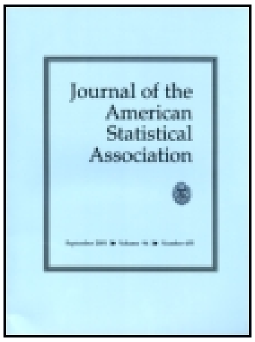

Journal of the American Statistical Association

Taylor & Francis
Taylor & Francis Group

Journal of the American Statistical Association

ISSN: 0162-1459 (Print) 1537-274X (Online) Journal homepage: http://www.tandfonline.com/loi/uasa20

Generalized additive models for gigadata: modelling the UK black smoke network daily data

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

Generalized additive models for gigadata: modelling the UK black smoke network daily data.

Simon N. Wood[0], Zheyuan Li[1], Gavin Shaddick[1] and Nicole H. Augustin[1]
[0] School of Mathematics, University of Bristol, Bristol, U.K.
[1]Mathematical Sciences, University of Bath, Bath, U.K.
simon.wood@bristol.ac.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 UK 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. Modelling 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 data set size, previous work has focussed on modelling time or space averages 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.

1 Introduction

This paper 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 et al., 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 cross-products. 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 $\mu\text{g m}^{-3}$ (microgrammes per cubic metre) of BS particulates (largely from coal and Diesel combustion) from 1961 to 2005. The UK 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 1a 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 9451232 daily measurements from 2862 monitoring sites.

Because of the data volume, previous attempts to model spatio-temporal 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 $213\mu\text{g m}^{-3}$.

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. Firstly, 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. Secondly, the reduced number of stations later in the data make spatial predictions difficult without a model that is able to share information across years. Thirdly there are strong covariate effects.

We will end up using a model structure

$$\begin{aligned}\log(\texttt{bs}_i) = {} & f_1(\texttt{y}_i) + f_2(\texttt{doy}_i) + f_3(\texttt{dow}_i) + f_4(\texttt{y}_i, \texttt{doy}_i) + f_5(\texttt{y}_i, \texttt{dow}_i) + f_6(\texttt{doy}_i, \texttt{dow}_i) \\ & + f_7(\texttt{n}_i, \texttt{e}_i) + f_8(\texttt{n}_i, \texttt{e}_i, \texttt{y}_i) + f_9(\texttt{n}_i, \texttt{e}_i, \texttt{doy}_i) + f_{10}(\texttt{n}_i, \texttt{e}_i, \texttt{dow}_i) \\ & + f_{11}(\texttt{h}_i) + f_{12}(\texttt{T}^0_i, \texttt{T}^1_i) + f_{13}(\overline{\texttt{T1}}_i, \overline{\texttt{T2}}_i) + f_{14}(\texttt{r}_i) + \alpha_{k(i)} + b_{\text{id}(i)} + e_i \qquad (1)\end{aligned}$$

where $\texttt{y}$, $\texttt{doy}$ and $\texttt{dow}$ denote, year, day of year and day of week; $\texttt{n}$ and $\texttt{e}$ denote location as kilometres north and east; $\texttt{h}$ and $\texttt{r}$ are height (elevation of station) and cube root transformed rainfall (unfortunately only available as monthly average); $\texttt{T}^0$ and $\texttt{T}^1$ are daily minimum and maximum temperature, while $\overline{\texttt{T1}}$ and $\overline{\texttt{T2}}$ 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 centre), X (mixed) or M (missing)); $b_{\text{id}(i)}$ is a random effect for the id^{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 et al. (2015) (available in R package `mgcv`) or to follow Shaddick and Zidek (2014) in using the method of Rue et al. (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 et al. (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 paper for REML smoothing parameter selection. Using just the published method would have required approximately 5 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 modelling 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

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

$\mu_i = E(y_i)$, EF denotes an exponential family distribution with known or unknown scale parameter ϕ, g is a known smooth monotonic link function, $A(i,:)$ 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 Sobolov) 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 recognised 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 et al. (2009), Claeskens et al. (2009) and Wang et al. (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 et al. (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 et al., 1994) are useful in some spatial applications. In all cases if $f_j = [f_j(x_{j1}), f_j(x_{j2}), \ldots]^T$ we can write $f_j = X_j\boldsymbol{\beta}_j$ where 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 $\boldsymbol{\beta}_j^T \boldsymbol{\mathcal{S}}_j \boldsymbol{\beta}_j$, where $\boldsymbol{\mathcal{S}}_j$ 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 et al. (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 X_j and $\boldsymbol{\mathcal{S}}_j$ (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 $X = (A, X_1, X_2, \ldots)$ with corresponding combined parameter vector $\boldsymbol{\beta}$. Given some smoothing parameters $\boldsymbol{\lambda}$ a combined smoothing penalty could then be written as $\sum_j \lambda_j \boldsymbol{\beta}_j^T \boldsymbol{\mathcal{S}}_j \boldsymbol{\beta}_j = \sum_j \lambda_j \boldsymbol{\beta}^T S_j \boldsymbol{\beta} = \boldsymbol{\beta}^T S_\lambda \boldsymbol{\beta}$, where S_j is simply a zero padded version of $\boldsymbol{\mathcal{S}}_j$ and $S_\lambda = \sum_j \lambda_j S_j$. Hence we have an overparameterized GLM structure, $g(\boldsymbol{\mu}) = X\boldsymbol{\beta}$. Given smoothing parameters it is estimated via

$$\hat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmax}} \;\; l(\boldsymbol{\beta}) - \boldsymbol{\beta}^T S_\lambda \boldsymbol{\beta}/2. \tag{3}$$

