



Wood, S. N., Li, Z., Shaddick, G., & Augustin, N. H. (2017). Generalized Additive Models for Gigadata: Modeling the U.K. Black Smoke Network Daily Data. *Journal of the American Statistical Association*. DOI: 10.1080/01621459.2016.1195744

Publisher's PDF, also known as Version of record

License (if available):
CC BY

Link to published version (if available):
[10.1080/01621459.2016.1195744](https://doi.org/10.1080/01621459.2016.1195744)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the final published version of the article (version of record). It first appeared online via Taylor & Francis at <http://amstat.tandfonline.com/doi/full/10.1080/01621459.2016.1195744>. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/pure/about/ebr-terms.html>



Generalized Additive Models for Gigadata: Modeling the U.K. Black Smoke Network Daily Data

Simon N. Wood, Zheyuan Li, Gavin Shaddick & Nicole H. Augustin

To cite this article: Simon N. Wood, Zheyuan Li, Gavin Shaddick & Nicole H. Augustin (2017): Generalized Additive Models for Gigadata: Modeling the U.K. Black Smoke Network Daily Data, Journal of the American Statistical Association, DOI: [10.1080/01621459.2016.1195744](https://doi.org/10.1080/01621459.2016.1195744)

To link to this article: <http://dx.doi.org/10.1080/01621459.2016.1195744>



© 2017 Published with license by Taylor & Francis. © Simon N. Wood, Zheyuan Li, Gavin Shaddick, and Nicole H. Augustin.



[View supplementary material](#)



Accepted author version posted online: 24 Jun 2016.
Published online: 25 Apr 2017.



[Submit your article to this journal](#)



Article views: 824



[View Crossmark data](#)



Citing articles: 4 [View citing articles](#)

Generalized Additive Models for Gigadata: Modeling the U.K. Black Smoke Network Daily Data

Simon N. Wood^a, Zheyuan Li^b, Gavin Shaddick^b, and Nicole H. Augustin^b

^aSchool of Mathematics, University of Bristol, Bristol, UK; ^bDepartment of Mathematical Sciences, University of Bath, Bath, UK

ABSTRACT

We develop scalable methods for fitting penalized regression spline based generalized additive models with of the order of 10^4 coefficients to up to 10^8 data. Computational feasibility rests on: (i) a new iteration scheme for estimation of model coefficients and smoothing parameters, avoiding poorly scaling matrix operations; (ii) parallelization of the iteration's pivoted block Cholesky and basic matrix operations; (iii) the marginal discretization of model covariates to reduce memory footprint, with efficient scalable methods for computing required crossproducts directly from the discrete representation. Marginal discretization enables much finer discretization than joint discretization would permit. We were motivated by the need to model four decades worth of daily particulate data from the U.K. Black Smoke and Sulphur Dioxide Monitoring Network. Although reduced in size recently, over 2000 stations have at some time been part of the network, resulting in some 10 million measurements. Modeling at a daily scale is desirable for accurate trend estimation and mapping, and to provide daily exposure estimates for epidemiological cohort studies. Because of the dataset size, previous work has focused on modeling time or space averaged pollution levels, but this is unsatisfactory from a health perspective, since it is often acute exposure locally and on the time scale of days that is of most importance in driving adverse health outcomes. If computed by conventional means our black smoke model would require a half terabyte of storage just for the model matrix, whereas we are able to compute with it on a desktop workstation. The best previously available reduced memory footprint method would have required three orders of magnitude more computing time than our new method. Supplementary materials for this article are available online.

ARTICLE HISTORY

Received November 2015
Revised May 2016

KEYWORDS

Air pollution; Big data;
Parallel computing;
Regression; Smoothing



1. Introduction


This article proposes a method for estimating generalized additive models (a particular class of Gaussian latent process models) for much larger datasets and models than has hitherto been possible. For our application we achieve a three order of magnitude speed up relative to previous big data GAM methods (e.g., Wood, Goude, and Shaw 2015). Our new method rests on three innovations: (i) an efficient new fitting iteration, employing a minimal number of matrix operations all of which scale reasonably well, (ii) OpenMP based parallelization of these matrix operations, and (iii) a novel *marginal* covariate discretization scheme, enabling compact model representation and efficient computation of key matrix crossproducts. These three elements work together, and dropping any one of them leads to an increase in fitting time of an order of magnitude or more.

We are motivated by a practical problem in spatial epidemiology: the local estimation of short-term exposure to air pollution, based on monitoring network data. Specifically we focus on the United Kingdom Black Smoke (BS) monitoring network, which collected daily data on $\mu\text{g m}^{-3}$ (micrograms per cubic meter) of BS particulates (largely from coal and Diesel combustion) from

1961 to 2005. The U.K. BS network fluctuated in size with different stations being added and removed over time, peaking at 1269 stations in 1967 but declining to 73 stations by 2005. Figure 1(a) shows the network in 1967, indicating the average log BS measurements in that year. The other panels in Figure 1 illustrate the temporal patterns in the data, and in the network size. In total the data comprise 9,451,232 daily measurements from 2862 monitoring sites.

Because of the data volume, previous attempts to model spatiotemporal patterns in the BS data have focused on annual averages (e.g., Shaddick and Zidek 2014). This is not entirely satisfactory from an epidemiological perspective, since acute respiratory disease is usually sensitive to exposure to high levels of pollution over short time periods, and such exposure can be completely hidden in an annual average. Retrospective cohort studies, for example, really require estimates of exposure at the daily level, rather than annual averages, if they are to successfully uncover acute effects. This difference between acute and long-term exposure is also reflected in the health guidelines, with EU regulations currently stipulating that annual average exposure should not exceed $68\mu\text{g m}^{-3}$ while daily peak exposure should not exceed $213\mu\text{g m}^{-3}$.

CONTACT Simon N. Wood  simon.wood@bristol.ac.uk  School of Mathematics, University of Bristol, Bristol, BS8 1TW United Kingdom. Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JASA.

 Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

© 2017 Simon N. Wood, Zheyuan Li, Gavin Shaddick, and Nicole H. Augustin. Published with license by Taylor & Francis.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

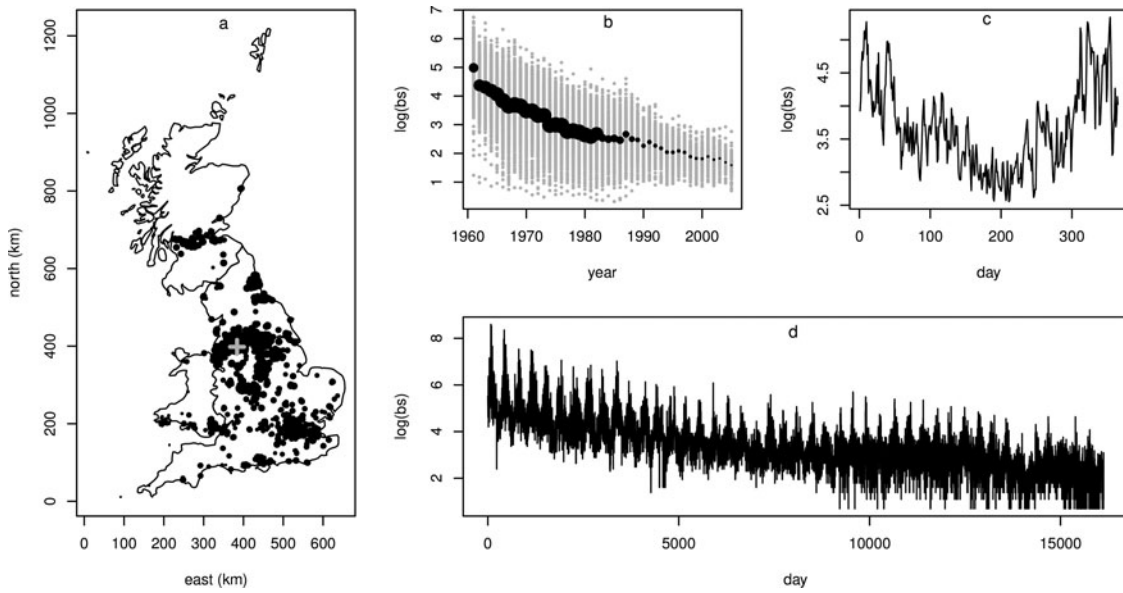


Figure 1. (a) The U.K. Black Smoke Network monitoring network at its largest in 1967. Symbol sizes are proportional to annual average log black smoke. (b) Annual average log black smoke against year. Black dots are averages over space, with dot size proportional to network size. Gray dots are station averages. (c) Daily averages for 1967, across all stations shown in (a). (d) All daily measurements for the longest running site, shown as a gray “+” in (a).

Given the data volume, an obvious option is not to model, but simply to estimate daily exposure directly from the raw measurement, but this is a poor option for several reasons. First, the network design is not random but shows a type of preferential sampling (Shaddick and Zidek 2014), so that a design based approach to exposure estimation will result in bias, which is only avoidable by taking a model-based approach. Second, the reduced number of stations later in the data make spatial predictions difficult without a model that is able to share information across years. Third, there are strong covariate effects.

We will end up using a model structure

$$\begin{aligned}
 \log(\text{bs}_i) &= f_1(y_i) + f_2(\text{doy}_i) + f_3(\text{dow}_i) + f_4(y_i, \text{doy}_i) \\
 &+ f_5(y_i, \text{dow}_i) + f_6(\text{doy}_i, \text{dow}_i) \\
 &+ f_7(n_i, e_i) + f_8(n_i, e_i, y_i) + f_9(n_i, e_i, \text{doy}_i) \\
 &+ f_{10}(n_i, e_i, \text{dow}_i) + f_{11}(h_i) + f_{12}(T_i^0, T_i^1) \\
 &+ f_{13}(\bar{T}_1, \bar{T}_2) \\
 &+ f_{14}(r_i) + \alpha_{k(i)} + b_{\text{id}(i)} + e_i
 \end{aligned} \quad (1)$$

where y , doy and dow denote, year, day of year, and day of week; n and e denote location as kilometers north and east; h and r are height (elevation of station) and cube root transformed rainfall (unfortunately only available as monthly average); T^0 and T^1 are daily minimum and maximum temperature, while \bar{T}_1 and \bar{T}_2 are daily mean temperature on and two days previously; $\alpha_{k(i)}$ is a fixed effect for the site type k of the i th observation (type is one of R (rural), A (industrial), B (residential), C (commercial), D (city/town center), X (mixed), or M (missing)); $b_{\text{id}(i)}$ is a random effect for the i th station, while e_i is a Gaussian error term following an AR process at each site.

