model matrix columns, **X**, for the interaction are given by the elementwise products of all possible pairings of columns of **A** with columns of **B**. That is the row-tensor-product or row-Kronecker-product of **A** and **B**. This is *exactly* the same as the way that any interaction of two main effects is produced in a linear model.

Smoothing penalties are not a standard part of any interaction, and their set-up requires some care. Firstly, we need to avoid the arbitrary scale sensitivity of the isotropic smoothers.[6] The way to do this is to have a separate penalty corresponding to each marginal smooth. For example, if we would use a penalty $\int f''(z)^2 \mathrm{d}z$ for smoothing with respect to $z$ in one dimension, then it makes sense to use an average of the same penalty applied over the whole of $f(x, z)$. We would then produce a similar penalty for smoothing with respect to $x$ and allow each penalty to have its own smoothing parameter. The separate smoothing parameters allow the smooth to be invariant to covariate scaling—any rescaling is effectively absorbed by the smoothing parameters. There are a number of ways to produce such penalties that are completely automatic, given a basis and penalty for each marginal smooth: see Wood (2017a, §5.6) for more. Figure 6 illustrates the tensor product basis and penalty construction for a two-dimensional case.

The basis and penalty constructions generalize directly to $> 2$ covariates and to using isotropic smoothers as marginals—an appealing construction for spatiotemporal modelling.

### 3.2.1 Smooth ANOVA

Another standard feature of statistical modelling with interactions is the desire to separate additive 'main effects' from pure interactions, excluding the main effects. In

[6] And the pseudoinsensitivity that occurs by ad hoc measures like transforming all covariates to the unit interval.

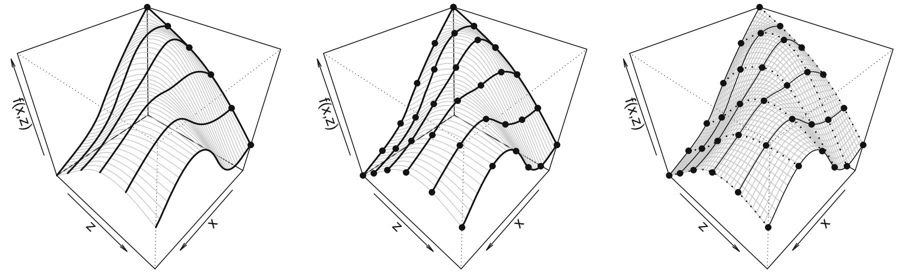

**Fig. 6** Tensor product smooth construction example. Left: A smooth function of $f(z)$ is represented using a spline parameterized in terms of function values at 6 equally spaced knots. These parameters are then allowed to vary smoothly with $x$ to create a smooth function of $x$ and $z$. Middle: the smooth variation of the parameters of $f(z)$ is facilitated by representing each using a spline of $x$. The construction is in fact symmetric in $x$ and $z$. Right: separate smoothness penalties in the $z$ and $x$ directions ensure scale invariance. We can construct an $x$ penalty by summing $\int f_{xx}^2 \mathrm{d}x$ along the thin black curves. A separate $z$ penalty is then constructed by summing the $\int f_{zz}^2 \mathrm{d}z$ along the dotted curves

standard parametric regression modelling this is achieved automatically by applying identifiability constraints to the main effect model matrix columns, before using them to construct the model matrix columns for the interaction. This is unchanged when an effect is represented using a basis expansion. For example (letting $f$s with different arguments be different functions), consider a model term

$$\alpha + f(x) + f(z) + f(x, z)$$

If we apply sum-to-zero constraints to the basis expansions for $f(x)$ and $f(z)$ *before* we construct the interaction term $f(x, z)$, then we automatically exclude functions of the form $f(x) + f(z)$ from the basis for the interaction. This is because the constraints have eliminated the constant function from the bases for $f(x)$ and $f(z)$, and without the constant function in the basis for $f(x)$ the interaction term will not contain a copy of the basis for $f(z)$, while the absence of the constant function from the $f(z)$ basis similarly puts pay to the copy of the $f(x)$ basis that would otherwise occur in the interaction. Since the constant functions are in the null space of any sensible smoothing penalty, their elimination does not change the penalty.

Hence the construction of ‘smooth-ANOVA’ models involves nothing new beyond standard regression modelling (and generalizes immediately beyond 2 covariates). Interpretation of the models is slightly different however, since a different penalty is generally assumed for the main effects + interaction model, as opposed to the interaction model. For example, a smooth term $\tilde{f}(x, z)$ (without constraints on the marginals) would have the same basis as a smooth term $f(x) + f(z) + f(x, z)$ (where the marginals are constrained before constructing the interaction). But the smoothing penalties are, for example

$$\lambda_1 \int \tilde{f}_{xx}(x, z)^2 \mathrm{d}x\mathrm{d}z + \lambda_2 \int \tilde{f}_{zz}(x, z)^2 \mathrm{d}x\mathrm{d}z$$

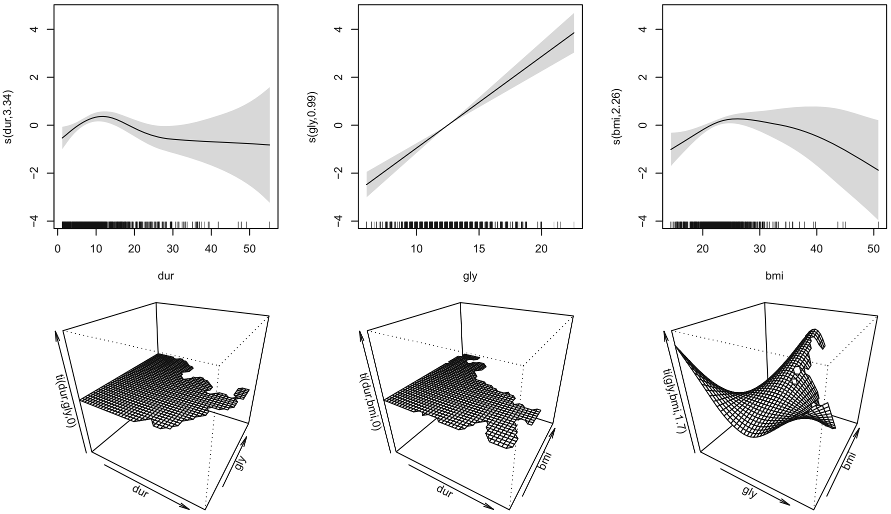

**Fig. 7** Smooth ANOVA logistic regression model example. The model is $\text{logit}(p) = f_1(\texttt{dur}) + f_2(\texttt{gly}) + f_3(\texttt{bmi}) + f_4(\texttt{dur}, \texttt{gly}) + f_5(\texttt{dur}, \texttt{bmi}) + f_6(\texttt{bmi}, \texttt{gly})$ where $p_i$ is probability of retinopathy in a cohort of diabetics; `dur`, `gly` and `bmi` are duration of disease, percent glycosylated haemoglobin and body mass index. Smoothing parameters estimated by marginal likelihood maximization. Top row: estimated main effects. Bottom row: estimated interactions (excluding main effects). Data originally from the `gss` package of Gu (2013)

versus the clearly different,

$$\lambda_1 \int f_{xx}(x)^2 \mathrm{d}x + \lambda_2 \int f_{zz}(z)^2 \mathrm{d}x + \lambda_3 \int f_{xx}(x, z)^2 \mathrm{d}x\mathrm{d}z + \lambda_4 \int f_{zz}(x, z)^2 \mathrm{d}x\mathrm{d}z,$$

Other penalty constructions are possible in which no such difference occurs, but these have less interpretable penalties. See Fig. 7 and Gu (2013) for a complete treatment of such models.

### 3.3 Extension III: linear functionals of smooth functions

Another extension allows linear functionals of smooths as model components (Wahba 1990). An example is ‘signal regression’ (e.g. Marx and Eilers 2005) where a spectrum or other measured function is used as a covariate. Consider predicting the octane rating of fuel (expensive to measure) from a near-infrared spectrum from the fuel (cheaper). The spectrum measures sample reflectance at a large number of closely space frequencies, $\nu$. A model might be

$$y_i = \int f(\nu) k_i(\nu) d\nu + \epsilon_i$$

where $y_i$ is the directly measured octance rating and $k_i(\nu)$ the corresponding spectrum. $f(\nu)$ is a smooth coefficient function to estimate and $\epsilon_i$ a noise term. Because the model is linear in $f(\nu)$ it fits readily into the framework already discussed.