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

$$\boldsymbol{\beta} \sim N(0, S_\lambda^-)$$

where S_λ^- is a Moore-Penrose pseudoinverse (S_λ being rank deficient because the penalties leave some space of functions un-penalized, 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 et al., 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) show that a (restricted) marginal likelihood approach (e.g. Wood, 2011) offers practical reliability advantages, in being less prone to multiple local optima and consequent under-smoothing.

3 The fitting iteration

The purpose of this paper is to allow the rich existing modelling 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 re-weighted 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 $W = \text{diag}(w_i)$ 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

$$z = X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\beta} \sim N(0, S_\lambda^-), E(\boldsymbol{\epsilon}) = 0 \text{ and } E(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) = \phi W^{-1}.$$

The key idea of performance iteration/PQL is to estimate $\boldsymbol{\lambda}$ and ϕ 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 $\boldsymbol{\epsilon} \sim N(0, W^{-1}\phi)$. If $\hat{\boldsymbol{\beta}}_\lambda = \text{argmin}_\beta \|z - X\boldsymbol{\beta}\|_w^2/\phi + \boldsymbol{\beta}^T S_\lambda \boldsymbol{\beta}$, where $\|x\|_w^2 = x^T W x$ and M is the dimension of the null space of S_λ, then the twice negative log REML (e.g. Wood, 2011) is

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

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

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

where $\tau = \text{tr}\{(X^T W X/\hat{\phi} + S_\lambda)^{-1} X^T W X/\hat{\phi}\}$ 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 z (see e.g., Wahba, 1983; McCullagh and Nelder, 1989; Hastie and Tibshirani, 1990).

Now let us eliminate the false assumption of normality of z, replacing it with central limit theorem justification. Consider the QR decomposition $\sqrt{W}X = \boldsymbol{\mathcal{Q}}\boldsymbol{\mathcal{R}}$, where $\boldsymbol{\mathcal{Q}}$ has orthogonal columns and $\mathcal{R}$ 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 $f = Q^T\sqrt{W}z$, $r = \|z\|_w^2 - \|f\|^2$. In that case $X^TWX = \mathcal{R}^T\mathcal{R}$, $\|z - X\hat{\boldsymbol{\beta}}_\lambda\|_w^2 = \|f - \mathcal{R}\boldsymbol{\beta}\|^2 + r$, and we have the alternative working model

$$f = \mathcal{R}\boldsymbol{\beta} + e,\ \boldsymbol{\beta} \sim N(0, S_\lambda^-) \text{ and } e \sim N(0, I\phi), \tag{6}$$

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

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

For a given ϕ, $\mathcal{V}$ and $\mathcal{V}_r$ differ only by an additive constant, and therefore result in identical inference about $\boldsymbol{\lambda}$ 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 $\boldsymbol{\lambda}$. This justifies use of (4) for $\boldsymbol{\lambda}, \phi$ estimation.

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

$$\boldsymbol{\beta} \sim N(\hat{\boldsymbol{\beta}}, (X^TWX/\phi + S_\lambda)^{-1}), \tag{7}$$

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 X whole.

2. The log determinant terms in $\mathcal{V}$ are potentially numerically unstable. Because having some $\lambda_j \to \infty$ is legitimate in GAM estimation, 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 et al. (2015) addressed 1 by iteratively updating the QR factorization of X, and then applying the method of Wood (2011) to (6). This approach ignored 3, requires pivoted QR decomposition and addressed 2 by stabilizing re-parameterizations 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 et al. (1998), only has around half the floating point operations as matrix-matrix computations. In consequence the Wood et al. (2015) was computationally impractical for the black smoke model. See appendix C for a discussion of the issues around multi-core computing.

Our proposal here addresses 3 by taking a single Newton step to update $\boldsymbol{\rho} = \log(\boldsymbol{\lambda})$ at each cycle of the iteration (rather than fully optimizing $\mathcal{V}$ at each cycle). We propose to avoid the stabilizing re-parameterization 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 in order 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 $\boldsymbol{\Delta}^T \nabla \mathcal{V}(\boldsymbol{\rho} + \boldsymbol{\Delta}) \leq 0$, i.e., that $\mathcal{V}$ is non-increasing in the direction of $\boldsymbol{\Delta}$ at the end of the Newton step (see e.g. Wood, 2015, §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 $X^T WX$, 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 re-parameterization described in section 4.3.
- Initialize $\boldsymbol{\rho}_0$, $\boldsymbol{\Delta}_0 = 0$, $\hat{\mu}_i = y_i + \delta_i$ 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 $X^T WX$, $f = X^T Wz$ and penalized deviance, D. $z_i = g'(\hat{\mu}_i)(y_i - \hat{\mu}_i) + \hat{\eta}_i$ and 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 $D^*/\phi + \boldsymbol{\beta}^{*T} S_\rho \boldsymbol{\beta}^* < D/\phi + \hat{\boldsymbol{\beta}}^T S_\rho \hat{\boldsymbol{\beta}}$ 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 $X^T W X$, 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. $\boldsymbol{\Delta}_0 \leftarrow \boldsymbol{\Delta}$, $\boldsymbol{\rho}_0 \leftarrow \boldsymbol{\rho}$, $D^* \leftarrow D$. Form $\hat{\boldsymbol{\eta}} = X\hat{\boldsymbol{\beta}}$ and $\hat{\mu}_i = g^{-1}(\hat{\eta}_i)$.