Using reduced rank spline basis expansions for the terms in (1) requires around 8000 model coefficients. So estimating the model as a penalized GLM in the manner of Wood (2011) would require half a terabyte of storage just for the model matrix

and is clearly infeasible. Our original intention was to use the method of Wood, Goude, and Shaw (2015) (available in R package `mgcv`) or to follow Shaddick and Zidek (2014) in using the method of Rue, Martino, and Chopin (2009) (via the `INLA` package), however this proved not to be feasible. Even if the computational load had been acceptable in terms of execution time, our experiments with smaller models and datasets suggested that `INLA` would require more than the 128Gb of memory that we had available. The Wood, Goude, and Shaw (2015) method would have been possible in terms of memory footprint, but we estimated that fitting would have taken in excess of a month of computing time (12 core Xeon E5-2670 2.3 GHz CPU), even using an enhanced efficiency version of the method employing some of the ideas from the current article for REML smoothing parameter selection. Using just the published method would have required approximately five times as long.

After reviewing model representation in Section 2, we develop a practical fitting method in Sections 3 and 4, which reduces the fitting time for model (1) to under an hour. The novel developments that allow this are covered in Section 4 and appendix A. Sections 5 and 5.1 then discuss the black smoke modeling in more detail.

2. Model Class and Representation

We first review the class of generalized additive models (GAM) introduced by Hastie and Tibshirani (1986, 1990) (see also Wahba 1990), relating a univariate response, y_i to predictors x_{ji} (which may be vector). A GAM has the structure

$$y_i \sim \text{EF}(\mu_i, \phi) \text{ where } g(\mu_i) = \mathbf{A}(i, :) \boldsymbol{\theta} + \sum_j f_j(x_{ji}), \quad (2)$$

$\mu_i = E(y_i)$, `EF` denotes an exponential family distribution with known or unknown scale parameter ϕ , g is a known smooth monotonic link function, $\mathbf{A}(i, :)$ the i th row of any parametric model matrix, and $\boldsymbol{\theta}$ the corresponding parameter vector. The f_j are unknown smooth functions to be estimated

(and must usually be subjected to sum-to-zero identifiability constraints).

For estimation purposes we adopt the widely used approach of representing the unknown functions using reduced rank smoothing splines. Full smoothing splines arise from solving variational problems. For example, the cubic spline problem seeks f , from some reproducing kernel Hilbert (or appropriate Sobolev) space, to minimize $\sum_{i=1}^n \{y_i - f(x_i)\}^2 + \lambda \int f''(x)^2 dx$ (λ is a smoothing parameter). The result can be represented in terms of an explicit n -dimensional basis, while the spline penalty becomes a quadratic penalty on the basis coefficients. However, since, at latest, Wahba (1980) and Parker and Rice (1985), it has been recognized that an n -dimensional basis representation is computationally wasteful for negligible statistical gain and use of a $k \ll n$ dimensional basis is often preferable. Theoretical work by Gu and Kim (2002), Hall and Opsomer (2005), Li and Ruppert (2008), Kauermann, Krivobokova, and Fahrmeir (2009), Claeskens, Krivobokova, and Opsomer (2009) and Wang, Shen, and Ruppert (2011) show that the reduced rank approach is well founded, with k needing to grow only rather slowly with sample size (e.g., $k = O(n^{1/5})$ for a cubic spline under REML smoothness estimation).

A rich variety of reduced rank model terms are available in addition to cubic splines. Examples are the P-splines of Eilers and Marx (1996); Marx and Eilers (1998); Ruppert, Wand, and Carroll (2003), and adaptive variants (e.g., Wood 2011), as well as the isotropic thin plate and other Duchon splines (Duchon 1977), for which rank reduction is conveniently performed by the eigen method of Wood (2003). Reduced rank tensor product splines (e.g., Eilers and Marx 2003; Wood 2006) are important for representing smooth interactions, splines on the sphere (Wahba 1981) and Gaussian process smoothers (Kammann and Wand 2003; Handcock, Meier, and Nychka 1994) are useful in some spatial applications. In all cases if $\mathbf{f}_j = [f_j(x_{j1}), f_j(x_{j2}), \dots]^T$ we can write $\mathbf{f}_j = \mathbf{X}_j \boldsymbol{\beta}_j$ where \mathbf{X}_j is an $n \times p_j$ model matrix for the smooth, containing its basis functions evaluated at the observed x_j values. $\boldsymbol{\beta}_j$ is the corresponding coefficient vector. The smoothing penalty for f_j can then be written $\boldsymbol{\beta}_j^T \mathbf{S}_j \boldsymbol{\beta}_j$, where \mathbf{S}_j contains known coefficients. Since the individual f_j in (2) are only estimable to within an intercept term, identifiability constraints need to be applied. As discussed in Wood, Scheipl, and Faraway (2013) the sum-to-zero constraints, $\sum_i f_j(x_{ji}) = 0$ have the advantage of leading to narrow confidence intervals on the constrained f_j , and it is easy to reparameterize to incorporate the constraints directly into \mathbf{X}_j and \mathbf{S}_j (which, respectively, lose a column, and a row and column in the process).

It is then straightforward to create a single $n \times p$ model matrix $\mathbf{X} = (\mathbf{A}, \mathbf{X}_1, \mathbf{X}_2, \dots)$ with corresponding combined parameter vector $\boldsymbol{\beta}$. Given some smoothing parameters $\boldsymbol{\lambda}$ a combined smoothing penalty could then be written as $\sum_j \lambda_j \boldsymbol{\beta}_j^T \mathbf{S}_j \boldsymbol{\beta}_j = \sum_j \lambda_j \boldsymbol{\beta}^T \mathbf{S}_j \boldsymbol{\beta} = \boldsymbol{\beta}^T \mathbf{S}_\lambda \boldsymbol{\beta}$, where \mathbf{S}_j is simply a zero padded version of \mathbf{S}_j and $\mathbf{S}_\lambda = \sum_j \lambda_j \mathbf{S}_j$. Hence, we have an overparameterized GLM structure, $g(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta}$. Given smoothing parameters it is estimated via

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} l(\boldsymbol{\beta}) - \boldsymbol{\beta}^T \mathbf{S}_\lambda \boldsymbol{\beta} / 2. \quad (3)$$

This penalized likelihood approach (e.g., Green and Silverman 1994) can be viewed as a reasonable approach in its own right.

An alternative is to view penalization as the expression of a belief that “smooth is more probable than wiggly” and to express this using the (improper) prior

$$\boldsymbol{\beta} \sim N(\mathbf{0}, \mathbf{S}_\lambda^-),$$

where \mathbf{S}_λ^- is a Moore-Penrose pseudoinverse (\mathbf{S}_λ being rank deficient because the penalties leave some space of functions unpenalized, and in any case do not penalize the fixed effects). In that case $\hat{\boldsymbol{\beta}}$ is the MAP estimator of $\boldsymbol{\beta}$, and it is clear that we can view the GAM as a Gaussian latent random field model (see Kimeldorf and Wahba 1970; Wahba 1983; Silverman 1985; Fahrmeir and Lang 2001; Ruppert, Wand, and Carroll 2003, etc.). The smoothing parameters, $\boldsymbol{\lambda}$, can be estimated by generalized cross-validation or similar (e.g., Craven and Wahba 1979), but Reiss and Ogden (2009) showed that a (restricted) marginal likelihood approach (e.g., Wood 2011) offers practical reliability advantages, in being less prone to multiple local optima and consequent undersmoothing.

3. The Fitting Iteration

The purpose of this article is to allow the rich existing modeling framework, described in Section 2, to be used with much larger models and datasets than has hitherto been possible, by providing substantially new scalable fitting methods. The new methods are based on the performance iteration (Gu 1992) or PQL (Breslow and Clayton 1993) approach to model fitting, modified to obtain reasonable scalability. Before introducing the modifications, we motivate the basic approach and provide an alternative justification for its use, suited to penalized regression.

It is readily shown that maximization of (3) by Fisher scoring is equivalent to the following penalized iteratively reweighted least squares (PIRLS) scheme. Initialize $\hat{\mu}_i = y_i + \delta_i$ and $\hat{\eta}_i = g(\hat{\mu}_i)$ where δ_i is a small constant (often zero) chosen to ensure $g(\hat{\mu}_i)$ exists. Then iterate the following to convergence

1. Form “pseudodata” $z_i = g'(\hat{\mu}_i)(y_i - \hat{\mu}_i) + \hat{\eta}_i$ and weight matrix $\mathbf{W} = \operatorname{diag}(w_i)$ where $w_i^{-1} = V(\hat{\mu}_i)g'(\hat{\mu}_i)^2$.
2. By penalized least squares, estimate $\boldsymbol{\beta}$ for the working model

$$\mathbf{z} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\beta} \sim N(\mathbf{0}, \mathbf{S}_\lambda^-),$$

$$E(\boldsymbol{\epsilon}) = \mathbf{0} \text{ and } E(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) = \boldsymbol{\phi}\mathbf{W}^{-1}.$$

The key idea of performance iteration/PQL is to estimate $\boldsymbol{\lambda}$ and $\boldsymbol{\phi}$ at each iteration from the working model. Consider using restricted marginal likelihood (REML) for this purpose. First suppose that we were to make the clearly false assumption that $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{W}^{-1}\boldsymbol{\phi})$. If $\hat{\boldsymbol{\beta}}_\lambda = \operatorname{argmin}_\beta \|\mathbf{z} - \mathbf{X}\boldsymbol{\beta}\|_w^2 / \boldsymbol{\phi} + \boldsymbol{\beta}^T \mathbf{S}_\lambda \boldsymbol{\beta}$, where $\|\mathbf{x}\|_w^2 = \mathbf{x}^T \mathbf{W} \mathbf{x}$ and M is the dimension of the null space of \mathbf{S}_λ , then the twice negative log REML (e.g. Wood 2011) is

$$\begin{aligned} \mathcal{V}(\boldsymbol{\lambda}) = & \|\mathbf{z} - \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda\|_w^2 / \boldsymbol{\phi} + \hat{\boldsymbol{\beta}}_\lambda^T \mathbf{S}_\lambda \hat{\boldsymbol{\beta}}_\lambda + \log |\mathbf{X}^T \mathbf{W} \mathbf{X} / \boldsymbol{\phi} + \mathbf{S}_\lambda| \\ & - \log |\mathbf{S}_\lambda|_+ + n \log(\boldsymbol{\phi}) + (n - M) \log(2\pi). \end{aligned} \quad (4)$$

Differentiating \mathcal{V} with respect to $\boldsymbol{\phi}$ and equating to zero, we find that the REML estimate of $\boldsymbol{\phi}$ must satisfy

$$\hat{\boldsymbol{\phi}} = \frac{\|\mathbf{z} - \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda\|_w^2}{n - \tau}, \quad (5)$$

where $\tau = \text{tr}\{(\mathbf{X}^T \mathbf{W} \mathbf{X} / \hat{\phi} + \mathbf{S}_\lambda)^{-1} \mathbf{X}^T \mathbf{W} \mathbf{X} / \hat{\phi}\}$ is the “effective degrees of freedom” of the model. So $\hat{\phi}$ is simply the “Pearson estimator” of the scale parameter, which is a reasonable estimator without any REML justification, and without assuming normality of \mathbf{z} (see, e.g., Wahba 1983; McCullagh and Nelder 1989; Hastie and Tibshirani 1990).