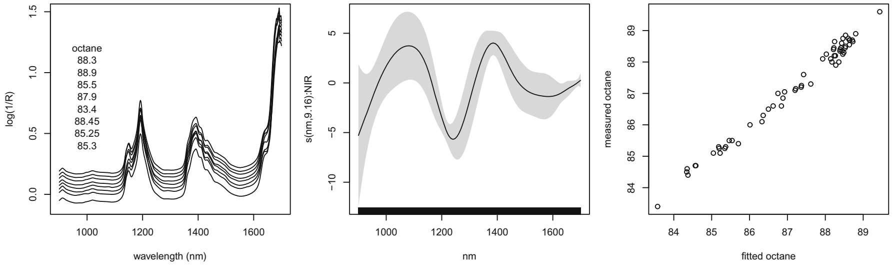

**Fig. 8** Simple signal regression example. Left: 8 (of 60) near-infrared spectra from gasoline samples along with their octane ratings. The spectra are vertically shifted for clarity, with the octane ratings in the same order. Since the spectra are cheap to measure, relative to the octane rating, it would be good to predict the latter from the former. Middle: estimated smooth coefficient function, $f$, from the model $\texttt{octane}_i = \int f(\nu)k_i(\nu)d\nu + \epsilon_i$ where $\nu$ is frequency and $k_i(\nu)$ the $i$th spectrum. Right: the relationship between measured and predicted octane

Given a basis expansion $f(\nu) = \sum_j \beta_j b_j(\nu)$, the model matrix elements are $X_{ij} = \int b_j(\nu)k_i(\nu)d\nu$. Of course in reality the integral is usually replaced by a discrete sum and some quadrature weights. Figure 8 illustrates the fit of this model. Other linear functional terms are possible, and such terms can of course be mixed with more conventional terms.

### 3.4 Extension IV: several smooth linear predictors

So far we have considered models where only a single location parameter of the response distribution depends on covariates. There is also nothing to stop us having a smooth additive linear predictor for several parameters of the response distribution. For example, if the $y_i$ are independent given covariates, we might have $y_i \sim \mathcal{D}(\theta_{1i}, \theta_{2i}, \ldots)$ where $g_j(\theta_{ji}) = \sum_k f_{jk}(x_{jki})$, $\mathcal{D}$ denotes a distribution, with parameter $\theta_j$, $g_j$ is a link function, and $f_{jk}$ is a smooth function of covariate $x_{jk}$. On replacing the unknown smooth functions with basis expansions and penalties, nothing fundamental has changed with this extension. We still have a quadratically penalized log likelihood to optimize for the model coefficients, and the smoothing parameter estimation problem is also similar. As a simple example consider the motorcycle crash data shown in the left panel of Fig. 12. A model for these data (available in `mgcv`, for example) is

$$a_i \sim N(f_1(t_i), e^{f_2(t_i)})$$

where $f_1$ and $f_2$ are smooth functions modelling the expected acceleration and the log standard deviation of the measured acceleration. Model estimates from fitting in `mgcv`, using marginal likelihood maximization to estimate smoothing parameters, are shown in black in Fig. 12.

Once we have allowed multiple linear predictors then allowing (low-dimensional) multivariate responses is also only a small step that does not fundamentally alter the fitting problem. See Yee and Wild (1996), Yee (2015), Rigby and Stasinopoulos (2005), Klein et al. (2014), Klein et al. (2015) and Wood et al. (2016) for various approaches to distributional and multivariate smooth regression and software (e.g. `mgcv` offers several such models, `gamlss` far more).

### 3.5 Extension V: general dependence on several smooth functions

Having got this far adding generalizations which leave basic fitting objective (8), and associated inferential framework unchanged, it is worth asking just how general the framework appears to be? The answer is that we can use basically the same framework for any model in which the likelihood depends on covariates via multiple smooth functions of those covariates, where 'smooth function' is taken to include any term represented by some model matrix columns with (optionally) a nonnegative quadratic penalty. Generally we assume that the penalties are linear in the smoothing parameters, but even this is relaxable, with some loss of numerical robustness. A simple example that fits into this general framework is the Cox proportional hazards model with a smooth additive predictor (implemented by the `cox.ph` family in `mgcv`), for which the log likelihood does not have the standard sum-of-independent-terms form of the models considered in previous sections (unless we are prepared to increase the computational cost by a factor of $n$ and use an equivalent Poisson likelihood for pseudodata).

Models in which some smooth terms are subject to shape constraint can also be covered by this extension. A simple approach follows Pya and Wood (2015) and represents the shape-constrained smooths (including smooth interactions) using a nonlinear reparameterization of P-splines (Eilers and Marx 1996). The key insight is that shape constraints such as monotonicity and convexity of a function can be imposed by applying the constraint to the coefficients of a B-spline basis expansion (e.g. if the coefficients are increasing with covariate $x$ then so is the resulting spline). A simple reparameterization imposes such conditions on the basis coefficients, and a quadratic smoothing penalty on the working coefficients imposes smoothness.[7]

## 4 Model checking

Model checking is the search for evidence that our model is detectably and substantially wrong. In linear modelling, checks are most usefully based on the model residuals, $y_i - \hat{\mu}_i$, which should approximate i.i.d. $N(0, \sigma^2)$ random deviates if the model is correct. Plots of residuals against covariates and $\hat{\mu}_i$ can reveal that they are not, as can QQ-plots, and in some cases plots of autocorrelation functions or variograms. There are a number of analogues of linear model residuals for more general likelihoods. For example deviance residuals are based on twice the difference between the maximum value that the likelihood contribution for $y_i$ could have taken, and the (generally lower) value it took under the model. The sign of $y_i - \hat{\mu}_i$ is often attached to this, when it

[7] There is a tendency for the literature on shape-constrained smoothing to contain assertions that smoothing penalties are unnecessary under shape constraint: there is no theoretical or empirical evidence that this is true. What is true is that elimination of smoothing penalties makes theorem proving much easier.

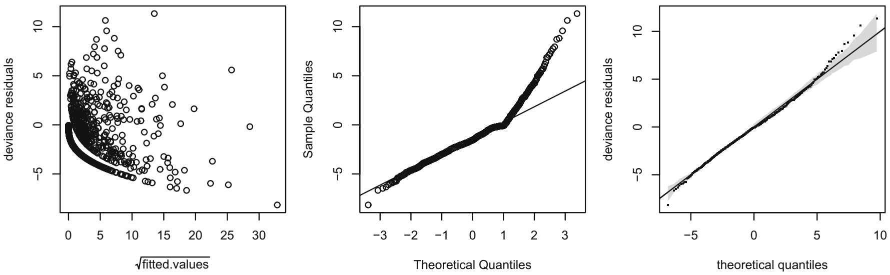

**Fig. 9** Simple residual checking example. The plots relate to a Tweedie model of Horse Mackerel egg count sample data collected by survey ships from the coastal waters of Western Britain, France and Northern Spain: smooth effects of location, salinity, surface and 20 m depth temperature were included, explaining 70% of the deviance. The count data are 71% zeroes, as large parts of the survey area returned zero counts: the zeroes produce the prominent lower bounding curve in the left plot of deviance residuals against square root of fitted values (which is unproblematic). Middle: a standard normal QQ-plot of deviance residuals for the model: it appears problematic and might lead to the wrong conclusion that a zero-inflated model is needed. Right a QQ-plot (with reference band) based on the simulation-based distribution of residuals expected if the model is correct: this appears largely unproblematic

makes sense to do so. For some models the deviance residuals have approximately independent $N(0, 1)$ distributions if all is well, which means that they can be checked in the same way as linear model residuals.

Such residual checks also apply when using smooth models, but there are two issues to be aware of. Firstly, there are many distributions for which the deviance residual distribution is far from $N(0, 1)$ even if the model is exactly correct. Low count data are an example. Neglect of this fact is a key driver of the overuse of zero-inflated models.[8] This problem can be overcome by repeatedly simulating new data from the fitted model and recomputing residuals, to build up the empirical distribution of the residuals when the model is correct. Figure 9 gives an example of some checking plots for count data with a high proportion of zeroes, illustrating the danger of being mislead by naive interpretation of the plots and data. See also Augustin et al. (2012).

The second problem occurs when the number of model coefficients is large (so that the $p = o(n^{1/3})$ assumption needed for a Gaussian posterior approximation is implausible). In this case the uncertainty in $\hat{\mu}_i$ becomes a nonnegligible part of the variability in the residuals, which can again cause the residual distribution to differ from that under the true $\mu_i$. Again simulation can help—but this time we have to simulate new data from the fitted model, and then refit to those data in order to simulate from the residual distribution.

The checking step unique to smooth models is the need to check the basis dimensions used for function approximation. Usually we look for patterns in the model residuals with respect to the covariates of the smooth function being checked. If we find pattern—especially related to positive correlation at small distances—then it could be that the basis dimension is too small. Variograms are one way to do this. Another is to compare the mean squared difference between neighbouring residuals to the dis-

[8] Although secondary to the even worse error of using the fact that the *marginal* distribution of the data has lots of zeroes and does not look marginally Poisson, negative binomial, or whatever.