Note that step 1 does not require the explicit formation of the whole matrix X. Step 3 reduces the $\boldsymbol{\beta}$ step taken if the Newton step was too long, in that it increased the penalized deviance at the $\boldsymbol{\rho}$ value at which it was computed. Step 5 reduces the $\boldsymbol{\rho}$ 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 $\boldsymbol{\rho}$.

Step 6 consists of estimating the $\hat{\boldsymbol{\beta}}_\lambda$ implied by the proposed $\boldsymbol{\rho}$ and the current W and 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, $\boldsymbol{\Delta}$, to improve (4). We have that $\hat{\boldsymbol{\beta}}_\lambda$ is the solution of $(X^T W X + \phi S_\lambda)\boldsymbol{\beta}_\lambda = X^T W z$. The actual computation proceeds by taking a Cholesky decomposition $R^T W R = X^T X/\phi + S_\lambda$ using a parallel version of Lucas (2004). This is usually done with pivoting, in which case the rank of R is then estimated and unidentifiable parameters set to zero and dropped from subsequent computations. We then compute $\hat{\boldsymbol{\beta}}_\lambda = R^{-1}R^{-T}X^T W z/\phi$ (by backwards 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 $\boldsymbol{\Delta} = -(\mathrm{d}^2\mathcal{V}/\mathrm{d}\boldsymbol{\rho}\mathrm{d}\boldsymbol{\rho}^T)^{-1}\mathrm{d}\mathcal{V}/\mathrm{d}\boldsymbol{\rho}$, where $\mathrm{d}^2\mathcal{V}/\mathrm{d}\boldsymbol{\rho}\mathrm{d}\boldsymbol{\rho}^T$ will have been perturbed if necessary to ensure definiteness (see Nocedal and Wright, 2006). Recalling that $\mathrm{d}(\|\mathrm{z} - X\boldsymbol{\beta}\|^2_w/\phi + \boldsymbol{\beta}^T S_\lambda\boldsymbol{\beta}$ 0, we have

$$\frac{\mathrm{d}\mathcal{V}}{\mathrm{d}\rho_j} = \lambda_j\hat{\boldsymbol{\beta}}^T_\lambda S_j\hat{\boldsymbol{\beta}}_\lambda + \frac{\mathrm{d}\log|X^TWX/\phi + S_\lambda|}{\mathrm{d}\rho_j} - \frac{\mathrm{d}\log|S_\lambda|_+}{\mathrm{d}\rho_j} \tag{8}$$

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

$$\frac{\mathrm{d}^2\mathcal{V}}{\mathrm{d}\rho_j\mathrm{d}\rho_k} = 2\frac{\mathrm{d}\hat{\boldsymbol{\beta}}^T_\lambda}{\mathrm{d}\rho_k}(X^TWX/\phi + S_\lambda)\frac{\mathrm{d}\hat{\boldsymbol{\beta}}_\lambda}{\mathrm{d}\rho_j} + 2\lambda_j\boldsymbol{\beta}^T_\lambda S_j\frac{\mathrm{d}\hat{\boldsymbol{\beta}}_\lambda}{\mathrm{d}\rho_k} + 2\lambda_k\boldsymbol{\beta}^T_\lambda S_k\frac{\mathrm{d}\hat{\boldsymbol{\beta}}_\lambda}{\mathrm{d}\rho_j} + \delta^j_k\lambda_j\hat{\boldsymbol{\beta}}^T_\lambda S_j\hat{\boldsymbol{\beta}}_\lambda + \frac{\mathrm{d}^2\log|X^TWX/\phi + S_\lambda|}{\mathrm{d}\rho_j\mathrm{d}\rho_k} - \frac{\mathrm{d}^2\log|S|_+}{\mathrm{d}\rho_j\mathrm{d}\rho_k}.$$

Implicit differentiation implies that

$$\frac{\mathrm{d}\hat{\boldsymbol{\beta}}_\lambda}{\mathrm{d}\rho_j} = -\lambda_jR^{-1}R^{-T}S_j\hat{\boldsymbol{\beta}}_\lambda.$$

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

4.3 Computing the derivatives of $\log|S_\lambda|_+$

S_λ has block diagonal structure that can be exploited. For example, denoting zero blocks by '.',

$$S_\lambda = \begin{pmatrix} \lambda_1S_1 & . & . & . & . \\ . & \lambda_2S_2 & . & . & . \\ . & . & \sum_j\lambda_jS_j & . & . \\ . & . & . & . & . \\ . & . & . & . & . \end{pmatrix}.$$

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,

$$\log|S_\lambda|_+ = \text{rank}(S_1)\log(\lambda_1) + \log|S_1|_+ + \text{rank}(S_2)\log(\lambda_2) + \log|S_2|_+ + \log|\sum_j \lambda_j S_j|_+ + \cdots$$

For any ρ_k relating to a single parameter block we have

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

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 pre-fit re-parameterization is undertaken, according to the type of block.