Now let us eliminate the false assumption of normality of \mathbf{z} , replacing it with central limit theorem justification. Consider the QR decomposition $\sqrt{\mathbf{W} \mathbf{X}} = \mathbf{Q} \mathbf{R}$, where \mathbf{Q} has orthogonal columns and \mathbf{R} is upper triangular (this decomposition is purely a theoretical device, nowhere in the new methods below do we actually need to compute a QR decomposition). Define $\mathbf{f} = \mathbf{Q}^T \sqrt{\mathbf{W}} \mathbf{z}$, $r = \|\mathbf{z}\|_w^2 - \|\mathbf{f}\|^2$. In that case $\mathbf{X}^T \mathbf{W} \mathbf{X} = \mathbf{R}^T \mathbf{R}$, $\|\mathbf{z} - \mathbf{X} \hat{\boldsymbol{\beta}}_\lambda\|_w^2 = \|\mathbf{f} - \mathbf{R} \boldsymbol{\beta}\|^2 + r$, and we have the alternative working model

$$\mathbf{f} = \mathbf{R} \boldsymbol{\beta} + \mathbf{e}, \quad \boldsymbol{\beta} \sim N(\mathbf{0}, \mathbf{S}_\lambda^-) \text{ and } \mathbf{e} \sim N(\mathbf{0}, \mathbf{I}\phi), \quad (6)$$

where the multivariate central limit theorem justifies $\mathbf{e} \sim N(\mathbf{0}, \mathbf{I}\phi)$ as an $n/p \rightarrow \infty$ approximation. The twice negative log restricted marginal likelihood for this model is

$$\begin{aligned} \mathcal{V}_r(\lambda) = & \|\mathbf{f} - \mathbf{R} \hat{\boldsymbol{\beta}}_\lambda\|^2 / \phi + \hat{\boldsymbol{\beta}}_\lambda^T \mathbf{S}_\lambda \hat{\boldsymbol{\beta}}_\lambda + \log |\mathbf{R}^T \mathbf{R} / \phi + \mathbf{S}_\lambda| \\ & - \log |\mathbf{S}_\lambda|_+ + p \log \phi + (p - M) \log(2\pi). \end{aligned}$$

For a given ϕ , \mathcal{V} and \mathcal{V}_r differ only by an additive constant, and therefore result in identical inference about λ and $\boldsymbol{\beta}$. Inference about ϕ would of course differ, since r carries information about ϕ , but if we plug the Pearson estimate (5) into \mathcal{V}_r then we obtain identical inference to that obtained by simply using \mathcal{V} for ϕ and λ . This justifies use of (4) for λ , ϕ estimation.

Note that once the coefficients and smoothing parameters are estimated, further inference can be based on the large sample Bayesian result,

$$\boldsymbol{\beta} \sim N(\hat{\boldsymbol{\beta}}, (\mathbf{X}^T \mathbf{W} \mathbf{X} / \phi + \mathbf{S}_\lambda)^{-1}), \quad (7)$$

which turns out to provide well calibrated frequentist inference (Wahba 1983; Silverman 1985; Nychka 1988; Marra and Wood 2012; Wood 2013).

4. A Practical Fitting Method

Implementation of the fitting iteration of Section 3 is limited by several practical considerations.

1. For the target datasets and models, it is impractical to explicitly form \mathbf{X} whole.
2. The log determinant terms in \mathcal{V} are potentially numerically unstable. Because having some $\lambda_j \rightarrow \infty$ is legitimate in GAM estimation, \mathbf{S}_λ can become so badly scaled that the computation of log determinants involves taking the logs of terms that are numerically zero.
3. For maximal efficiency it is not sensible to optimize \mathcal{V} at each iteration step, when it will anyway be modified at the next step.
4. The update step for \mathcal{V} should involve computations that scale well to multi-core computation.

Wood, Goude, and Shaw (2015) addressed 1 by iteratively updating the QR factorization of \mathbf{X} , and then applying the method of

Wood (2011) to (6). This approach ignored 3, requires pivoted QR decomposition and addressed 2 by stabilizing reparameterizations involving $p \times p$ symmetric eigen decomposition: the QR and eigen decompositions do not scale well. For example the state of the art block pivoted QR decomposition of Quintana-Ortí, Sun, and Bischof (1998), only has around half the floating point operations as matrix-matrix computations. In consequence the Wood, Goude, and Shaw (2015) was computationally impractical for the black smoke model. See appendix C for a discussion of the issues around multicore computing.

Our proposal here addresses 3 by taking a single Newton step to update $\boldsymbol{\rho} = \log(\lambda)$ at each cycle of the iteration (rather than fully optimizing \mathcal{V} at each cycle). We propose to avoid the stabilizing reparameterization step by avoiding evaluation of the log determinants altogether (hence, addressing 2). This is based on the observation that the Newton step, $\boldsymbol{\Delta}$, only involves the derivative of \mathcal{V} , and the derivatives of the log determinants are less numerically problematic. Evaluation of \mathcal{V} is usually required to ensure that the Newton step results in an improvement of \mathcal{V} . We cannot skip such a check, but we can substitute the alternative check that $\boldsymbol{\Delta}^T \nabla \mathcal{V}(\boldsymbol{\rho} + \boldsymbol{\Delta}) \leq 0$, that is, that \mathcal{V} is nonincreasing in the direction of $\boldsymbol{\Delta}$ at the end of the Newton step (see, e.g., Wood, 2015, sec. 5.1.1).

Adopting this approach we find that the derivatives of \mathcal{V} can be obtained using simple matrix operations and a pivoted Cholesky decomposition of $\mathbf{X}^T \mathbf{W} \mathbf{X}$, which can be accumulated blockwise, thereby dealing with 1. Lucas (2004) provides a block oriented pivoted Cholesky decomposition readily parallelized using openMP (OpenMP Architecture Review Board, 2008), which deals with point 4. The resulting method has the further advantage that, with some further work, it turns out to be possible to produce further substantial efficiency savings by discretization of the model covariates (see Section 4.5).

4.1. The Modified Fitting Iteration

Based on the above considerations, the proposed fitting iteration is as follows. Its convergence properties are discussed in Appendix B.

- Perform the term by term reparameterization described in Section 4.3.
- Initialize $\boldsymbol{\rho}_0$, $\boldsymbol{\Delta}_0 = \mathbf{0}$, $\hat{\mu}_i = y_i + \delta_i$ and $\hat{\eta}_i = g(\hat{\mu}_i)$. δ_i is 0 or a small value chosen to ensure that $\hat{\eta}_i$ exists.
- Repeat...
 1. Accumulate $\mathbf{X}^T \mathbf{W} \mathbf{X}$, $\mathbf{f} = \mathbf{X}^T \mathbf{W} \mathbf{z}$ and penalized deviance, D . $z_i = g'(\hat{\mu}_i)(y_i - \hat{\mu}_i) + \hat{\eta}_i$ and \mathbf{W} is diagonal with entries $w_i = \{V(\hat{\mu}_i)g'(\hat{\mu}_i)^2\}^{-1}$.
 2. Test for convergence, terminate if achieved.
 3. Except at iteration one, if $D^*/\phi + \boldsymbol{\beta}^{*T} \mathbf{S}_\rho \boldsymbol{\beta}^* < D/\phi + \hat{\boldsymbol{\beta}}^T \mathbf{S}_\rho \hat{\boldsymbol{\beta}}$ set $\hat{\boldsymbol{\beta}} \leftarrow (\boldsymbol{\beta}^* + \hat{\boldsymbol{\beta}})/2$ and return to 1.
 4. $\boldsymbol{\beta}^* \leftarrow \hat{\boldsymbol{\beta}}$.
 5. $\boldsymbol{\rho} = \boldsymbol{\rho}_0 + \boldsymbol{\Delta}_0$.
 6. Given $\mathbf{X}^T \mathbf{W} \mathbf{X}$, \mathbf{f} and $\boldsymbol{\rho}$, obtain $\boldsymbol{\Delta}$, the Newton step for the working model, $\nabla \mathcal{V}$ the gradient of the working REML and $\hat{\boldsymbol{\beta}}$.
 7. If $\nabla \mathcal{V}^T \boldsymbol{\Delta}_0 > \epsilon D$ then $\boldsymbol{\Delta}_0 \leftarrow \boldsymbol{\Delta}_0/2$ and return to 5.

8. $\Delta_0 \leftarrow \Delta$, $\rho_0 \leftarrow \rho$, $D^* \leftarrow D$. Form $\hat{\eta} = \mathbf{X}\hat{\beta}$ and $\hat{\mu}_i = g^{-1}(\hat{\eta}_i)$.