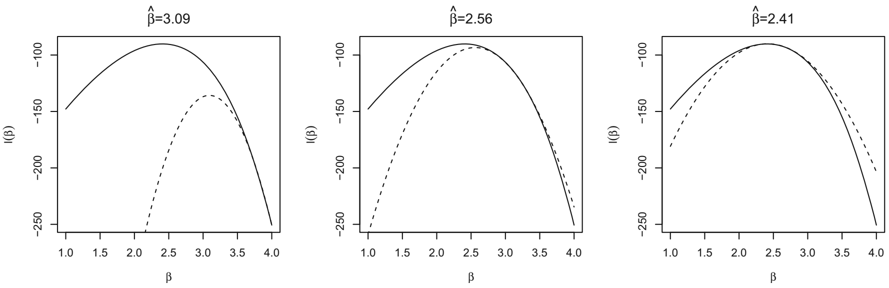

**Fig. 10** Newton's method illustrated for a single parameter, $\beta$. The fitting objective, $l(\beta)$ (i.e. a log likelihood, penalized log likelihood or log marginal likelihood), is shown as a black curve in each panel. Left: we start with a parameter guess $\hat{\beta} = 4$ and evaluate $l$, $\mathrm{d}l/\mathrm{d}\beta$ and $\mathrm{d}^2l/\mathrm{d}\beta^2$ at this point. The dashed curve shows the quadratic function matching $l$ and its first 2 derivatives at $\hat{\beta} = 4$. This quadratic is maximized at $\hat{\beta} = 3.09$, so this becomes the next estimate. Middle: the evaluation of $l(\beta)$ and derivatives is repeated at $\hat{\beta} = 3.09$, and the matching quadratic is maximized again to get $\hat{\beta} = 2.56$. Right: the process is repeated again and is almost converged

tribution of this quantity under randomization of the residuals (see Wood 2017a, §5.9 and `mgcv` function `gam.check`).

## 5 Computational methods

Writing down model extensions is easy: computing with them in an efficient and stable manner is more challenging, and this section outlines some alternative approaches. The preceding discussion implicitly favours an approach that might be termed *Bayes empirical smoothing theory* (BEST). It is appropriate when the total number, $p$, of model coefficients, $\boldsymbol{\beta}$, (smooth coefficients plus random effects) is sufficiently modest that $p = o(n^{1/3})$ is a reasonable assumption, leading to asymptotically Gaussian posterior (5) and a well-founded Laplace approximation to the marginal likelihood. For small samples, a complex random effects structure, or when $p$ is too large for the $p = o(n^{1/3})$ assumption to be reasonable, a fully Bayesian approach is required, and the choice is then between *stochastic simulation* and higher-order approximation for the coefficients, via the *INLA* method. The latter was designed for the $p = O(n)$ case, enabling modelling of short-range autocorrelation (exploiting sparse bases and penalties). When the model structure is uncertain and large numbers of smooth terms have to be screened for inclusion, then *gradient boosting* is often effective. Smooth term estimates are built up iteratively from sums of the oversmoothed versions of each smooth term, fitted to the gradient of the log likelihood. This approach can elegantly integrate model selection with fitting.

### 5.1 Bayes empirical smoothing theory

This approach finds $\hat{\boldsymbol{\lambda}}$ by maximizing (6) with respect to $\boldsymbol{\lambda}$ (or in practice $\boldsymbol{\rho} = \log \boldsymbol{\lambda}$). A Newton or quasi-Newton method is usually employed for this purpose. That is we iteratively maximize quadratic approximations to $\log \pi_L(\mathbf{y}|\boldsymbol{\lambda})$, where the approxima-

tions are based on derivatives of $\log \pi_L$ w.r.t. $\boldsymbol{\rho}$ at each successive trial $\hat{\boldsymbol{\lambda}}$. Because (6) depends on $\boldsymbol{\lambda}$ via $\hat{\boldsymbol{\beta}}$, as well as directly, each step of the optimization for $\hat{\boldsymbol{\lambda}}$ requires an 'inner' maximization of (8) to find the $\hat{\boldsymbol{\beta}}$ corresponding to the current $\hat{\boldsymbol{\lambda}}$. Furthermore we require the derivatives of $\hat{\boldsymbol{\beta}}$ w.r.t. $\boldsymbol{\rho}$, in order to compute the derivatives of $\log \pi_L$ w.r.t. $\boldsymbol{\rho}$, required by the 'outer' iteration: this can be achieved by implicit differentiation. For example, it is straightforward to show that, $\mathrm{d}\hat{\boldsymbol{\beta}}/\mathrm{d}\rho_i = -\lambda_i(\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda)^{-1}\mathbf{S}_i\hat{\boldsymbol{\beta}}$, and to differentiate again to get second derivatives. Figure 10 illustrates Newton's method. The discussion here is based on Wood (2011) and Wood et al. (2016).
To fix ideas, here is an outline of the algorithm, which is iterated to convergence

1. Given $\hat{\boldsymbol{\lambda}} = \exp(\hat{\boldsymbol{\rho}})$ iterate $\hat{\boldsymbol{\beta}} \leftarrow \hat{\boldsymbol{\beta}} + (\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_{\hat{\lambda}})^{-1}(\mathrm{d}l/\mathrm{d}\boldsymbol{\beta} - \mathbf{S}_{\hat{\lambda}}\hat{\boldsymbol{\beta}})$ to convergence.
2. Compute $\partial\hat{\boldsymbol{\beta}}/\partial\boldsymbol{\rho}$ and $\partial^2\hat{\boldsymbol{\beta}}/\partial\boldsymbol{\rho}\partial\boldsymbol{\rho}^T$, and hence obtain $\partial \log \pi_L/\partial\boldsymbol{\rho}$ and $\partial^2 \log \pi_L/\partial\boldsymbol{\rho}\partial\boldsymbol{\rho}^T$.
3. Set

$$\hat{\boldsymbol{\rho}} \leftarrow \hat{\boldsymbol{\rho}} - \left(\frac{\partial^2 \log \pi_L}{\partial\boldsymbol{\rho}\partial\boldsymbol{\rho}^T}\right)^{-1} \frac{\partial \log \pi_L}{\partial\boldsymbol{\rho}}.$$

*Practical Details:* 1. To ensure convergence, the Hessian matrices, $-\partial^2 \log \pi_L/\partial\boldsymbol{\rho}\partial\boldsymbol{\rho}^T$ and $\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda$, must be perturbed to be positive definite if they are not already. Also the update steps taken at steps 1 and 3 must be (repeatedly) halved if they decrease (8) or $\log \pi_L$, respectively. 2. Convergence occurs when the derivatives of the penalized log likelihood w.r.t. $\boldsymbol{\beta}$ are near zero (step 1. iteration) or when $\partial\hat{\boldsymbol{\beta}}/\partial\boldsymbol{\rho} \simeq 0$ (whole iteration). 3. Starting $\hat{\boldsymbol{\beta}}$ from its previous value, or from $\hat{\boldsymbol{\beta}} + \Delta\boldsymbol{\rho}^T\partial\hat{\boldsymbol{\beta}}/\partial\boldsymbol{\rho}$, ensures rapid convergence at step 1. 4. Indefiniteness at step 2 can be dealt with using a pivoting approach when obtaining the Cholesky factor of $\hat{\boldsymbol{\mathcal{I}}}+\mathbf{S}_\lambda$. 5. Quasi-Newton optimization can be substituted for optimization w.r.t. $\boldsymbol{\rho}$. 6. A final numerical unpleasantness is that the Gaussian prior and the approximate Gaussian posterior used in $\log \pi_L$ involve two log determinant terms that have to be computed: $\log|\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda|$ and $\log|\mathbf{S}_\lambda|_+$.[9] Naïve computation of these can fail badly, especially when some smoothing parameters tend to infinity, as they may legitimately do if some terms should be 'completely smooth'. The difficulty then is that the eigenvalues of $\mathbf{S}_\lambda$ or $\hat{\boldsymbol{\mathcal{I}}} + \mathbf{S}_\lambda$ can become so disparate in size that the smaller values lose all precision, so that the evaluation of the log determinant similarly loses all precision. There are two alternative solutions to this problem. Alternative 1: if $\mathbf{S}_\lambda$ is block diagonal with only one smoothing parameter per block, then the log determinant can be computed blockwise without problem, but when blocks have multiple smoothing parameters (e.g. tensor product smooths), then it is possible to automatically reparameterize to recover stability (see Wood 2011, 2017a, §6.2.7). Alternative 2 recognizes that the log determinants only need to be evaluated in order to check that the Newton steps are increasing $\log \pi_L$, rather than diverging. An alternative is to check that the less numerically problematic directional derivative

[9] $|\mathbf{S}_\lambda|_+$ is a generalized determinant—the product of the nonzero eigenvalues of $\mathbf{S}_\lambda$.