1. Single parameter diagonal blocks. These can be re-parameterized so that all non-zero elements are one, and the rank pre-computed.

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 $\sum_j \lambda_j S_j$ has full rank in the new parameterization. Again a similarity transform is used. Typically the S_j are of smaller dimension in the reparameterization and consequently an extra zero block is introduced on the diagonal of S_j.

The generalized determinant of type 3 blocks becomes an ordinary determinant of $\sum_j \lambda_j S_j$ after re-parameterization. Hence its derivatives follow from the standard result

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

4.4 Computing the derivatives of $\log|X^TWX/\phi + S_\lambda|$

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

1. Form $P = R^{-1}$, and un-pivot the rows of **P**. Then form PP^T. These steps are $O(p^3)$, but can be parallelized.

2. Form the matrices containing the non-zero rows of $S_k PP^T$ ($\forall k$). This step is cheap for all but type 3 blocks.

3. Compute the required derivatives using

$$\frac{\mathrm{d}\log|X^T WX/\phi + S_\lambda|}{\mathrm{d}\rho_k} = \lambda_k \mathrm{tr}(S_k PP^T)$$

   and

$$\frac{\mathrm{d}^2\log|X^T WX/\phi + S_\lambda|}{\mathrm{d}\rho_k \mathrm{d}\rho_j} = \delta_k^j \lambda_k \mathrm{tr}(S_k PP^T) - \lambda_j \lambda_k \mathrm{tr}(S_k PP^T S_j PP^T)$$

Note that $PP^T = (X^T WX/\phi + S_\lambda)^{-1}$, the Bayesian covariance matrix (which needs unpivoting).

The trace computations in step 3 are very efficient, given the block structure of the S_k, if we employ the following tricks. In general $\mathrm{tr}(AB) = \sum_{kj} A_{kj} B_{jk}$. Now let A have non-zero rows only between k_1 and k_2, while B has non-zero rows only between j_1 and j_2.

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

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

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

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

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

where $\bar{X}_j$ has only m rows and k is an index vector. Storing $\bar{X}_j$ and k uses much less memory than storing X_j directly. This idea is introduced in Lang et al. (2014) to obtain efficient storage and computation for large datasets. However in that paper they employ smooths of one covariate and only require terms of the form $X_j^T W X_j$, but not $X_j^T W X_k$. 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 X and its products when the sub-matrices of 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 non-tensor 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 cross products 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 discretisation 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 twentieth century major pollution episodes occurred in London, notably in 1952 an episode of fog, in which levels of black smoke exceeded 4,500 μg 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 US (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 worlds first co-ordinated national air pollution monitoring network was established in the UK, 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 sulphur 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 UK 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. Modelling offers the possibility to alleviate these problems, at least partially, and our approach to the UK 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 UK (Perry and Hollis, 2005b,a) to use as covariates, alongside site elevation (of Terrain-50, 2015). Given the volume of data, our initial exploratory model development concentrated first on modelling 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 (`y`) 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 `y` 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 2D spatial smoothers of `n` and `e`, in order to avoid extrapolation artefacts 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 towards the constant function, which is a reasonable modelling 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 et al. (1994), as well as thin plate splines, but in both cases basic model checking revealed artefacts in model predictions towards the end of the data. The supplementary material includes an animation of predicted log black smoke, clearly illustrating such artefacts 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 auto-correlation function plots and semi-variogram plots to check for un-modelled 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 supplementary material.

To illustrate the importance of 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 et al., 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 spatio-temporal pollution fields were repeated each year, but with a long term decay matching the full data set. 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 supplementary material. So our simulated data comes from a ‘truth’ that maintains a degree of spatio-temporal 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 modelling period (that is 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 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 supplementary material includes an animation showing the evolution of the predicted spatial pollution field over time. Careful examination shows some artefacts in the fields, usually in coastal regions away from observation stations, but otherwise the results appear reasonable, predicting high pollution levels in the industrial centres 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 μg 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 centre location, for several years in the 1960s. These figures are obtained by simply counting up predicted exceedance days by 5km square.

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 UK 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 parallelised matrix computations and a pivoted Cholesky decompostion; 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 discretised 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 modelling, 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 minutes to over 7.5 hours 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 cross product 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 cross product is around a factor of 10^2 less floating point intensive using the new discrete methods relative to direct cross product formation, while the sub leading 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 et al. (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 decompositon and several QR steps. Hence all three components of the new method are required to achieve the observed efficiency gains.

For discretisation we chose to generalize the approach of Lang et al. (2014), rather than attempt to use the grid based approach of Currie et al. (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 et al. (2006) approach (unlike, for example, 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 artefacts 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/`).