Note that Step 1 does not require the explicit formation of the whole matrix \mathbf{X} . Step 3 reduces the β step taken if the Newton step was too long, in that it increased the penalized deviance at the ρ value at which it was computed. Step 5 reduces the ρ step if it was so long that the REML score was increasing at the end of the step. When $\log \phi$ is unknown it can be included as an extra element of ρ .

Step 6 consists of estimating the $\hat{\beta}_\lambda$ implied by the proposed ρ and the current \mathbf{W} and \mathbf{z} . Further more the marginal likelihood of the working penalized linear model is uses as a smoothing parameter estimation criterion, and the gradient vector of this criterion along with the first Newton step for optimizing it are also computed. The next sections detail how Step 6 is accomplished.

4.2. The REML Update

Now consider the calculation of the Newton step, Δ , to improve (4). We have that $\hat{\beta}_\lambda$ is the solution of $(\mathbf{X}^T\mathbf{W}\mathbf{X} + \phi\mathbf{S}_\lambda)\beta_\lambda = \mathbf{X}^T\mathbf{W}\mathbf{z}$. The actual computation proceeds by taking a Cholesky decomposition $\mathbf{R}^T\mathbf{R} = \mathbf{X}^T\mathbf{W}\mathbf{X}/\phi + \mathbf{S}_\lambda$ using a parallel version of Lucas (2004). This is usually done with pivoting, in which case the rank of \mathbf{R} is then estimated and unidentifiable parameters set to zero and dropped from subsequent computations. We then compute $\hat{\beta}_\lambda = \mathbf{R}^{-1}\mathbf{R}^{-T}\mathbf{X}^T\mathbf{W}\mathbf{z}/\phi$ (by backward and forwards substitution). In what follows “pivoting” and “unpivoting” refer to the application of the Cholesky pivoting order and its reversal.

The Newton step is $\Delta = -(\mathrm{d}^2\mathcal{V}/\mathrm{d}\rho\mathrm{d}\rho^T)^{-1}\mathrm{d}\mathcal{V}/\mathrm{d}\rho$, where $\mathrm{d}^2\mathcal{V}/\mathrm{d}\rho\mathrm{d}\rho^T$ will have been perturbed if necessary to ensure definiteness (see Nocedal and Wright 2006). Recalling that $\mathrm{d}(\|\mathbf{z} - \mathbf{X}\beta\|_w^2/\phi + \beta^T\mathbf{S}_\lambda\beta)/\mathrm{d}\beta|_{\hat{\beta}_\lambda} = \mathbf{0}$, we have

$$\frac{\mathrm{d}\mathcal{V}}{\mathrm{d}\rho_j} = \lambda_j \hat{\beta}_\lambda^T \mathbf{S}_j \hat{\beta}_\lambda + \frac{\mathrm{d} \log |\mathbf{X}^T \mathbf{W} \mathbf{X} / \phi + \mathbf{S}_\lambda|}{\mathrm{d}\rho_j} - \frac{\mathrm{d} \log |\mathbf{S}_\lambda|_+}{\mathrm{d}\rho_j} \quad (8)$$

and, defining $\delta_k^j = 1$ if $k = j$ and 0 otherwise,

$$\begin{aligned} \frac{\mathrm{d}^2\mathcal{V}}{\mathrm{d}\rho_j\mathrm{d}\rho_k} &= 2 \frac{\mathrm{d}\hat{\beta}_\lambda^T}{\mathrm{d}\rho_k} (\mathbf{X}^T\mathbf{W}\mathbf{X}/\phi + \mathbf{S}_\lambda) \frac{\mathrm{d}\hat{\beta}_\lambda}{\mathrm{d}\rho_j} \\ &+ 2\lambda_j \hat{\beta}_\lambda^T \mathbf{S}_j \frac{\mathrm{d}\hat{\beta}_\lambda}{\mathrm{d}\rho_k} + 2\lambda_k \hat{\beta}_\lambda^T \mathbf{S}_k \frac{\mathrm{d}\hat{\beta}_\lambda}{\mathrm{d}\rho_j} + \delta_k^j \lambda_j \hat{\beta}_\lambda^T \mathbf{S}_j \hat{\beta}_\lambda \\ &+ \frac{\mathrm{d}^2 \log |\mathbf{X}^T \mathbf{W} \mathbf{X} / \phi + \mathbf{S}_\lambda|}{\mathrm{d}\rho_j\mathrm{d}\rho_k} - \frac{\mathrm{d}^2 \log |\mathbf{S}|_+}{\mathrm{d}\rho_j\mathrm{d}\rho_k}. \end{aligned}$$

Implicit differentiation implies that

$$\frac{\mathrm{d}\hat{\beta}_\lambda}{\mathrm{d}\rho_j} = -\lambda_j \mathbf{R}^{-1} \mathbf{R}^{-T} \mathbf{S}_j \hat{\beta}_\lambda.$$

This latter computation is most efficient if $\hat{\beta}_\lambda$ is first unpivoted, $\mathbf{S}_j \hat{\beta}_\lambda$ is formed and this is then repivoted: the block structure of \mathbf{S}_j (see next section) can then be exploited. The next two sections cover computation of the derivatives of the log determinants.

4.3. Computing The Derivatives Of $\log |\mathbf{S}_\lambda|_+$

\mathbf{S}_λ has block diagonal structure that can be exploited. For example, denoting zero blocks by \circ ,

$$\mathbf{S}_\lambda = \begin{pmatrix} \lambda_1 \mathbf{S}_1 & \circ & \circ & \circ \\ \circ & \lambda_2 \mathbf{S}_2 & \circ & \circ \\ \circ & \circ & \sum_j \lambda_j \mathbf{S}_j & \circ \\ \circ & \circ & \circ & \circ \end{pmatrix}.$$

That is there are some blocks with single smoothing parameters, and others with a more complicated additive structure. There are usually also some zero blocks on the diagonal. The block structure means that the generalized determinant and its derivatives w.r.t. $\rho_k = \log \lambda_k$ can be computed blockwise. Note in particular that, for the above example,

$$\begin{aligned} \log |\mathbf{S}_\lambda|_+ &= \text{rank}(\mathbf{S}_1) \log(\lambda_1) + \log |\mathbf{S}_1|_+ + \text{rank}(\mathbf{S}_2) \log(\lambda_2) \\ &+ \log |\mathbf{S}_2|_+ + \log \left| \sum_j \lambda_j \mathbf{S}_j \right|_+ + \dots \end{aligned}$$

For any ρ_k relating to a single parameter block we have

$$\frac{\mathrm{d} \log |\mathbf{S}|_+}{\mathrm{d}\rho_k} = \text{rank}(\mathbf{S}_k)$$

and zero second derivatives. For multi- λ blocks there will generally be first and second derivatives to compute. There are no second derivatives “between-blocks.” To facilitate computations some prefit reparameterization is undertaken, according to the type of block.

1. Single parameter diagonal blocks. These can be reparameterized so that all nonzero elements are one, and the rank precomputed.
2. Single parameter dense blocks. These can be reparameterized to look like the previous type, by similarity transform, again computing rank.
3. Multi- λ blocks are transformed so that $\sum_j \lambda_j \mathbf{S}_j$ has full rank in the new parameterization. Again a similarity transform is used. Typically the \mathbf{S}_j are of smaller dimension in the reparameterization and consequently an extra zero block is introduced on the diagonal of \mathbf{S}_j .

The generalized determinant of type 3 blocks becomes an ordinary determinant of $\sum_j \lambda_j \mathbf{S}_j$ after reparameterization. Hence, its derivatives follow from the standard result

$$\frac{\mathrm{d} \log |\mathbf{S}|}{\mathrm{d}\rho} = \text{tr} \left(\mathbf{S}^{-1} \frac{\partial \mathbf{S}}{\partial \rho} \right).$$

4.4. Computing The Derivatives of $\log |\mathbf{X}^T\mathbf{W}\mathbf{X}/\phi + \mathbf{S}_\lambda|$

The following computations build on the Cholesky decomposition of the previous sections

1. Form $\mathbf{P} = \mathbf{R}^{-1}$, and unpivot the rows of \mathbf{P} . Then form $\mathbf{P}\mathbf{P}^T$. These steps are $O(p^3)$, but can be parallelized.
2. Form the matrices containing the nonzero rows of $\mathbf{S}_k\mathbf{P}\mathbf{P}^T$ ($\forall k$). This step is cheap for all but type 3 blocks.
3. Compute the required derivatives using

$$\frac{\mathrm{d} \log |\mathbf{X}^T \mathbf{W} \mathbf{X} / \phi + \mathbf{S}_\lambda|}{\mathrm{d}\rho_k} = \lambda_k \text{tr}(\mathbf{S}_k \mathbf{P} \mathbf{P}^T)$$

and

$$\frac{d^2 \log |\mathbf{X}^\top \mathbf{W} \mathbf{X} / \phi + \mathbf{S}_\lambda|}{d\rho_k d\rho_j} = \delta_k^j \lambda_k \text{tr}(\mathbf{S}_k \mathbf{P} \mathbf{P}^\top) - \lambda_j \lambda_k \text{tr}(\mathbf{S}_k \mathbf{P} \mathbf{P}^\top \mathbf{S}_j \mathbf{P} \mathbf{P}^\top).$$

Note that $\mathbf{P} \mathbf{P}^\top = (\mathbf{X}^\top \mathbf{W} \mathbf{X} / \phi + \mathbf{S}_\lambda)^{-1}$, the Bayesian covariance matrix.