$\Delta \boldsymbol{\rho}^T \partial \log \pi_L / \partial \boldsymbol{\rho}$ is not too negative at the proposed $\hat{\boldsymbol{\rho}}$, as this would indicate that we have overstepped the maximum (Wood et al. 2017).[10]

Once we have $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\lambda}}$ we can use (5) for further inference. On occasion we may want to adjust this result for smoothing parameter uncertainty. The simplest correction (Wood et al. 2016) is to set the covariance matrix for $\boldsymbol{\beta}|\mathbf{y}$ to

$$\mathbf{V}_{\boldsymbol{\beta}} = (\hat{\mathcal{I}} + \mathbf{S}_\lambda)^{-1} - \frac{\partial \hat{\boldsymbol{\beta}}}{\partial \boldsymbol{\rho}} \left( \frac{\partial^2 \log \pi_L}{\partial \boldsymbol{\rho} \partial \boldsymbol{\rho}^T} \right)^{-1} \frac{\partial \hat{\boldsymbol{\beta}}^T}{\partial \boldsymbol{\rho}^T}.$$

A useful application of this correction is in the calculation of the AIC, where we set the effective degrees of freedom to $\tau = \text{tr}(\mathbf{V}_\beta \hat{\mathcal{I}})$, to avoid the problems otherwise caused by neglecting smoothing parameter uncertainty in AIC computation.

### 5.1.1 Automatic differentiation, less differentiation and BEST

The numerical approach sketched out above requires 3rd- or 4th-order derivatives of the model log-likelihood with respect to its parameters, depending on whether we use quasi-Newton or Newton's method. This acts as an impediment to the implementation of new model likelihoods, since the derivatives must first be found and then implemented carefully to avoid numerical instability (via cancellation error, for example). There are two approaches to easing the burden: find methods that require fewer derivatives, or automate the differentiation.

An approach that avoids the higher-order derivatives is a generalization of the method of Fellner (1986) and Schall (1991), which alternates Newton updates of $\hat{\boldsymbol{\beta}}$ given $\hat{\boldsymbol{\lambda}}$ with updates

$$\lambda_j \leftarrow \frac{\text{tr}(\mathbf{S}_\lambda^- \mathbf{S}_j) - \text{tr}\{(\hat{\mathcal{I}} + \mathbf{S}_\lambda)^{-1} \mathbf{S}_j\}}{\hat{\boldsymbol{\beta}}^T \mathbf{S}_j \hat{\boldsymbol{\beta}}} \lambda_j.$$

$\text{tr}(\mathbf{S}_\lambda^- \mathbf{S}_j)$ is the formal expression for the derivative of the log generalized determinant $\log |\mathbf{S}_\lambda|_+$ w.r.t. $\lambda_j$. It is of course not computed by forming $\mathbf{S}_\lambda^-$ explicitly: for example, for terms with a single smoothing parameter $\lambda_j$, $\text{tr}(\mathbf{S}_\lambda^- \mathbf{S}_j) = \text{rank}(\mathbf{S}_j)$. Wood and Fasiolo (2017) show that the approach approximately optimizes the Laplace approximate marginal likelihood $\pi_L$, and discuss convergence properties. Clearly it only requires first and second derivatives of the model log likelihood, but we lose access to information on smoothing parameter uncertainty.

Alternatively we can seek to automate the differentiation process. An obvious approach is to use a symbolic algebra package to obtain the derivatives of the log likelihood and to try to automate the process of turning those derivatives into code. The generalized extreme value distribution is an example in which the derivative systems are very tedious to attempt to deal with in a nonautomated way. In consequence

[10] Alternative 2 does not carry the same convergence guarantees as 1 and does not work with quasi-Newton optimization of smoothing parameters. Quasi-Newton methods maintain an approximation to the Hessian or its inverse and require careful step length control to maintain positive definiteness of this approximation.

the `gevlss` log likelihood in `mgcv` was produced by symbolic differentiation in Maxima, export of the resulting Maxima expressions to R and auto-translation and simplification in R. This works, but still required manual intervention to recode some expressions in more stable, less cancellation or overflow prone manners.

The less obvious approach to automation is to use automatic differentiation (AD). This eliminates the ‘maths-to-code’ translation of derivatives, by differentiating the computer code implementing the derivatives directly. It should not be confused with approximate evaluation of derivatives by finite differencing. AD methods apply the chain rule directly to the computer code implementing the evaluation of a function. See Wood (2015, §5.5) for an introduction. Given libraries for AD the complete BEST approach can be implemented for any model by simply coding up the log likelihood: the TMB package in R does just that (Kristensen et al. 2016). For nonstandard models in particular this is a compelling option. The downside is that AD carries overheads that can reduce efficiency, and for large-scale models in particular, one may have to work quite hard to maintain enough sparsity to avoid high memory costs. There is also no guarantee that the AD derivatives will avoid cancellation and overflow instabilities any more easily than ‘hand coded’ derivatives.

## 5.2 Full Bayes

The BEST approach estimates the smoothing parameters, $\hat{\boldsymbol{\lambda}}$, and performs further inference with these values fixed or uses simple corrections for their uncertainty. It also uses simple Gaussian approximations to the posterior distribution of $\boldsymbol{\beta}$. We only have strong theoretical backup for this approach when $n$ is large and the assumption $p = o(n^{1/3})$ is reasonable. In particular for situations in which the model effective degrees of freedom is fairly large in proportion to the sample size, or the sample size is small, the approximations are likely to show nonnegligible errors, and a fully Bayesian approach is needed. There are two main approaches at present: stochastic sampling or higher-order approximation.

### 5.2.1 Stochastic sampling

For a fully Bayesian approach we will need a prior for $\boldsymbol{\lambda}$, so that the joint prior for all the model parameters is now $\pi(\boldsymbol{\beta}|\boldsymbol{\lambda})\pi(\boldsymbol{\lambda})$. Let us write $\boldsymbol{\beta}$ and $\boldsymbol{\lambda}$ in a single parameter vector $\boldsymbol{\vartheta}$. We can simulate from the posterior distribution $\pi(\boldsymbol{\vartheta}|\mathbf{y})$ using the Metropolis Hastings algorithm. Firstly assume that we have some distribution $q(\boldsymbol{\vartheta}'|\boldsymbol{\vartheta})$ from which to generate *proposal* values, $\boldsymbol{\vartheta}'$, for the parameters, given current values, $\boldsymbol{\vartheta}$. The sampling algorithm is as follows.

Set $\boldsymbol{\vartheta}$ to any possible value and repeat the following steps

1. Generate a proposal $\boldsymbol{\vartheta}' \sim q(\boldsymbol{\vartheta}'|\boldsymbol{\vartheta})$ and a $U(0, 1)$ random deviate $u$.
2. Set $\boldsymbol{\vartheta} \leftarrow \boldsymbol{\vartheta}'$ if (computing on log scale to avoid underflow)

$$\begin{aligned}\log u \le\ & \log \pi(\mathbf{y}|\boldsymbol{\vartheta}') + \log \pi(\boldsymbol{\vartheta}') + \log q(\boldsymbol{\vartheta}|\boldsymbol{\vartheta}) \\ & - \log \pi(\mathbf{y}|\boldsymbol{\vartheta}) - \log \pi(\boldsymbol{\vartheta}) - \log q(\boldsymbol{\vartheta}'|\boldsymbol{\vartheta})\end{aligned}$$

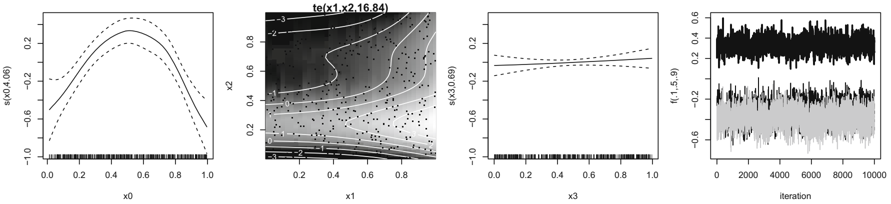

**Fig. 11** Simple stochastic simulation example. Sampling from the posterior of $y_i \sim \text{Gamma}(\mu_i, \phi)$, $\log \mu_i = f_0(x_{0i}) + f_1(x_{1i}, x_{2i}) + f_2(x_{3i})$ was performed using JAGS (Plummer 2003), set up using the `jagam` function in `mgcv` (Wood 2016). The leftmost three panels show posterior mean estimates for the functions (with credible intervals for $f_0$ and $f_2$) . The rightmost plot is a trace plot showing the simulated values of $f_0$ at $x_0 = 0.1, 0.5$ and 0.9

3. Store the current $\boldsymbol{\vartheta}$.

The stored $\boldsymbol{\vartheta}$ vectors form a nonindependent sample from $\pi(\boldsymbol{\vartheta}|\mathbf{y})$. Because the samples are not independent and the initial $\boldsymbol{\vartheta}$ may be improbable, we discard some samples from the early part of the iteration, only retaining those from the point at which the simulated chain of values appears to have settled down to the centre of the posterior distribution. All other inferential questions are then addressed using this sample from the posterior. See Fig. 11 for an example.