Acknowledgements

This work was funded by EPSRC grant EP/K005251/1 and a University of Bath studentship to ZL. We are grateful to the editor and anonymous referees for comments which substantially improved the paper, to Electricité de France for funding background investigation of parallel computing methods, and to Yannig Goude for discussion of GAM scalable computing issues.

A Methods for discretised 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 $X = (X_0 : X_1 : \cdots)$. Each X_j represents either a single smooth, or a tensor product smooth (e.g. Wood, 2006). In the case of a single smooth

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

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

$$X_j = M_0^j \odot M_1^j \odot \cdot M_{d_j-1}^j Q^j.$$

where M_k^j are marginal model matrices and 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 re-arrangement.

Note that in principle covariates could be discretized *jointly* onto a multi-dimensional 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 X^TWX is easy to compute. If $\bar{X}$ and $\bar{W}$ contain the unique model matrix rows and corresponding unique weights, respectively, while $\bar{N}$ is the diagonal matrix containing the number of occurrences of each now of $\bar{X}$ in X then $X^TWX = \bar{X}^T\bar{N}\bar{W}\bar{X}$. The problem is that the number of unique combinations of covariates, and hence number of rows of $\bar{X}$ 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 $\sum_k f_j(z_{ik})L_{ik} = \mathbf{\Sigma}_j\{f_j(\text{vec}(z)) \odot \text{vec}(L)\} = \mathbf{\Sigma}_j\{\tilde{X} \odot \text{vec}(L)\}\boldsymbol{\beta}$, where

$$\mathbf{\Sigma}_j = \begin{pmatrix} 1 & 0 & . & . & . & 0 & 1 & 0 & . & . \\ 0 & 1 & 0 & . & . & . & 0 & 1 & 0 & . \\ . & . & . & . & . & . & . & . & . & . \end{pmatrix}.$$

If z is $n \times m$, then $\mathbf{\Sigma}_j$ is $n \times nm$, and the index vectors must be of length nm, which is also the number of rows in $\tilde{X}$ (the model matrix for $f_j(\text{vec}(z))$). The regular case corresponds to $\mathbf{\Sigma}_j = I$. Note that an L term can be treated as an extra single column tensor product marginal. A1, A2, A5 and A6, below, simply require $\mathbf{\Sigma}_j$ 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 X_j uses (9) at $O(n)$ cost.

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

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

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

A3 Single term $X_j^T y$.

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

which has cost $O(n) + O(m_j p_j)$. If $\mathbf{\Sigma}_j \ne I$ then

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

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

A4 Tensor product term $v = X_j^T y$ 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 $l = 0 \ldots \bar{p} - 1$.

1. Extract column l of $A = M_0^j \odot M_1^j \odot \cdot M_{d_j-2}^j \odot y$ using A2 (without $\boldsymbol{\Sigma}_j$).
2. Form $v(lp_{d_j} : (lp_{d_j} + p_{d_j} - 1)) = M_{d_j-1}^T A(:, l)$ using A3 (with $\boldsymbol{\Sigma}_j$, if present).
3. Set $v \leftarrow Q_j^T v$

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

A6 $f = X_j\boldsymbol{\beta}$ for tensor product term. Notation as A4. Let B be $p_{d_j} \times \bar{p}$ such that vec$(B) = Q_j\boldsymbol{\beta}$. Let $C = \bar{M}_{d_j-1}B$, and $A = M_0^j \odot M_1^j \odot \cdot M_{d_j-2}^j$. Then repeat the following for $l = 0 \ldots \bar{p} - 1$.

1. Extract column j of A using A2 (without $\boldsymbol{\Sigma}_j$).
2. For $i = 0 \ldots n - 1$ $f(i) \leftarrow f(i) + C(k_{d_j-1}(i), j)A(i, j)$.

The formation of $X_j^T W X_k$ then uses these basic algorithms as follows. Firstly if the final marginal of k has more columns than the final marginal of j then form $X_k^T W X_j$ 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 $i = 0, \ldots, p_k - 1 \ldots$
   (a) Extract $X_k(:, i)$ using A1 or A2 as appropriate.
   (b) Form $WX_k(:, i)$.
   (c) Form $X_j^T W X_k(:, i)$ using A3 or A4 as appropriate.

2. If the X_k is a tensor product then we may need to update

$$X_j^T W X_k \leftarrow X_j^T W X_k Q_k$$

Q is usually implemented as a single Householder matrix, so that multiplication by 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 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 $\text{diag}(XVX^T)$ where V is a covariance matrix. This computation can also be built from A5 and A6 using the fact that

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

B Convergence properties

Here we show that the section 4.1 iteration is guaranteed to converge when W is independent of $\boldsymbol{\beta}$ (e.g. for a Gaussian with identity link, gamma with log link or Poisson with square root link). The notation follows sections 3 and 4.