The trace computations in step 3 are very efficient, given the block structure of the \mathbf{S}_k , if we employ the following tricks. In general $\text{tr}(\mathbf{A} \mathbf{B}) = \sum_{k,j} A_{kj} B_{jk}$. Now let \mathbf{A} have nonzero rows only between k_1 and k_2 , while \mathbf{B} has nonzero rows only between j_1 and j_2 .

$$\text{tr}(\mathbf{A}) = \sum_{k=k_1}^{k_2} A_{kk} \text{ and } \text{tr}(\mathbf{A} \mathbf{B}) = \sum_{k=k_1}^{k_2} \sum_{j=j_1}^{j_2} A_{kj} B_{jk}.$$

Of course normally the initial zero rows would not actually be stored in which case we have

$$\text{tr}(\mathbf{A}) = \sum_{k=k_1}^{k_2} A_{k-k_1,k} \text{ and } \text{tr}(\mathbf{A} \mathbf{B}) = \sum_{k=k_1}^{k_2} \sum_{j=j_1}^{j_2} A_{k-k_1,j} B_{j-j_1,k}.$$

4.5. The Model Matrix: Efficient Storage and Computation

We are interested in computing with models in which it is impractical to store the whole model matrix, and in which computing the required matrix cross product may be prohibitively expensive. For this reason we discretize the model covariates so that the columns of the model matrix corresponding to a single smooth term can be stored in compact form. Specifically, suppose that the covariate for the j th term is discretized into m discrete values, then the model matrix columns for that term can be written as

$$\mathbf{X}_j(i, j) = \bar{\mathbf{X}}_j(k(i), j),$$

where $\bar{\mathbf{X}}_j$ has only m rows and k is an index vector. Storing $\bar{\mathbf{X}}_j$ and \mathbf{k} uses much less memory than storing \mathbf{X}_j directly. This idea is introduced in Lang et al. (2014) to obtain efficient storage and computation for large datasets. However, in that article they employ smooths of one covariate and only require terms of the form $\mathbf{X}_j^\top \mathbf{W} \mathbf{X}_j$, but not $\mathbf{X}_j^\top \mathbf{W} \mathbf{X}_k$. For smoothing parameter estimation we require these ‘‘off diagonal’’ product terms as well. In addition we require tensor product smooths of multiple covariates. Discretizing multiple covariates onto multidimensional grids requires either substantial storage or substantial approximation error, and in the tensor product context it makes sense to instead discretize each component marginal model matrix separately, constructing the full tensor product model matrix ‘‘on the fly.’’

Appendix A develops the identities and algorithms required to compute with \mathbf{X} and its products when the submatrices of \mathbf{X} corresponding to individual terms are stored compactly, and when tensor product terms are computed ‘‘on-the-fly’’ from compactly stored marginal model matrices. With the correct structuring each matrix inner product is a factor of p faster than it would be under direct computation, where p is the number of columns in the largest marginal model matrix

involved in the product and for nontensor product smooths their only model matrix is their single model matrix. The crucial advance over Lang et al. (2014) is the ability to deal with tensor product smooths efficiently, and to compute the off diagonal crossproducts efficiently (between single smooths, tensor product smooths or a mixture of the two). Our method has the major advantage over alternative discretization approaches (e.g., Helwig and Ma 2016) of discretizing covariates independently (marginally), rather than discretizing jointly so that the unique combinations of discretized covariates are stored (or the basis functions evaluated at those unique combinations). The joint approach typically requires more storage, and/or coarser discretization than our fully marginal approach.

An obvious question is how fine a discretization is necessary? Suppose we discretize n observations of covariate x onto a regular grid of m values (just covering the x range). In the large m limit an upper bound on the resulting approximation error is $0.5m^{-1} \max |g'(x)|$ where g is the true function we are trying to recover. The sampling error on the estimate of g is at best $O(n^{-1/2})$, implying that $m = O(n^{1/2})$ is more than adequate. For any finite sample analysis the approximation error bound can be evaluated to check the adequacy of m . Note however, that for the black smoke network data, many covariates are already discrete: for example, there are only a finite number of site locations, site labels and elevations, temperature is only recorded to within 0.1°C , etc.

5. Black Smoke Model Development

Following the industrial revolution, problems associated with air pollution worsened in many countries. During the first half of the 20th century major pollution episodes occurred in London, notably in 1952 an episode of fog, in which levels of black smoke exceeded $4500 \mu\text{g m}^{-3}$, was associated with 4000 excess deaths (Ministry of Health 1954). Other early episodes, which were caused by a combination of industrial pollution sources and adverse weather conditions, and resulted in large numbers of deaths among the surrounding populations, include those in the Meuse valley (Firket 1936) and the United States (Ciocco and Thompson 1961). Attempts to measure levels of air pollution in a regular and systematic way arose as a result of these episodes. In 1961 the world’s first coordinated national air pollution monitoring network was established in the United Kingdom, to monitor black smoke and sulphur dioxide at around 1000 sites (Clifton 1964). Since then all European countries have established monitoring networks, some of them run at the national level, others by local authorities or municipalities, with the initial focus on black smoke (soot) and sulfur dioxide, initially largely from coal burning but shifting more recently to other pollutants. Monitoring has increased in the wake of national and international legislation and the issuing of air quality guidelines, but most monitoring networks share features of the U.K. BS network that challenge the interpretation of the data for epidemiological and policy purposes: (i) monitoring is expensive and so monitoring networks are typically sparse and change over time, (ii) concentrations may vary greatly over small distances, especially in urban areas and (iii) networks designed to monitor compliance with standards, may not give a good representation of levels over an area. Modeling offers the possibility

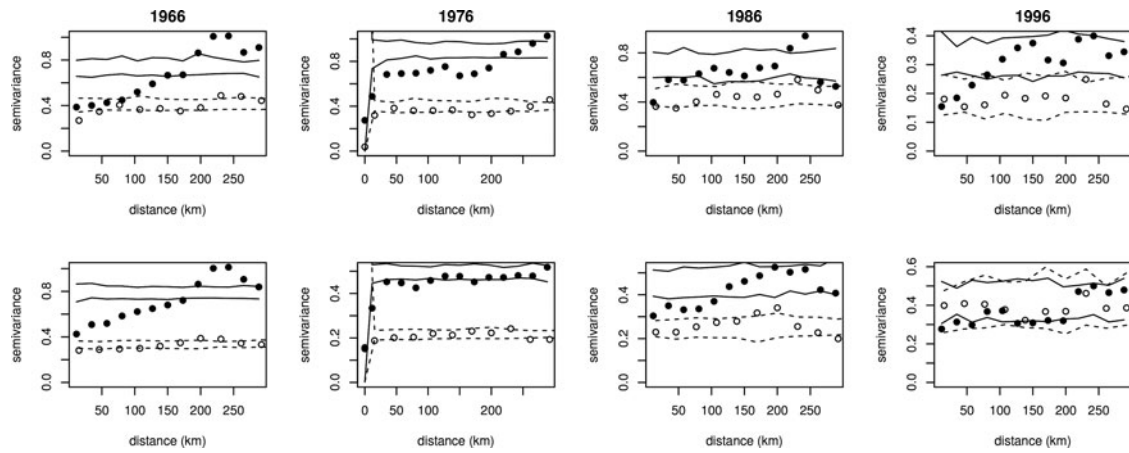


Figure 2. Semivariograms for the 40th (top row) and 200th (bottom row) days of years 1966, 76, 86, and 96, checking for residual spatial autocorrelation. Each plot shows the empirical semivariogram for the log black smoke measurements as black dots, with the corresponding reference bands under zero autocorrelation as black lines. The white dots and dotted lines show the equivalent for the residuals of model (1). The reduction of the network in later years leads to wide reference bands, but in all plots the model appears to offer a reasonable representation of the spatial pattern.

to alleviate these problems, at least partially, and our approach to the U.K. black smoke data should be applicable to other monitoring networks.

In addition to the Black smoke data (Loader 2002), we obtained daily temperature and monthly rainfall data for the United Kingdom (Perry and Hollis 2005b, 2005a) to use as covariates, alongside site elevation (of Terrain-50 2015). Given the volume of data, our initial exploratory model development concentrated first on modeling space without time, and then time without space. In this way we were able to develop candidate temporal decompositions (in terms of year, day of year, and day of week), and candidate models for covariates and space, which were then combined while allowing space and time effects to interact.

Our basic approach was first to decompose the black smoke signal into components dependent on different temporal scales: year (γ) for the long term changes, day of year (doy) for the annual cycle and day of week (dow) for the working week related cycle. These are represented by $f_1 - f_3$ in model (1). These effects were all allowed to interact: for example, the weekly pattern could change with time of year, and over longer timescales. These interactions are $f_4 - f_6$ in model (1). We then allowed the effects of year, time of year and day of the week to vary spatially (terms $f_8 - f_{10}$), as well as allowing a “main effect” of space, f_7 . Elevation and rainfall effects f_{11} and f_{14} were also included alongside effects for site type and a site specific random effect. Residual analysis for a model including only these effects suggested strong temperature dependence, with an interaction of daily minimum and maximum temperatures (f_{12}). Including this latter term still left a correlation with mean temperatures at lags of one and two days, resulting in f_{13} .

Main effects of time were represented using cubic regression splines for γ and cyclic cubic regression splines for doy and dow . Tensor product smooths (e.g., Wood 2006) were used for the interactions. In cases in which smooth main effects and interactions were present, then the interaction smooths were constructed to exclude the main effects, by the simple expedient of applying sum-to-zero constraints to the marginal bases of the tensor product smooth, prior to construction of the tensor product basis. Space time interaction terms follow

Augustin et al. (2009), that is tensor product smoothers with isotropic smoothers used for the spatial marginal smooth and cubic splines for the temporal margin.

Due to the marked reduction in the size of the network in its last decade, and the uneven spatial coverage, some care is required in the specification of the two-dimensional spatial smoothers of n and e , to avoid extrapolation artifacts in later years. We chose to use Duchon splines (see Duchon 1977; Miller and Wood 2014), using first derivative penalties with Duchon’s s parameter therefore set to 1/2. The use of first derivative penalties means that such smoothers are smoothing toward the constant function, which is a reasonable modeling assumption for black smoke data in sparsely observed regions. Duchon splines are the general class of splines introduced in Duchon (1977) of which the popular thin plate spline is a special case: see Miller and Wood (2014) for an accessible introduction. For comparison we also tried Gaussian process smoothers with a Matérn covariance function following Kammann and Wand (2003) and Handcock, Meier, and Nychka (1994), as well as thin plate splines, but in both cases basic model checking revealed artifacts in model predictions toward the end of the data. The online supplementary material includes an animation of predicted log black smoke, clearly illustrating such artifacts for the thin plate spline based model (the equivalent animation for the Duchon spline based model is also included).

Given our interest in using the model for prediction away from the stations, we aimed to keep the station specific random effects structure of the model as simple as possible, however it proved impossible to achieve an adequate fit without any random effects at all, and the model therefore includes a single random intercept term per station, reflecting the individual idiosyncrasies of station locations not captured by the available covariates.