*Practical details:* 1. Computational efficiency rests on the nontrivial task of finding a proposal distribution, $q$, that takes large steps likely to be accepted. 2. Nothing in the algorithm prevents us from updating only some parameters at each iteration: such block updating can make it easier to find good proposals. If we propose from the distribution of the updated parameters, conditional on the other parameters, we have *Gibbs sampling*. 3. An obvious approach is to base $q$ on (5): a) We could use (5) as a fixed proposal, independent of the current simulated $\boldsymbol{\vartheta}$. But then the chain can get ‘stuck’ in regions that are much more probable under the posterior than under the proposal (making it heavier tailed can help). b) We can use (5) as the basis for a random walk proposal centred on the current $\boldsymbol{\vartheta}$. Usually the covariance matrix is shrunk in this case. c) We might base blockwise proposals on approximate versions of (5) built only from the information in that block. 4. Greater efficiency can be gained using hybrid/Hamiltonian Monte Carlo methods which augment $\boldsymbol{\vartheta}$ with some auxiliary momentum variables, giving each component of $\boldsymbol{\vartheta}$ the tendency to keep going in the direction of increasing probability.

See Wood (2015, ch. 6) for MCMC basics and e.g. Fahrmeir and Lang (2001), Fahrmeir et al. (2004), Lang and Brezger (2004) and Lang et al. (2014) for more on this approach.

### 5.2.2 INLA

Stochastic sampling becomes increasingly difficult as the dimension increases. Rue et al. (2009) realized that there is a rather efficient way to obtain very accurate approximations to the marginal posterior distributions of model coefficients with no simulation. The original *integrated nested Laplace approximation* (INLA, Rue et al. 2009; Lindgren et al. 2011; Martins et al. 2013; Rue et al. 2017) is closely tied to sparse representation of effects as Gaussian Markov random fields (GMRF), but the ideas can also be extended to the nonsparse case.

INLA obtains the marginal distribution of $\boldsymbol{\beta}$ and hyperparameters (including smoothing parameters), $\boldsymbol{\vartheta}$, from

$$\pi(\beta_i|\mathbf{y}) = \int \pi(\beta_i|\boldsymbol{\vartheta}, \mathbf{y})\pi(\boldsymbol{\vartheta}|\mathbf{y})d\boldsymbol{\vartheta} \text{ and } \pi(\vartheta_i|\mathbf{y}) = \int \pi(\boldsymbol{\vartheta}|\mathbf{y})d\boldsymbol{\vartheta}_{-i} \tag{9}$$

where a subscript '$-i$' denotes a vector without its $i$th element. Laplace approximations are used for the distributions in the integrands, and the integrals are evaluated using relatively coarse numerical quadrature (see Rue et al. 2009, especially §6.5). Alternatively we might choose to skip the integration step, simply setting $\boldsymbol{\vartheta}$ to its posterior mode.

The posterior of $\boldsymbol{\vartheta}$ is approximated using the same first-order Laplace approximation (6) employed in the empirical Bayes approach

$$\tilde{\pi}(\boldsymbol{\vartheta}|\mathbf{y}) \propto \pi(\hat{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\vartheta})/\pi_G(\hat{\boldsymbol{\beta}}|\mathbf{y}, \boldsymbol{\vartheta})$$

where $\hat{\boldsymbol{\beta}}$ is the maximizer of $\pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\vartheta})$ and $\pi_G(\boldsymbol{\beta}|\boldsymbol{\vartheta}, \mathbf{y}) = N(\hat{\boldsymbol{\beta}}, \mathbf{H}^{-1})$ where $\mathbf{H}$ is the Hessian of $-\log \pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\vartheta})$ w.r.t. $\boldsymbol{\beta}$ at $\hat{\boldsymbol{\beta}}$. Since $\pi_G$ is evaluated at its mode the approximation is simply $\pi(\boldsymbol{\theta}|\mathbf{y}) \propto \pi(\hat{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\vartheta})/|\mathbf{H}|^{1/2}$. $\hat{\boldsymbol{\beta}}$ and the Hessian are identically those used in BEST.

The most important step in INLA is the approximation

$$\tilde{\pi}(\beta_i|\boldsymbol{\vartheta}, \mathbf{y}) \propto \pi(\tilde{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\vartheta})/\pi_{GG}(\tilde{\boldsymbol{\beta}}_{-i}|\beta_i, \mathbf{y}, \boldsymbol{\vartheta}), \tag{10}$$

where $\tilde{\boldsymbol{\beta}}$ maximizes $\pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\vartheta})$ given the constraint $\tilde{\beta}_i = \beta_i$, and $\pi_{GG}$ is a Gaussian approximation to $\pi(\boldsymbol{\beta}_{-i}|\beta_i, \mathbf{y}, \boldsymbol{\vartheta})$. Following the empirical Bayes route, we could approximate $\pi(\beta_i|\boldsymbol{\vartheta}, \mathbf{y})$ directly from $\pi_G(\boldsymbol{\beta}|\boldsymbol{\vartheta}, \mathbf{y})$, but this would involve evaluating a Gaussian approximation in the distribution's tails, where it is often inaccurate. In contrast (10) only requires the evaluation of a Gaussian approximation *at its mode* and is therefore much more accurate. At worst, a relative error in $\pi_{GG}$ at its mode produces an equivalent relative error in (10): in comparison for the marginal based on $\pi_G$ the error simply grows as we move into the tails. Finally, the approximate $\pi(\beta_i|\boldsymbol{\vartheta}, \mathbf{y})$ is always renormalized in practice: this is easily done for a one-dimensional function and eliminates any constant error due to inaccuracy of $\pi_{GG}$ at its mode.

If $\pi_{GG}$ is based directly on the mode and Hessian of $\log \pi(\boldsymbol{\beta}_{-i}|\beta_i, \mathbf{y}, \boldsymbol{\vartheta})$ then (10) is exactly the Laplace approximation to $\int \pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\vartheta})d\boldsymbol{\beta}_{-i}$, and the informal discussion of approximation error, above, can be formalized (Shun and McCullagh 1995; Rue et al. 2009). But direct evaluation of the required Hessian *for each* $\beta_i$ is computationally prohibitive. Computationally efficient approximations to the Laplace approximation are required. One possibility is to base $\pi_{GG}$ on the conditional density implied by $\pi_G$, in which case the Hessian is constant and

$$\tilde{\boldsymbol{\beta}}_{-i} = \hat{\boldsymbol{\beta}}_{-i} + \boldsymbol{\Sigma}_{-i,i}\Sigma_{i,i}^{-1}(\beta_i - \hat{\beta}_i), \tag{11}$$

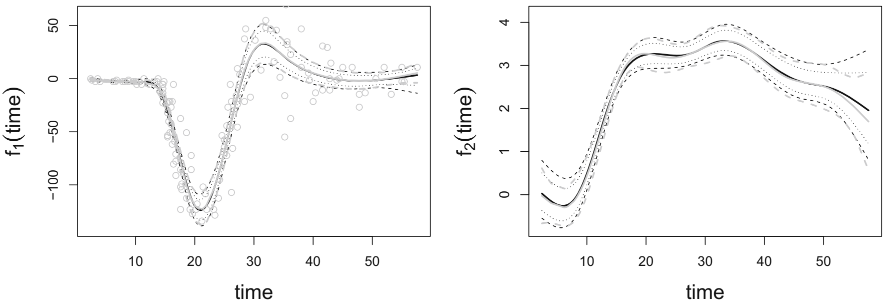

**Fig. 12** Distributional regression and INLA example. Left: the grey circles show acceleration, $a$, of the head of a crash test dummy against time, $t$, in simulated motorcycle accidents. A model for the data is $a_i \sim N(f_1(t_i), e^{f_2(t_i)})$. The partially obscured continuous black curve shows the $\hat{f}_1$ using the BEST approach, while the black dashed lines denote the corresponding 95% CIs for $f_1$. The overlayed grey curves are the 0.025, 0.1, 0.5, 0.9 and 0.975 quantiles of the posterior for $f_1$ computed using the INLA approach. Right: similar plot for $f_2$

where $\boldsymbol{\Sigma} = \mathbf{H}^{-1}$. This results in the approximation $\pi(\beta_i|\boldsymbol{\vartheta}, \mathbf{y}) \propto \pi(\tilde{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\vartheta})$, which is demonstrably a substantial improvement on directly using the marginal from $\pi_G$. Better approximations are possible, however. Rue et al. (2009) use (11), but also approximate the dependence on $\beta_i$ of the Hessian of $\log \pi(\boldsymbol{\beta}_{-i}|\beta_i, \mathbf{y}, \boldsymbol{\vartheta})$. They offer two alternatives. The first exploits the heuristic that only elements of $\boldsymbol{\beta}_{-i}$ showing sufficiently high correlation to $\beta_i$ according to $\pi_G$ need be considered when approximating how the Hessian varies with $\beta_i$: this leads to relatively efficient computation for GMRF models. The second, faster and recommended, approach replaces the log determinant of the required Hessian with a first-order Taylor approximation about $\hat{\boldsymbol{\beta}}$. The required log determinant derivative is fairly cheap for GMRF models.