Theorem 1. *Let $V(\boldsymbol{\beta}, \boldsymbol{\lambda}) = D(\boldsymbol{\beta})/\phi + \boldsymbol{\beta}^T S_\lambda \boldsymbol{\beta} + \log|X^T WX/\phi + S_\lambda| - \log|S_\lambda|_+$ have a single (minimum) turning point and W be independent of $\boldsymbol{\beta}$. Then the iteration of section 4.1 converges to the minimum of V.*

Proof. Under the stated assumptions $\partial V/\partial \boldsymbol{\beta}$ coincides with the partial derivative of the penalized least squares objective minimized to compute $\hat{\boldsymbol{\beta}}_\lambda$. Since the step, $\Delta\boldsymbol{\beta}$ to $\hat{\boldsymbol{\beta}}_\lambda$ is hence minus the product of a positive definite matrix with $\partial V/\partial \boldsymbol{\beta}$, $\Delta\boldsymbol{\beta}$ is a descent direction for V (see e.g. Wood, 2015, §5.1.1). Since V only depends on $\boldsymbol{\beta}$ via $D(\boldsymbol{\beta})/\phi + \boldsymbol{\beta}^T S_\lambda \boldsymbol{\beta}$, then step 3 of the section 4.1 iteration is equivalent to testing that V has decreased. Furthermore it is readily checked that $\mathrm{d}\mathcal{V}/\mathrm{d}\boldsymbol{\rho}$

given in (8) coincides with $\partial V/\partial \boldsymbol{\rho}$. Since $\boldsymbol{\Delta}$ is again given by minus the product of a positive definite matrix and $\partial V/\partial \boldsymbol{\rho}$, it is therefore a descent direction for V. Given the coincidence of derivatives, step 7 of the section 4.1 iteration is equivalent to applying the same test to V, which will guarantee reduction of V if it has a single minimum (and will often do so when it is multimodal). Hence each step of the algorithm is guaranteed to reduce V until its minimum is reached. □

Obviously the strong assumption that W is independent of $\boldsymbol{\beta}$ is only sufficient, rather than necessary, for convergence. The proof suggests that convergence should occur whenever the derivatives of $D(\boldsymbol{\beta})/\phi + \boldsymbol{\beta}^T S_\lambda \boldsymbol{\beta}$ and V are 'close', but does not preclude convergence when this is not the case. Note that V is a version of the Laplace approximate marginal likelihood (e.g. Wood, 2011) but with the Hessian replaced by its expected value.

Practically the section 4.1 iteration typically takes around the same number of steps as a performance iteration in which the working REML score is fully optimized at each step: that is 5-20 for most cases but often around 10-40 for binary data. Since both the $\boldsymbol{\beta}$ updates and $\boldsymbol{\rho}$ updates are based on Newton methods, which tend towards quadratic convergence, there is rather little dependence of iteration number on tolerance, at least below the relative tolerances of around 10^{-6} used here (and little to be gained by loosening that tolerance). Small sample sizes *can* promote increased numbers of iterations (in both the old and new methods), as any identifiability/collinearity problems are then at their most acute, while the quadratic approximation on which each Newton step is based can also be poor on the scale of the Newton step at small sample sizes. There is little reason to expect sensitivity of the number of fitting steps to the basis dimensions used, and simulation experiments tend to back this up.

C Multi-core computing

Several computational details cannot be ignored in multi-core computing, if algorithms are to scale well with the number of cores used. These are briefly discussed here. We consider only the case of shared memory multi-core machines — that is the kind of architecture typical of a server or desktop workstation with 2 to 100 cores, rather than the distributed memory architecture of a supercomputer or GPU.

1. Current computers are memory bandwidth limited. While a floating point operation may take only 1 or 2 core cycles, retrieving the data on which to perform the operation may take 10 times as long. In consequence CPUs have a small amount of very fast access cache memory available as a buffer between main memory and the CPU cores. There are big performance gains to be had from structuring algorithms to re-use data already in cache. For numerical linear algebra this means constructing algorithms in a block oriented manner, where most floating point work consists of matrix-matrix multiplication. To see why this matter consider two operations with equal floating point operation count: Forming Ay where A is 1000×1000 and y is a vector, and forming BC where B and C are both 100×100 matrices. The former involves no re-use of the elements of A, whereas the latter involves multiple re-use of the elements of B and C.

2. Many modern CPUs use some form of 'Hyper-threading' where each physical core appears to the operating system as 2 cores. The idea is that two programme threads will often be performing tasks using different parts of the core, and hence can run at the same time on the same core. In floating point intensive computations all threads spend most of their time using the floating point unit, and there is no performance gain, but rather a loss, from hyper-threading. It is therefore usually better to disable it.

3. A multi-core CPU can run at a higher speed without overheating when only a few cores are in use, as opposed to all being in use. Most CPUs exploit this and run faster under low core loading. In consequence using 10 cores of a 10 core CPU can never be 10 times as fast as using one core, even with perfectly scalable code.

4. Under high energy efficiency settings it is possible for a thread doing relatively little work to fail to increase the speed of its core, relative to cores running more intensive threads. This can lead to the paradoxical situation in which the low work thread actually becomes the rate limiting one. Setting the CPU control policy to favour high performance will remove the problem (note that for modern CPUs the core speed is controlled in hardware and no longer directly by the operating system).