Model adequacy was checked using standard residual plots, as well as autocorrelation function plots and semivariogram plots to check for unmodeled spatial and temporal correlation. Figures 2 and 3 show such plots for model 1, showing that the model does a reasonable job of capturing spatial and temporal correlation, in the data. Further plots are shown in the online supplementary material. To illustrate the importance of

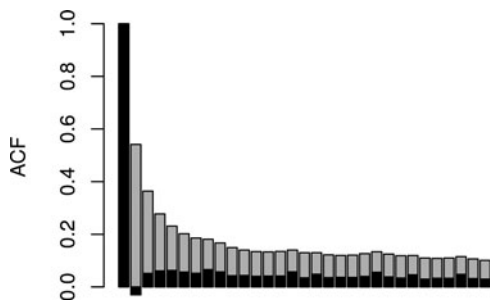


Figure 3. Aggregate ACF for model (1) residuals assuming independent residuals in gray, with the equivalent for the standardized residuals assuming AR1 residuals, overlaid in black. While not perfect, the AR1 model greatly reduces the unmodeled temporal autocorrelation.

the weather variables and site-specific random effects, models were fitted without these leading to AIC increases of 1.6×10^6 and 2.4×10^6 for models without weather variables and the random effect, respectively (the corresponding r^2 reductions were approximately 2% and 1%).

A concern with these data is that they show evidence of a type of preferential sampling (Shaddick and Zidek 2014): as the network was reduced over time, monitors in areas of low concentrations were more likely to be dropped than those in high pollution areas (note that this is different in nature to preferential sampling considered by Diggle, Menezes, and Su (2010), for example). If we had a perfect model without penalties (smoothing priors) then this preferential sampling might reduce efficiency but would not introduce bias. However, when using penalties there is a danger that the reduction of the network so reduces the coverage over some space-time regions that the model predictions for these regions are dominated by the influence of the penalty. If the network reduction is subject to preferential sampling, then it is possible that these space-time regions are systematically those in which pollution is actually lowest, and that the reliance on the penalty/prior then introduces systematic positive bias.

To investigate the potential for such effects, we fitted a reduced model (1) to the data from the year with the most complete spatial coverage, 1967, dropping all terms involving long-term effects of time. We also dropped the temperature and rainfall effects, to force the spatial effects to do as much of the explanatory work as possible. Using the actual network design (i.e., with stations added and dropped over time), we then simulated from a model in which the 1967 fitted model

spatiotemporal pollution fields were repeated each year, but with a long-term decay matching the full dataset. Station-specific random effects were added with standard deviations as estimated from our fit of (1) to the full dataset. Further details are given in the online supplementary material. So our simulated data comes from a “truth” that maintains a degree of spatiotemporal complexity driven by the most “spatially complete” year throughout the simulated dataset, and in which the sampling is given by the real network evolution and, therefore, preferentially drops stations from low pollution regions of the simulation. We then fitted the complete model (1) to the simulated data, and examined its ability to reconstruct the simulated “true” pollution field at each of the locations of stations present in 1967, throughout the whole modeling period (i.e., without any drop out). If our model is sensitive to the preferential sampling evident in the network evolution, then we should be able to detect a positive bias in the full model predictions, which would be likely to grow over time. In fact we can only detect a very small constant bias of about 0.006 on the log scale (corresponding to a 0.6% bias on the original scale). There is no evidence for a trend in the bias: the online supplementary material includes a plot illustrating this and a fuller discussion.

5.1. Results and Predictions

The model (1) has a conditional r^2 of 0.79 (i.e., treating the AR process as induced by a random field), and a marginal r^2 of 0.7 (i.e., ignoring the auto-regressive structure of the residuals). The online supplementary material includes an animation showing the evolution of the predicted spatial pollution field over time. Careful examination shows some artifacts in the fields, usually in coastal regions away from observation stations, but otherwise the results appear reasonable, predicting high pollution levels in the industrial centers especially in the first decade or so, generally showing cleaner air in wetter regions, and tending to show an annual cycle reflecting higher fossil fuel use in the winter.

In this section we illustrate the model results with two sets of plots examining how the chance of exceeding current daily recommended limits ($213 \mu\text{g m}^{-3}$) has changed over time. Figure 4 shows the log of the number of days for which levels are predicted to exceed the daily limit, for a town center location, for several years in the 1960s. These figures are obtained by simply counting up predicted exceedance days by 5 km^2 .

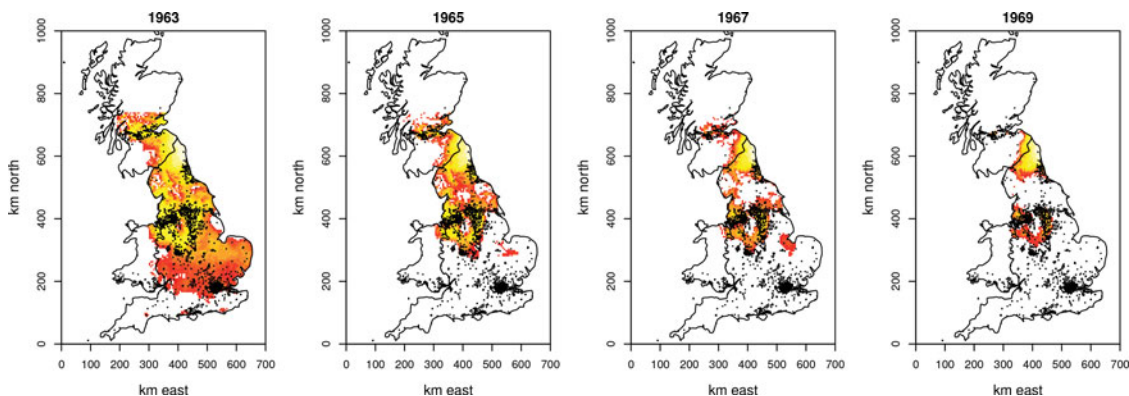


Figure 4. Image plots of log predicted number of days exceeding the EU daily exposure threshold for town center locations for several years in the 1960s. By 1975 there were essentially no exceedance days predicted.

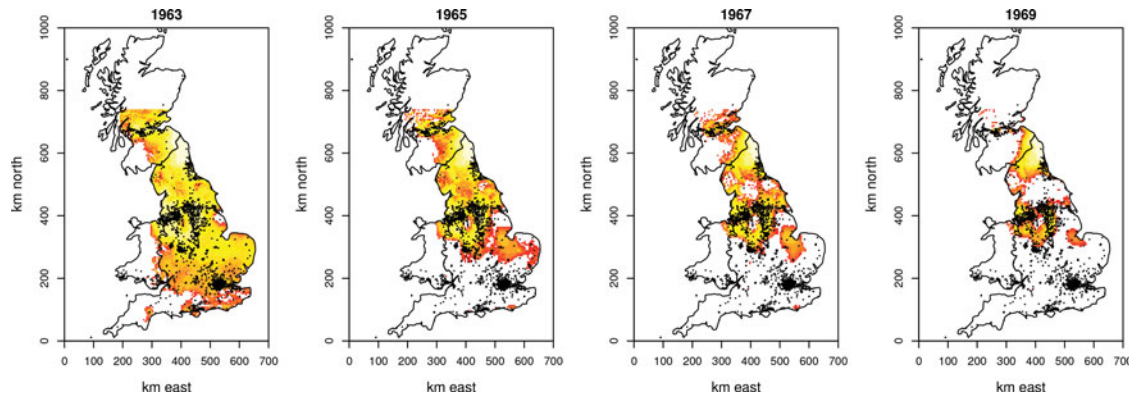


Figure 5. Image plots of log average probability of exceeding the EU daily exposure threshold for town center locations for several years in the 1960s. Red is -6 corresponding to less than one exceedance day expected per year, while the top of the scale is 0.

An alternative is to compute the average posterior probability of the mean exceeding the recommended level using the predicted level and its standard deviation, based on (7). Figure 5 shows such a plot. Broadly both figures show the same pattern, with the situation improving rapidly in London in the wake of the U.K. Clean Air Acts, but taking much longer to improve in the cold northern industrial conurbations.

6. Discussion

Our development of scalable additive model fitting methods rests on three innovations: (i) the development of a fitting method which required only basic easily parallelized matrix computations and a pivoted Cholesky decomposition; (ii) the use of a scalable parallel block pivoted Cholesky algorithm; and (iii) an efficient approach to model matrix storage and computations with the model matrix, using discretized covariates. The approach allows much larger additive/latent Gaussian process models of much larger datasets than has hitherto been feasible, and is general enough for routine use (see R package `mgcv`). For the black smoke modeling, fitting is three orders of magnitude faster than we could have achieved otherwise.

The three method innovations are interlinked so that cleanly attributing elements of the speed up to each separately is not really possible. However, model fitting time increases from around 55 min to over 7.5 hr if we use a single core, instead of 12 (CPU turbo modes disabled to aid comparability). Using the new method, profiling reveals that the time spent on the matrix crossproduct is approximately equal to the time spent on the other method steps, for the black smoke model. From the operations counts in Appendix A the crossproduct is around a factor of 10^2 less floating point intensive using the new discrete methods relative to direct crossproduct formation, while the subleading order cost of basis function evaluation is up to 10^4 times less costly. Similarly the leading order costs of each smoothing parameter update can be compared. The Wood, Goude, and Shaw (2015) method requires approximately 40 times the floating point operations per smoothing parameter update, due to $O(p^3)$ costs per smoothing parameter, coupled with a symmetric eigen decomposition and several QR steps. Hence, all three components of the new method are required to achieve the observed efficiency gains.

For discretization we chose to generalize the approach of Lang et al. (2014), rather than attempt to use the grid-based

approach of Currie, Durban, and Eilers (2006). This is largely as a result of the very irregular nature of our “grids”: for example, the approach here avoids having to compute anything that will then be given zero weight as a result of data being missing at a grid node. However, our smoothing parameter selection method should be directly applicable to models fit using the Currie, Durban, and Eilers (2006) approach (unlike, e.g., the approach of Wood (2011)).