The Rue et al. (2009) strategies are inefficient for nonsparse reduced rank basis expansions, but Wood (2019) provides an alternative for this case. The key is to note that the Cholesky factor of $\mathbf{H}_{-i,-i}$ can be obtained by cheap, $O(p^2)$, update of the Cholesky factor of $\mathbf{H}$. This immediately gives a computationally efficient means to find the exact $\tilde{\boldsymbol{\beta}}_{-i}$ by improving (11) using modified Newton updates based on a fixed Hessian $\mathbf{H}_{-i,-i}$. Hence one of the approximations in the Rue et al. (2009) method is removed. The second thing that it permits is to approximate the required Hessian matrix by a BFGS update of $\mathbf{H}_{-i,-i}$. The log determinant of this update is very efficiently computed, and the log determinant is bounded between that of $\mathbf{H}_{-i,-i}$ and the true Hessian. Figure 12 shows an example distributional regression fit where there are noticeable differences between BEST and INLA. The computations used the `ginla` function in `mgcv`.

### 5.3 Boosting

A rather different approach to smooth model estimation uses boosting (e.g. Tutz and Binder 2006; Schmid and Hothorn 2008; Robinzonov and Hothorn 2010; Mayr et al. 2012). Boosting is a forward selection strategy, in which smooth model components are iteratively built up from over-smoothed versions, fitted to generalized residuals

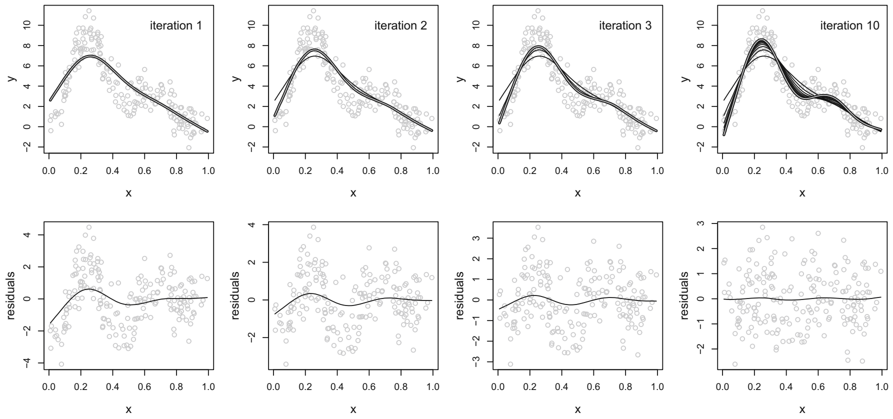

**Fig. 13** Univariate smoothing of Gaussian data, shown in grey, via boosting. Top row: several boosting steps, each showing the current estimate as a thick black–grey curve and previous estimates as thin black curves. At top right the initial estimate is just the base learner applied directly to smooth the data. Bottom row: the base learner is used to smooth the residuals (grey) from the panel immediately above. The resulting smooth (thin black) is added to the previous function estimate (black–grey curve, panel above) to give the updated function estimate shown in the upper row of the next column to the right. Notice how the estimate successively improves, while the residuals become de-correlated

of the model. It can be viewed as a variation of the backfitting method (Hastie and Tibshirani 1986) for additive model fitting.

The 'oversmoothed versions' of the smooth terms are generally termed 'base learners', and the fitting to residuals is done by least squares. For example, if $f$ is a simple single penalty smooth with basis matrix $\mathbf{X}$ and penalty $\mathbf{S}$, and $\mathbf{g}$ denotes some generalized residuals, then the over-smoothed version of $f$ is given by

$$\tilde{\mathbf{f}} = \mathbf{A}\mathbf{g} \text{ where } \mathbf{A} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda_{\text{big}}\mathbf{S})^{-1}\mathbf{X}^T.$$

$\lambda_{\text{big}}$ is large enough that $\tilde{f}$ has very low effective degrees of freedom (perhaps 1-2 more than its null space dimension). $\mathbf{A}$ is known as the 'smoother', 'influence' or 'hat'—matrix (for efficiency reasons it is not calculated explicitly, of course). The 'generalized residuals', $\mathbf{g}$, are the partial derivatives of $-2\times$ the log likelihood w.r.t. the elements of $\tilde{\mathbf{f}}$, given the model fit so far. For a model with a single linear predictor, these are simply the derivatives with respect to the linear predictor, at its current estimate. For a model with multiple linear predictors then the generalized residuals for a term $f_j$ are the derivatives w.r.t. the linear predictor that it is part of.

Boosting cycles through the base learners, and in most cases we would choose the single base learner that leads to the biggest increase in likelihood at each cycle, and add this to the fitted model. In this way boosting can integrate model selection and fitting, since some base learners may never occur in the model. For a more concrete understanding of how it works, here is the algorithm for the case of a single linear predictor, and see Fig. 13.

$\mathbf{A}^{[j]}$ is the smoother matrix for $f_j$, $M$ the number of smoothers and $\boldsymbol{\eta}$ the linear predictor. Initialize $\hat{\boldsymbol{\eta}} = \hat{\mathbf{f}}_1 = \cdots = \hat{\mathbf{f}}_M = \mathbf{0}$. Iterate the following until some convergence criteria is met.

1. Compute $g_i = -2\partial l/\partial \hat{\eta}_i$ for all $i$.
2. For $j = 1, \ldots, M$ ...
   (a) Compute $\tilde{\mathbf{f}}_j = \mathbf{A}^{[j]}\mathbf{g}$
   (b) Compute $\alpha_j = \text{argmax}_\alpha l(\hat{\boldsymbol{\eta}} + \alpha\tilde{\mathbf{f}}_j)$.
3. Find $k = \text{argmax}_j l(\hat{\boldsymbol{\eta}} + \alpha_j\tilde{\mathbf{f}}_j)$.
4. Set $\hat{\boldsymbol{\eta}} \leftarrow \hat{\boldsymbol{\eta}} + \alpha_k\tilde{\mathbf{f}}_k$, and $\hat{\mathbf{f}}_k \leftarrow \hat{\mathbf{f}}_k + \alpha_k\tilde{\mathbf{f}}_k$.

*Practical details:* (1) Without step length optimization at 2b, term selection is oversensitive to the base learner λs, although these should anyway be chosen to ensure similar complexity for each base learner. (2) Uncertainty estimates and a stopping criterion are needed. We can bootstrap the original data, maintain a separate linear predictor for each bootstrap replicate and terminate when the average error in predicting data omitted from the bootstrap resamples is minimized. The replicate linear predictors at termination provide uncertainty estimates.[11] (3) For multiple linear predictors, compute a $\mathbf{g}$ vector for each linear predictor at step 1, and make sure that we use the one of these corresponding to $f_j$ at step 2a. (4) The $\hat{f}_j$ estimates typically end up more complex than any individual base learner, and the effective degree of freedom of the base learners has little or no influence on the complexity of the final term estimate. However, it is not possible to obtain a fit more complex than the unpenalized basis would allow.

The major advantage of boosting relative to other approaches is the ability to efficiently use, and perform model selection with, a very large number of smooth model terms.

### 5.4 Big data methods

Big data problems are only statistically interesting when they also require large models, and in this case the main challenge is to find computational methods that are feasible on the computer hardware generally available for modelling. If the number of data, $n$, and number of coefficients, $p$, become large, then there are two obvious problems. The first is that the storage for the model matrix is $O(np)$, while the computations involving it are generally $O(np^2)$. For example, naively implemented the $n = 10^7$, $p = 10^4$ air pollution model example in Wood et al. (2017) would require about 1 terabyte just to store the model matrix in double precision and weeks of computing time to fit a model.

The first step in solving both problems is to exploit special structure in the model matrix to reduce both the storage and computing costs. There are two alternatives.