A further complication is the advent of non-uniform memory access (NUMA). When a machine has multiple CPUs (each with multiple cores), then memory is often arranged in blocks associated with a CPU. A CPU can still access all memory, but access is fastest from its associated block. In the work reported here we were unable to make good use of NUMA.

References

Augustin, N. H., M. Musio, K. von Wilpert, E. Kublin, S. N. Wood, and M. Schumacher (2009). Modeling spatiotemporal forest health monitoring data. *Journal of the American Statistical Association 104*(487), 899–911.

Breslow, N. E. and D. G. Clayton (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association 88*, 9–25.

Ciocco, A. and D. Thompson (1961). A follow-up of donora ten years after: methodology and findings. *Am J Public Health Nations Health 51*, 155–164.

Claeskens, G., T. Krivobokova, and J. D. Opsomer (2009). Asymptotic properties of penalized spline estimators. *Biometrika 96*(3), 529–544.

Clifton, M. (1964). Air pollution. *Journal of the Royal Society of Medicine 57*(7), 615–618.

Craven, P. and G. Wahba (1979). Smoothing noisy data with spline functions. *Numerische Mathematik 31*, 377–403.

Currie, I. D., M. Durban, and P. H. Eilers (2006). Generalized linear array models with applications to multidimensional smoothing. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) 68*(2), 259–280.

Diggle, P. J., R. Menezes, and T.-l. Su (2010). Geostatistical inference under preferential sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 59*(2), 191–232.

Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in Solobev spaces. In W. Schemp and K. Zeller (Eds.), *Construction Theory of Functions of Several Variables*, Berlin, pp. 85–100. Springer.

Eilers, P. H. and B. D. Marx (2003). Multivariate calibration with temperature interaction using two-dimensional penalized signal regression. *Chemometrics and intelligent laboratory systems 66*(2), 159–174.

Eilers, P. H. C. and B. D. Marx (1996). Flexible smoothing with B-splines and penalties. *Statistical Science 11*(2), 89–121.

Fahrmeir, L. and S. Lang (2001). Bayesian inference for generalized additive mixed models based on markov random field priors. *Applied Statistics 50*, 201–220.

Firket, J. (1936). Fog along the Meuse valley. *Trans Faraday Soc 32*, 1191–1194.

Green, P. J. and B. W. Silverman (1994). *Nonparametric Regression and Generalized Linear Models*. Chapman & Hall.

Gu, C. (1992). Cross-validating non-gaussian data. *Journal of Computational and Graphical Statistics 1*, 169–179.

Gu, C. and Y. J. Kim (2002). Penalized likelihood regression: general approximation and efficient approximation. *Canadian Journal of Statistics 34*(4), 619–628.

Hall, P. and J. D. Opsomer (2005). Theory for penalised spline regression. *Biometrika 92*(1), 105–118.

Handcock, M. S., K. Meier, and D. Nychka (1994). Comment. *Journal of the American Statistical Association 89*(426), 401–403.

Hastie, T. and R. Tibshirani (1986). Generalized additive models (with discussion). *Statistical Science 1*, 297–318.

Hastie, T. and R. Tibshirani (1990). *Generalized Additive Models*. Chapman & Hall.

Helwig, N. E. and P. Ma (2016). Smoothing spline anova for super-large samples: Scalable computation via rounding parameters. *arXiv preprint arXiv:1602.05208*.

Kammann, E. E. and M. P. Wand (2003). Geoadditive models. *Applied Statistics 52*(1), 1–18. Matern splines.

Kauermann, G., T. Krivobokova, and L. Fahrmeir (2009). Some asymptotic results on generalized penalized spline smoothing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71*(2), 487–503.

Kimeldorf, G. S. and G. Wahba (1970). A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics 41*(2), 495–502.

Lang, S., N. Umlauf, P. Wechselberger, K. Harttgen, and T. Kneib (2014). Multilevel structured additive regression. *Statistics and Computing 24*(2), 223–238.

Li, Y. and D. Ruppert (2008). On the asymptotics of penalized splines. *Biometrika 95*(2), 415–436.

Loader, A. (2002). *Instruction manual: UK Smoke and Sulphur Dioxide Network*. Culham Science Centre: Netcen, AEA Technology.

Lucas, C. (2004). Lapack-style codes for level 2 and 3 pivoted cholesky factorizations. *LAPACK Working*.

Marra, G. and S. N. Wood (2012). Coverage properties of confidence intervals for generalized additive model components. *Scandinavian Journal of Statistics 39*(1), 53–74.

Marx, B. D. and P. H. Eilers (1998). Direct generalized additive modeling with penalized likelihood. *Computational Statistics and Data Analysis 28*, 193–209.

McCullagh, P. and J. A. Nelder (1989). *Generalized linear models* (2nd ed.). London: Chapman & Hall.

Miller, D. L. and S. N. Wood (2014). Finite area smoothing with generalized distance splines. *Environmental and Ecological Statistics*, 1–17.

Ministry of Health (1954). *Mortality and morbidity during the London fog of December 1952*. London: HMSO.

Nocedal, J. and S. Wright (2006). *Numerical optimization* (2nd ed.). New York: Springer verlag.