The Black smoke model presented here is the first successful attempt to model these data on a daily basis over several decades, and offers a basis for estimating daily exposures for use in retrospective cohort studies, for example. While a major advance, we do not believe that this model is definitive. For example, the only meteorological variables available to us on a daily basis were temperature, and the fact that we are forced to use monthly rainfall data offers an obvious area for improvement. The model as it stands shows some artifacts in coastal areas that we are working to improve. Another obvious deficiency is the lack of any pollution source data. One might expect substantial improvements if fine scale data on coal and diesel use were available as predictors.

The method is implemented in the `bam` function of R package `mgcv` from version 1.8-9, and is invoked via `bam` arguments `discrete` and `nthreads`. The black smoke data are available from the first author’s web page (<http://www.maths.bris.ac.uk/~sw15190/>).

A. Methods for Discretized Covariates

This section describes the algorithms required to compute efficiently with *marginally* gridded covariates in detail. The idea is that we have a model matrix $\mathbf{X} = (\mathbf{X}_0 : \mathbf{X}_1 : \dots)$. Each \mathbf{X}_j represents either a single smooth, or a tensor product smooth (e.g. Wood 2006). In the case of a single smooth

$$\mathbf{X}_j(i, l) = \bar{\mathbf{X}}_j(k_j(i), l), \tag{9}$$

where $\bar{\mathbf{X}}_j$ is an $m_j \times p_j$ matrix evaluating the smooth at the corresponding gridded values. For a tensor product

$$\mathbf{X}_j = \mathbf{M}_0^j \odot \mathbf{M}_1^j \odot \dots \odot \mathbf{M}_{d_j-1}^j \mathbf{Q}^j,$$

where \mathbf{M}_k^j are marginal model matrices and \mathbf{Q}^j is a constraint matrix, usually imposing a sum to zero constraint over a

representative subset of the data. \odot denotes the Kronecker product (\otimes) applied row-wise (i.e., one row at a time). In this case the marginal model matrices are stored in compact form:

$$M_l^j(i, m) = \bar{M}_l^j(k_l^j(i), m).$$

The following algorithms are most efficient if tensor product terms are always arranged so that the marginal model matrix with the most columns is last, but this can be achieved by automatic rearrangement.

Note that in principle covariates could be discretized *jointly* onto a multidimensional grid, so that we store the unique *combinations* of covariates, rather than storing the unique covariate values independently. With the joint scheme the cross product $\mathbf{X}^T \mathbf{W} \mathbf{X}$ is easy to compute. If $\bar{\mathbf{X}}$ and $\bar{\mathbf{W}}$ contain the unique model matrix rows and corresponding unique weights, respectively, while $\bar{\mathbf{N}}$ is the diagonal matrix containing the number of occurrences of each row of $\bar{\mathbf{X}}$ in \mathbf{X} then $\mathbf{X}^T \mathbf{W} \mathbf{X} = \bar{\mathbf{X}}^T \bar{\mathbf{N}} \bar{\mathbf{W}} \bar{\mathbf{X}}$. The problem is that the number of unique combinations of covariates, and hence number of rows of $\bar{\mathbf{X}}$ can be very large, unless very coarse discretisation is used. Hence the requirement for the methods of this appendix.

A variant of the scheme is required when the model contains terms of the form $\sum_k f_j(z_{ik}) L_{ik} = \Sigma_j \{f_j(\text{vec}(\mathbf{z})) \odot \text{vec}(\mathbf{L})\} = \Sigma_j \{\bar{\mathbf{X}} \odot \text{vec}(\mathbf{L})\} \beta$, where

$$\Sigma_j = \begin{pmatrix} 1 & 0 & \dots & 0 & 1 & 0 & \dots \\ 0 & 1 & 0 & \dots & 0 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}.$$

If \mathbf{z} is $n \times m$, then Σ_j is $n \times nm$, and the index vectors must be of length nm , which is also the number of rows in $\bar{\mathbf{X}}$ (the model matrix for $f_j(\text{vec}(\mathbf{z}))$). The regular case corresponds to $\Sigma_j = \mathbf{I}$. Note that an \mathbf{L} term can be treated as an extra single column tensor product marginal. A1, A2, A5 and A6, below, simply require Σ_j to be applied as the final step, while A3 and A4 require the extra work detailed.

The matrix products required in fitting require the following basic algorithms.

A1 Extraction of a single column of a single term \mathbf{X}_j uses (9) at $O(n)$ cost.

A2 Extraction of a single column of a tensor product term \mathbf{X}_j . Let p_k denote the number of columns of \mathbf{M}_k^j , and $q_k = \prod_{i=k+1}^{d_j-1} p_i$, with $q_{d_j-1} = 1$. Then

$$X_j(i, l) = \prod_{m=0}^{d_j-1} \bar{M}_m^j(k_m(i), j_m)$$

where the j_m are defined by the following recursion. $q_{-1} = \prod_{i=0}^{d_j-1} p_i$, $j'_{-1} = j$, then iterate from $i = 0$: $q_i = q_{i-1}/p_i$, $j_i = \lfloor j'_{i-1}/q_i \rfloor$, $j'_i = j'_{i-1} \bmod q_i$. The cost of the whole column is $O(nd_j)$.

A3 Single term $\mathbf{X}_j^T \mathbf{y}$.

$$\mathbf{X}_j^T \mathbf{y} = \bar{\mathbf{X}}_j^T \bar{\mathbf{y}} \text{ where } \bar{y}_l = \sum_{k_j(i)=l} y_i,$$

which has cost $O(n) + O(m_j p_j)$. If $\Sigma_j \neq \mathbf{I}$ then

$$\bar{y}_l = \sum_{k_j(i)=l} (\Sigma_j^T \mathbf{y})_i,$$

where the latter is readily computable without explicit formation of $\Sigma_j^T \mathbf{y}$.

A4 Tensor product term $\mathbf{v} = \mathbf{X}_j^T \mathbf{y}$ at cost $O(n\bar{p}) + O(m_{d_j-1} p_j)$.

Let p_k be as in A2 and $\bar{p} = \prod_{i=0}^{d_j-2} p_i$. Then repeat the following for $l = 0 \dots \bar{p} - 1$.

1. Extract column l of $\mathbf{A} = \mathbf{M}_0^j \odot \mathbf{M}_1^j \odot \dots \odot \mathbf{M}_{d_j-2}^j \odot \mathbf{y}$ using A2 (without Σ_j).

2. Form $v(l p_{d_j} : (l p_{d_j} + p_{d_j} - 1)) = \mathbf{M}_{d_j-1}^T \mathbf{A}(:, l)$ using A3 (with Σ_j , if present).

3. Set $\mathbf{v} \leftarrow \mathbf{Q}_j^T \mathbf{v}$

A5 $\mathbf{X}_j \beta$ for single term. $(\mathbf{X}_j \beta)(i) = (\bar{\mathbf{X}}_j \beta)(k_j(i))$. Cost $O(m_j p_j) + O(n)$.

A6 $\mathbf{f} = \mathbf{X}_j \beta$ for tensor product term. Notation as A4. Let \mathbf{B} be $p_{d_j} \times \bar{p}$ such that $\text{vec}(\mathbf{B}) = \mathbf{Q}_j \beta$. Let $\mathbf{C} = \bar{\mathbf{M}}_{d_j-1} \mathbf{B}$, and $\mathbf{A} = \mathbf{M}_0^j \odot \mathbf{M}_1^j \odot \dots \odot \mathbf{M}_{d_j-2}^j$. Then repeat the following for $l = 0 \dots \bar{p} - 1$.

1. Extract column j of \mathbf{A} using A2 (without Σ_j).

2. For $i = 0 \dots n - 1$ $f(i) \leftarrow f(i) + C(k_{d_j-1}(i), j) A(i, j)$.

The formation of $\mathbf{X}_j^T \mathbf{W} \mathbf{X}_k$ then uses these basic algorithms as follows. First, if the final marginal of k has more columns than the final marginal of j then form $\mathbf{X}_k^T \mathbf{W} \mathbf{X}_j$ and transpose (a single smooth is its own marginal, of course). This maximizes efficiency, since the factor saved relative to direct formation is the dimension of the largest final marginal. The algorithm is then as follows.

1. For $i = 0, \dots, p_k - 1 \dots$

(a) Extract $X_k(:, i)$ using A1 or A2 as appropriate.

(b) Form $\mathbf{W} X_k(:, i)$.

(c) Form $\mathbf{X}_j^T \mathbf{W} X_k(:, i)$ using A3 or A4 as appropriate.

2. If the \mathbf{X}_k is a tensor product then we may need to update

$$\mathbf{X}_j^T \mathbf{W} \mathbf{X}_k \leftarrow \mathbf{X}_j^T \mathbf{W} \mathbf{X}_k \mathbf{Q}_k$$

\mathbf{Q} is usually implemented as a single Householder matrix, so that multiplication by \mathbf{Q} is an efficient rank one update. Step one is easily parallelized using openMP (OpenMP Architecture Review Board 2008). Finally note that it is easy to substitute \mathbf{W} with a banded matrix, such as the tri-diagonal precision matrix implied by an AR1 residual error model.

Prediction from the fitted model can use A5 and A6, but the computation of prediction variances also requires that we compute $\text{diag}(\mathbf{X} \mathbf{V} \mathbf{X}^T)$ where \mathbf{V} is a covariance matrix. This computation can also be built from A5 and A6 using the fact that

$$\text{diag}(\mathbf{X} \mathbf{V} \mathbf{X}^T) = \sum_i \mathbf{X} \mathbf{V}(:, i) \odot \mathbf{X}(:, i).$$

Supplementary Materials

The supplementary material contains further information on the data, model checking and preferential sampling. In addition, gigam2-AppBC.pdf contains appendices B and C of the paper, on convergence properties and parallel computing issues. Finally files tps.mp4 and duchon.mp4 contain movies showing the fitted model predictions for the whole UK.

Acknowledgments

We are grateful to the editor and anonymous referees for comments which substantially improved the article, to Electricité de France for funding background investigation of parallel computing methods, and to Yannig Goude for discussion of GAM scalable computing issues.

Funding

This work was funded by EPSRC grant EP/K005251/1 and a University of Bath studentship to ZL.