1. Use sparse bases and penalties so that the model matrix and penalty matrices contain mostly zeroes. There is much work on exploiting sparsity in matrix com-

[11] We must choose whether step 3 is repeated for each bootstrap replicate, or whether we simply use the $k$ chosen for the full dataset in each replicate. The latter is efficient, but neglects the term selection uncertainty.

putation (see Davis 2006), and the INLA software of Rue et al. (2009) makes very effective use of this approach. Note, however, that exploitation of sparse matrices is not simply a matter of substituting sparse routines for dense ones. The curse of sparse matrix computation is infill: the fact that many operations on sparse matrices will result in a dense matrix. It takes effort to avoid this.

2. Recognize that most covariates take only a finite number of discrete values, and even if they are truly continuous there is no real statistical loss associated with discretizing to $O(n^{1/2})$ bins. This means that the model matrix for a smooth term can be represented by a modest sized matrix of unique rows plus an index vector giving the unique row corresponding to each full model matrix row. This obviously saves storage, but it can also be exploited to greatly reduce the cost of computing with products involving the model matrix, as first proposed for single smooths by Lang et al. (2014) and generalized to models with multiple smooths by Wood et al. (2017) and Li and Wood (2019). To maintain statistical performance the covariates are discretized separately (discretizing jointly onto a multidimensional grid is much easier, but requires overly coarse discretization). `bam` in `mgcv` can use this approach.

Exploiting the model matrix structure is typically not sufficient on its own: further high-performance computing is usually needed and both `mgcv`'s `bam` and the INLA software do this. Here it is important to be aware of a third problem. Modern computers are memory bandwidth limited: the rate-limiting process is not how fast the multiple cores of a CPU can perform floating point operations, but how quickly the data can be fetched from main memory (RAM) to the CPU in order to be computed with. The issue is serious: it can take 20 times as long to fetch an item of data as to perform a floating point operation with it.[12] In hardware this bottleneck is partially ameliorated by cache memory: a small amount of super-fast access memory between the main memory and the CPU. To exploit the cache we need to arrange computations so that most required data are in the cache *before* being used: the way to do that is to structure things so that data that need to be reused are reused as soon as possible. In practice this means that fitting methods need to be dominated by matrix computations that are *block oriented*, meaning that they can be broken down into operations involving submatrix–submatrix operations (rather than matrix-vector), where the whole submatrices fit in cache. Matrix products and Cholesky decomposition can be structured to be almost entirely block oriented, while QR and eigen decompositions are more problematic. If we develop block-oriented fitting methods (as in Wood et al. 2017) then we will be able to get high performance using either an optimized BLAS or parallelization, or a combination of the two. Note, however, that bandwidth limitation problems are exacerbated by multicore computing, since multiple cores clamouring for data have an even higher capacity to saturate the data channels and now have to share the cache. This is the reason that multicore BLAS performance is usually disappointing relative

[12] If you doubt this, try comparing the speed of a matrix cross-product using the reference BLAS and an optimized BLAS, such as `openblas`: the difference is down to structuring the code to get around the latency problem.

to a single core BLAS, and that parallelizing using a single-core BLAS also tends to give poor scaling.

Big data, big model methods are still an active area of research. For example, at time of writing, the methods of Wood et al. (2017) are able to deal with larger model–data combinations than seems possible with other approaches, but they are limited to GAM like model structures and are not yet usable with the extensions discussed in Sects. 3.4 and 3.5.

## 6 Model selection

As mentioned above, within the BEST framework, AIC and term specific p values can be computed and used for stepwise model selection. Boosting integrates model selection and fitting, albeit from a prediction error minimization perspective. Model selection in the fully Bayesian setting is somewhat less straightforward, but the deviance information criteria (DIC, Spiegelhalter et al. 2002) is often used in practice as an analogue of AIC.

In fact much of what is traditionally thought of as model selection is carried out by smoothing parameter estimation, but for smooths, $\lambda_j \rightarrow \infty$ usually produces an $\hat{f}_j$ in the null space of the smoothing penalty, rather than resulting in $\hat{f}_j = 0$. One possibility is to associate an extra penalty with each smooth, designed to penalize functions in the null space of the smoothing penalty towards zero. To this end, consider the smoothing penalty matrix eigen decomposition $\mathbf{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, where the columns of $\mathbf{U}$ are eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix of eigenvalues. Let $\mathbf{U}_0$ denote the eigenvectors corresponding to zero eigenvalues. Then $\lambda_0 \boldsymbol{\beta}^T \mathbf{U}_0 \mathbf{U}_0^T \boldsymbol{\beta}$ is a penalty on the null space of $\lambda \boldsymbol{\beta}^T \mathbf{S} \boldsymbol{\beta}$ and if $\lambda, \lambda_0 \rightarrow \infty$ then the corresponding smooth will be estimated as zero, i.e. it will be ‘selected out’ of the model.

Such selection penalties can be included for all smooth terms in a model, so that $\boldsymbol{\lambda}$ estimation controls not only the complexity of terms, but whether they contribute to the model at all. In practice some terms may be estimated as close, but not exactly equal, to zero, and a decision threshold is needed: e.g. we might exclude terms with effective degrees of freedom $< 0.1$. Sometimes we might want smoother, simpler models than GCV or marginal likelihood selects by default. Computing the GCV, AIC or marginal likelihood as if the sample size were smaller than it actually is achieves this. In `mgcv` function `gam`, increasing the parameter `gamma` above its default of one does this: e.g. $\log(n)/2$ achieves BIC like model selection.

When there are large numbers of terms to screen for inclusion in the model, then selection penalties and conventional stepwise methods are computationally costly. By contrast boosting retains efficiency, but in the context of screening large numbers of effects its inability to drop a term, once included, is not optimal. A useful alternative is to repeatedly alternate a few steps of boosting, for ‘forward selection’ of new terms for inclusion in the model, with a fit of the model with all currently selected terms, using selection penalties to allow ‘backwards selection’.

## 7 Beyond likelihood

It is also possible to extend smooth regression methods to cases where we are interested in a loss function other than a regular likelihood. Obvious examples are provided by robust loss functions or the ‘pinball loss’ used in quantile regression (where we want to directly model some specified quantile of the response distribution). Fasiolo et al. (2017) shows how to use the belief updating framework of Bissiri et al. (2016) to do this in a well-founded manner. The idea is that we can use a general loss to update a prior to a posterior, just as we would use a likelihood, but to do so we now have to choose the ‘loss rate’ setting the relative weighting of the loss and the prior. The main challenge is to find well-founded means for selecting the loss rate. Note that in the quantile regression case the pinball loss can be identified with the log likelihood of an asymmetric Laplace distribution, and there are several papers using this to perform inference for quantile regression using standard Bayesian or penalized likelihood methods, while ignoring the selection of the loss rate. This is invalid since the asymmetric Laplace is mis-specified as a probability model, and this mis-specification tends to become extreme as we move away from the median quantile (of course there will be cases where the model fits appear sensible despite the mis-specification, but counter examples can always be found).

## 8 Conclusions

The basic framework, outlined above, represents smooth functions in regression models using basis expansions, with associated quadratic penalties on the basis coefficients designed to control smoothness of the functions during estimation/inference. If we view the penalties as being induced by Gaussian priors on the basis coefficients, then Bayes theorem allows us to perform further inference based on the posterior distribution of the coefficients, while smoothing parameter inference can be based on the marginal likelihood (the basis coefficients being integrated out). Supplementing this Bayesian outlook with some frequentist model selection tools leads to a quite practically useful framework for a wide variety of models, and depending on taste and exact practical needs we may choose to use empirical Bayes, stochastic simulation or higher-order Bayesian approximation methods for inference. The aim of this paper was to emphasize the basic unity of the model and inferential frameworks, which sometimes appear rather more different than they really are in the literature. But as in regression modelling more generally, in the high-dimensional case in which there are many effects to screen then quite different approaches tend to be useful: gradient boosting is an example that offers computational efficiency while being general enough to use with almost all the model extensions considered here.

What future developments are likely in this area? I do not know, but the further development of methods for large models of large data sets, multivariate data and short-range auto-correlation seem likely to feature prominently.