References

- Augustin, N. H., Musio, M., von Wilpert, K., Kublin, E. Wood, S. N., and Schumacher, M. (2009), "Modeling Spatiotemporal Forest Health Monitoring Data," *Journal of the American Statistical Association*, 104, 899–911. [7]
- Breslow, N. E., and Clayton, D. G. (1993), "Approximate Inference in Generalized Linear Mixed Models," *Journal of the American Statistical Association*, 88, 9–25. [3]
- Ciocco, A., and Thompson, D. (1961), "A Follow-Up of Donora Ten Years After: Methodology and Findings," *American journal of Public Health Nations Health*, 51, 155–164. [6]
- Claeskens, G., Krivobokova, T. and Opsomer, J. D. (2009), "Asymptotic Properties of Penalized Spline Estimators," *Biometrika*, 96, 529–544. [3]
- Clifton, M. (1964), "Air Pollution," *Journal of the Royal Society of Medicine*, 57, 615–618. [6]
- Craven, P., and Wahba, G. (1979), "Smoothing Noisy Data With Spline Functions," *Numerische Mathematik*, 31, 377–403. [3]
- Currie, I. D., Durban, M. and Eilers, P. H. (2006), "Generalized Linear Array Models With Applications to Multidimensional Smoothing," *Journal of the Royal Statistical Society, Series B*, 68, 259–280. [9]
- Diggle, P. J., Menezes, R., and Su, T.-I. (2010), "Geostatistical Inference Under Preferential Sampling," *Journal of the Royal Statistical Society, Series C*, 59, 191–232. [8]
- Duchon, J. (1977), "Splines Minimizing Rotation-Invariant Semi-Norms in Solobev Spaces," in *Construction Theory of Functions of Several Variables*, eds. W. Schemp and K. Zeller, Berlin: Springer, pp. 85–100. [3,7]
- Eilers, P. H., and Marx, B. D. (2003), "Multivariate Calibration With Temperature Interaction Using Two-Dimensional Penalized Signal Regression," *Chemometrics and Intelligent Laboratory Systems*, 66(2), 159–174. [3]
- Eilers, P. H. C., and Marx, B. D. (1996), "Flexible Smoothing With B-Splines and Penalties," *Statistical Science*, 11(2), 89–121. [3]
- Fahrmeir, L., and Lang, S. (2001), "Bayesian Inference for Generalized Additive Mixed Models Based on Markov Random Field Priors," *Applied Statistics*, 50, 201–220. [3]
- Firket, J. (1936), "Fog Along the Meuse Valley," *Transactions of the Faraday Society*, 32, 1191–1194. [6]
- Green, P. J., and Silverman, B. W. (1994), *Nonparametric Regression and Generalized Linear Models*, London: Chapman & Hall. [3]
- Gu, C. (1992), "Cross-Validating Non-Gaussian Data," *Journal of Computational and Graphical Statistics*, 1, 169–179. [3]
- Gu, C., and Kim, Y. J. (2002), "Penalized Likelihood Regression: General Approximation and Efficient Approximation," *Canadian Journal of Statistics*, 34(4), 619–628. [3]
- Hall, P., and Opsomer, J. D. (2005), "Theory For Penalised Spline Regression," *Biometrika*, 92(1), 105–118. [3]
- Handcock, M. S., Meier, K., and Nychka, D. (1994), "Comment," *Journal of the American Statistical Association*, 89(426), 401–403. [3,7]
- Hastie, T., and Tibshirani, R. (1986), "Generalized Additive Models" (with discussion), *Statistical Science*, 1, 297–318. [2]
- Hastie, T., and Tibshirani, R. (1990), *Generalized Additive Models*, London: Chapman & Hall. [2,4]
- Helwig, N. E., and Ma, P. (2016), "Smoothing Spline Anova for Super-Large Samples: Scalable Computation Via Rounding Parameters," *arXiv preprint arXiv:1602.05208*. [6]
- Kammann, E. E., and Wand, M. P. (2003), "Geoadditive Models," *Applied Statistics*, 52(1), 1–18. [3,7]
- Kauermann, G., Krivobokova, T., and Fahrmeir, L. (2009), "Some Asymptotic Results on Generalized Penalized Spline Smoothing," *Journal of the Royal Statistical Society, Series B*, 71, 487–503. [3]
- Kimeldorf, G. S., and Wahba, G. (1970), "A Correspondence Between Bayesian Estimation on Stochastic Processes and Smoothing by Splines," *The Annals of Mathematical Statistics*, 41, 495–502. [3]
- Lang, S., Umlauf, N., Wechselberger, P., Harttgen, K., and Kneib, T. (2014), "Multilevel Structured Additive Regression," *Statistics and Computing*, 24, 223–238. [6,9]
- Li, Y., and Ruppert, D. (2008), "On the Asymptotics of Penalized Splines," *Biometrika*, 95, 415–436. [3]
- Loader, A. (2002), *Instruction manual: UK Smoke and Sulphur Dioxide Network*, Culham Science Centre: Netcen, AEA Technology. [7]
- Lucas, C. (2004), "Lapack-Style Codes for Level 2 and 3 Pivoted Cholesky Factorizations," *LAPACK Working*. [4,5]
- Marra, G., and Wood, S. N. (2012), "Coverage Properties of Confidence Intervals for Generalized Additive Model Components," *Scandinavian Journal of Statistics*, 39, 53–74. [4]
- Marx, B. D., and Eilers, P. H. (1998), "Direct Generalized Additive Modeling With Penalized Likelihood," *Computational Statistics and Data Analysis*, 28, 193–209. [3]
- McCullagh, P., and Nelder, J. A. (1989), *Generalized Linear Models* (2nd ed.), London: Chapman & Hall. [4]
- Miller, D. L., and Wood, S. N. (2014), "Finite Area Smoothing With Generalized Distance Splines," *Environmental and Ecological Statistics*, 1–17. [7]
- Ministry of Health, (1954), *Mortality and Morbidity During the London Fog of December 1952*, London: HMSO. [6]
- Nocedal, J., and Wright, S. (2006), *Numerical Optimization* (2nd ed.), New York: Springer verlag. [5]
- Nychka, D. (1988), "Bayesian Confidence Intervals for Smoothing Splines," *Journal of the American Statistical Association*, 83, 1134–1143. [4]
- of Terrain-50, C. (2015), *OS Terrain 50: User Guide and Technical Specification*, Adanac Drive, Southampton, SO16 0AS: Ordnance Survey. [7]
- OpenMP Architecture Review Board (2008, May). OpenMP application program interface version 3.0. [4,10]
- Parker, R., and Rice, J. (1985), Discussion of "Some Aspects of the Spline Smoothing Approach to Non-Parametric Regression Curve Fitting," by Silverman, *Journal of the Royal Statistical Society, Series B*, 47, 40–42. [3]
- Perry, M., and Hollis, D. (2005a), "The Development of a New Set of Long-Term Climate Averages for the UK," *International Journal of Climatology*, 25, 1023–1039. [7]
- Perry, M., and Hollis, D. (2005b), "The Generation of Monthly Gridded Datasets for a Range of Climatic Variables Over the UK," *International Journal of Climatology*, 25, 1041–1054. [7]
- Quintana-Ortí, G., Sun, X., and Bischof, C. H. (1998), "A BLAS-3 Version of the QR Factorization With Column Pivoting," *SIAM Journal on Scientific Computing*, 19, 1486–1494. [4]
- Reiss, P. T., and Ogden, T. R. (2009), "Smoothing Parameter Selection for a Class of Semiparametric Linear Models," *Journal of the Royal Statistical Society, Series B*, 71, 505–523. [3]
- Rue, H., S. Martino, and Chopin, N., (2009), "Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations," *Journal of the Royal Statistical Society, Series B*, 71(2), 319–392. [2]
- Ruppert, D., Wand, M. P., and Carroll, R. J. (2003), *Semiparametric Regression*, Cambridge, MA: Cambridge University Press. [3]
- Shaddick, G., and Zidek, J. V. (2014), "A Case Study in Preferential Sampling: Long Term Monitoring of Air Pollution in the UK," *Spatial Statistics*, 9, 51–65. [1,2,8]
- Silverman, B. W. (1985), "Some Aspects of the Spline Smoothing Approach to Non-Parametric Regression Curve Fitting," *Journal of the Royal Statistical Society Series B*, 47, 1–53. [3,4]
- Wahba, G. (1980), "Spline Bases, Regularization, and Generalized Cross Validation for Solving Approximation Problems With Large Quantities of Noisy Data," in *Approximation Theory III*, eds. E. Cheney, London: Academic Press. [3]

- Wahba, G. (1981), “Spline Interpolation and Smoothing on the Sphere,” *SIAM Journal on Scientific and Statistical Computing*, 2, 5–16. [3]
- Wahba, G. (1983), “Bayesian Confidence Intervals for the Cross Validated Smoothing Spline,” *Journal of the Royal Statistical Society, Series B*, 45, 133–150. [3,4]
- Wahba, G. (1990), *Spline Models for Observational Data*, Philadelphia, PA: SIAM. [2]
- Wang, X., J. Shen, and Ruppert, D., (2011), “On the Asymptotics of Penalized Spline Smoothing,” *Electronic Journal of Statistics*, 5, 1–17. [3]
- Wood, S. N. (2003), “Thin Plate Regression Splines,” *Journal of the Royal Statistical Society, Series B*, 65, 95–114. [3]
- Wood, S. N. (2006), “Low-Rank Scale-Invariant Tensor Product Smoother for Generalized Additive Mixed Models,” *Biometrics*, 62, 1025–1036. [3,7,9]
- Wood, S. N. (2011), “Fast Stable Restricted Maximum Likelihood and Marginal Likelihood Estimation of Semiparametric Generalized Linear Models,” *Journal of the Royal Statistical Society, Series B*, 73(1), 3–36. [2,3,4,9]
- Wood, S. N. (2013), “On p -Values for Smooth Components of an Extended Generalized Additive Model,” *Biometrika*, 100, 221–228. [4]
- Wood, S. N. (2015), *Core Statistics*, Cambridge, MA: Cambridge University Press. [4]
- Wood, S. N., Goude, Y., and Shaw, S. (2015), “Generalized Additive Models for Large Data Sets,” *Journal of the Royal Statistical Society, Series C*, 64, 139–155. [1,2,4,9]
- Wood, S. N., Scheipl, F., and Faraway, J. J. (2013), “Straightforward Intermediate Rank Tensor Product Smoothing in Mixed Models,” *Statistics and Computing*, 23, 341–360. [3]