## References

Akaike H (1973) Information theory and an extension of the maximum likelihood principle. In: Petran B, Csaaki F (eds) International symposium on information theory. Akadeemiai Kiadi, Budapest, pp 267–281

Anderssen R, Bloomfield P (1974) A time series approach to numerical differentiation. Technometrics 16(1):69–75

Augustin NH, Sauleau E-A, Wood SN (2012) On quantile quantile plots for generalized linear models. Comput Stat Data Anal 56(8):2404–2409

Belitz C, Brezger A, Kneib T, Lang S, Umlauf N (2015). BayesX: software for Bayesian inference in structured additive regression models

Bissiri PG, Holmes C, Walker SG (2016) A general framework for updating belief distributions. J R Stat Soc Ser B (Stat Methodol) 78(5):1103–1130

Claeskens G, Krivobokova T, Opsomer JD (2009) Asymptotic properties of penalized spline estimators. Biometrika 96(3):529–544

Clarke E, Speirs D, Heath M, Wood S, Gurney W, Holmes S (2006) Calibrating remotely sensed chlorophyll—a data by using penalized regression splines. J R Stat Soc Ser C (Appl Stat) 55(3):331–353

Craven P, Wahba G (1979) Smoothing noisy data with spline functions. Numer Math 31:377–403

Davis TA (2006) Direct methods for sparse linear systems. SIAM, Philadelphia

de Boor C (2001) A practical guide to splines, Revised edn. Springer, New York

Duchon J (1977) Splines minimizing rotation-invariant semi-norms in Solobev spaces. In: Schemp W, Zeller K (eds) Construction theory of functions of several variables. Springer, Berlin, pp 85–100

Eilers PH, Marx BD, Durbán M (2015) Twenty years of p-splines. SORT Stat Oper Res Trans 39(2):149–186

Eilers PHC, Marx BD (1996) Flexible smoothing with B-splines and penalties. Stat Sci 11(2):89–121

Fahrmeir L, Kneib T, Lang S (2004) Penalized structured additive regression for space-time data: a Bayesian perspective. Stat Sin 14(3):731–761

Fahrmeir L, Lang S (2001) Bayesian inference for generalized additive mixed models based on markov random field priors. Appl Stat 50:201–220

Fasiolo M, Goude Y, Nedellec R, Wood SN (2017) Fast calibrated additive quantile regression. arXiv preprint arXiv:1707.03307

Fellner WH (1986) Robust estimation of variance components. Technometrics 28(1):51–60

Greven S, Kneib T (2010) On the behaviour of marginal and conditional AIC in linear mixed models. Biometrika 97(4):773–789

Gu C (1992) Cross-validating non-Gaussian data. J Comput Graph Stat 1:169–179

Gu C (2013) Smoothing spline ANOVA models, 2nd edn. Springer, New York

Gu C, Wahba G (1991) Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method. SIAM J Sci Stat Comput 12(2):383–398

Hastie T, Tibshirani R (1986) Generalized additive models (with discussion). Stati Sci 1:297–318

Hastie T, Tibshirani R (1990) Generalized additive models. Chapman & Hall, Boca Raton

Kimeldorf GS, Wahba G (1970) A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. Ann Math Stat 41(2):495–502

Klein N, Kneib T, Klasen S, Lang S (2014) Bayesian structured additive distributional regression for multivariate responses. J R Stat Soc Seri C (Appl Stat) 64:569–591

Klein N, Kneib T, Lang S, Sohn A (2015) Bayesian structured additive distributional regression with an application to regional income inequality in Germany. Ann Appl Stat 9:1024–1052

Kristensen K, Nielsen A, Berg CW, Bell HSBM (2016) TMB: automatic differentiation and laplace approximation. J Stat Softw 70(5):1–21

Lang S, Brezger A (2004) Bayesian P-splines. J Comput Graph Stat 13:183–212

Lang S, Umlauf N, Wechselberger P, Harttgen K, Kneib T (2014) Multilevel structured additive regression. Stat Comput 24(2):223–238

Li Z, Wood SN (2019) Faster model matrix crossproducts for large generalized linear models with discretized covariates. Stat Comput 30:19–25

Lindgren F, Rue H, Lindström J (2011) An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. J R Stat Soc Ser B (Stat Methodol) 73(4):423–498

Martins TG, Simpson D, Lindgren F, Rue H (2013) Bayesian computing with INLA: new features. Comput Stat Data Anal 67:68–83

Marx BD, Eilers PH (2005) Multidimensional penalized signal regression. Technometrics 47(1):13–22

Mayr A, Fenske N, Hofner B, Kneib T, Schmid M (2012) Generalized additive models for location, scale and shape for high dimensional data—a flexible approach based on boosting. J R Stat Soc Ser C (Appl Stat) 61(3):403–427

Nychka D (1988) Bayesian confidence intervals for smoothing splines. J Am Stat Assoc 83(404):1134–1143

Parker R, Rice J (1985) Discussion of Silverman (1985). J R Stat Soc Ser B 47(1):40–42

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 (DSC 2003), pp 20–22

Pya N, Wood SN (2015) Shape constrained additive models. Stat Comput 25(3):543–559

Rigby R, Stasinopoulos DM (2005) Generalized additive models for location, scale and shape. J R Stat Soc Ser C (Appl Stat) 54(3):507–554

Robinzonov N, Hothorn T (2010) Boosting for estimating spatially structured additive models. In: Statistical modelling and regression structures. Springer, pp 181–196

Rue H, Held L (2005) Gaussian Markov random fields: theory and applications. CRC Press, Boca Raton

Rue H, Martino S, Chopin N (2009) Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations (with discussion). J R Stat Soc Ser B 71(2):319–392

Rue H, Riebler A, Sørbye SH, Illian JB, Simpson DP, Lindgren FK (2017) Bayesian computing with INLA: a review. Ann Rev Stat Its Appl 4:395–421

Ruppert D, Wand MP, Carroll RJ (2003) Semiparametric regression. Cambridge University Press, Cambridge

Schall R (1991) Estimation in generalized linear models with random effects. Biometrika 78(4):719–727

Schmid M, Hothorn T (2008) Boosting additive models using component-wise P-splines. Comput Stat Data Anal 53(2):298–311

Shun Z, McCullagh P (1995) Laplace approximation of high dimensional integrals. J R Stat Soc Ser B 57(4):749–760

Silverman BW (1985) Some aspects of the spline smoothing approach to non-parametric regression curve fitting. J R Stat Soc Ser B 47(1):1–53

Spiegelhalter DJ, Best NG, Carlin BP, van der Linde A (2002) Bayesian measures of model complexity and fit. J R Stat Soci Ser B 64(4):583–639

Stasinopoulos DM, Rigby RA et al (2007) Generalized additive models for location scale and shape (gamlss) in r. J Stat Softw 23(7):1–46

Stasinopoulos MD, Rigby RA, Heller GZ, Voudouris V, De Bastiani F (2017) Flexible regression and smoothing: using GAMLSS in R. Chapman and Hall/CRC, Boca Raton

Tutz G, Binder H (2006) Generalized additive modeling with implicit variable selection by likelihood-based boosting. Biometrics 62(4):961–971

Umlauf N, Adler D, Kneib T, Lang S, Zeileis A (2015) Structured additive regression models: an R interface to BayesX. J Stat Softw 63(21):1–46

Wahba G (1980) Spline bases, regularization, and generalized cross validation for solving approximation problems with large quantities of noisy data. In: Cheney E (ed) Approximation theory III. Academic Press, London

Wahba G (1983) Bayesian confidence intervals for the cross validated smoothing spline. J R Stat Soc B 45:133–150

Wahba G (1985) A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. Ann Stat 13(4):1378–1402

Wahba G (1990) Spline models for observational data. SIAM, Philadelphia

Wood SN (2000) Modelling and smoothing parameter estimation with multiple quadratic penalties. J R Stat Soc Ser B (Stat Methodol) 62:413–428

Wood SN (2003) Thin plate regression splines. J R Stat Soc Ser B (Stat Methodol) 65:95–114

Wood SN (2008) Fast stable direct fitting and smoothness selection for generalized additive models. J R Stat Soc Ser B (Stat Methodol) 70(3):495–518

Wood SN (2011) Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. J R Stat Soc Ser B (Stat Methodol) 73(1):3–36

Wood SN (2013a) On p-values for smooth components of an extended generalized additive model. Biometrika 100(1):221–228

Wood SN (2013b) A simple test for random effects in regression models. Biometrika 100(4):1005–1010

Wood SN (2015) Core statistics. Cambridge University Press, Cambridge

Wood SN (2016) Just another Gibbs additive modeller: interfacing JAGS and mgcv. J Stati Softw 75(7)

Wood SN (2017a) Generalized additive models: an introduction with R, 2nd edn. CRC Press, Boca Raton

Wood SN (2017b) P-splines with derivative based penalties and tensor product smoothing of unevenly distributed data. Stat Comput 27(4):985–989

Wood SN (2019) Simplified integrated nested laplace approximation. Biometrika (in press)

Wood SN, Fasiolo M (2017) A generalized Fellner–Schall method for smoothing parameter optimization with application to Tweedie location, scale and shape models. Biometrics 73(4):1071–1081

Wood SN, Li Z, Shaddick G, Augustin NH (2017) Generalized additive models for gigadata: modelling the UK black smoke network daily data. J Am Stat Assoc 112(519):1199–1210

Wood SN, Pya N, Säfken B (2016) Smoothing parameter and model selection for general smooth models (with discussion). J Am Stat Assoc 111:1548–1575

Yee TW (2015) Vector generalized linear and additive models: with an implementation in R. Springer, Berlin

Yee TW, Wild C (1996) Vector generalized additive models. J R Stat Soc Ser B (Methodol) 58(3):481–493

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.