Nychka, D. (1988). Bayesian confidence intervals for smoothing splines. *Journal of the American Statistical Association 83*(404), 1134–1143.

of Terrain-50, C. (2015). *OS Terrain 50: user guide and technical specification*. Adanac Drive, Southampton, SO16 0AS: Ordnance Survey.

OpenMP Architecture Review Board (2008, May). OpenMP application program interface version 3.0.

Parker, R. and J. Rice (1985). Discussion of Silverman (1985). *Journal of the Royal Statistical Society, Series B 47*(1), 40–42.

Perry, M. and D. Hollis (2005a). The development of a new set of long-term climate averages for the uk. *International Journal of Climatology 25*(8), 1023–1039.

Perry, M. and D. Hollis (2005b). The generation of monthly gridded datasets for a range of climatic variables over the uk. *International Journal of Climatology 25*(8), 1041–1054.

Quintana-Ortí, G., X. Sun, and C. H. Bischof (1998). A BLAS-3 version of the QR factorization with column pivoting. *SIAM Journal on Scientific Computing 19*(5), 1486–1494.

Reiss, P. T. and T. R. Ogden (2009). Smoothing parameter selection for a class of semiparametric linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71*(2), 505–523.

Rue, H., S. Martino, and N. Chopin (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.

Ruppert, D., M. P. Wand, and R. J. Carroll (2003). *Semiparametric Regression*. Cambridge University Press.

Shaddick, G. and J. V. Zidek (2014). A case study in preferential sampling: Long term monitoring of air pollution in the uk. *Spatial Statistics 9*, 51–65.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society B 47*(1), 1–53.

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

Wahba, G. (1981). Spline interpolation and smoothing on the sphere. *SIAM Journal on Scientific and Statistical Computing 2*(1), 5–16.

Wahba, G. (1983). Bayesian confidence intervals for the cross validated smoothing spline. *Journal of the Royal Statistical Society B 45*, 133–150.

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

Wang, X., J. Shen, and D. Ruppert (2011). On the asymptotics of penalized spline smoothing. *Electronic Journal of Statistics 5*, 1–17.

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society, Series B 65*, 95–114.

Wood, S. N. (2006). Low-rank scale-invariant tensor product smooths for generalized additive mixed models. *Biometrics 62*(4), 1025–1036.

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 (Statistical Methodology) 73*(1), 3–36.

Wood, S. N. (2013). On p-values for smooth components of an extended generalized additive model. *Biometrika 100*(1), 221–228.

Wood, S. N. (2015). *Core Statistics*. Cambridge University Press.

Wood, S. N., Y. Goude, and S. Shaw (2015). Generalized additive models for large data sets. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*.

Wood, S. N., F. Scheipl, and J. J. Faraway (2013). Straightforward intermediate rank tensor product smoothing in mixed models. *Statistics and Computing 23*(3), 341–360.

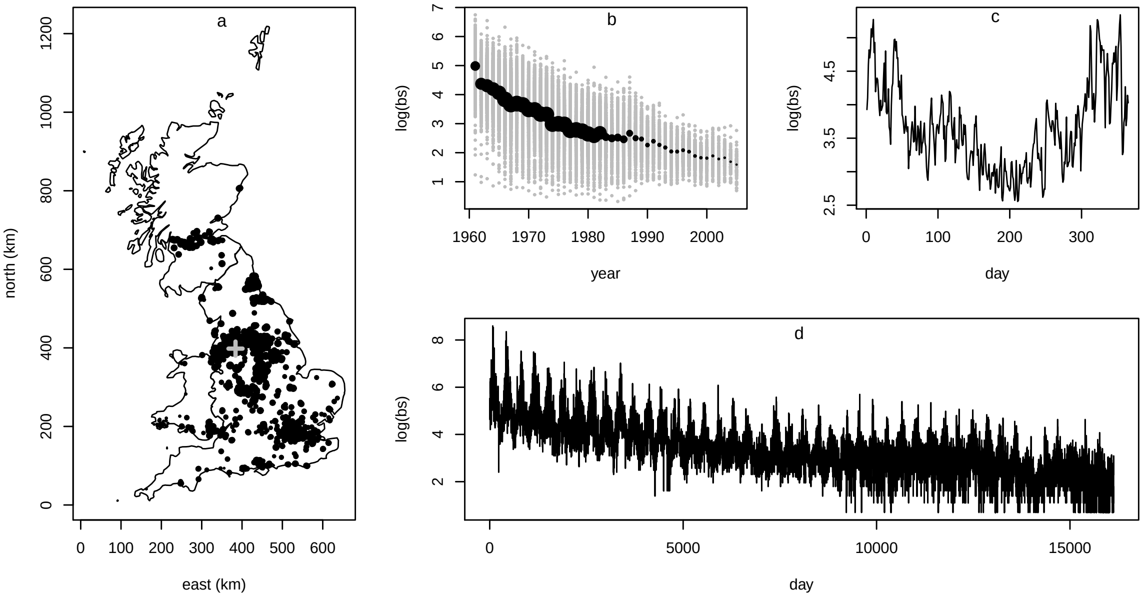

Figure 1: **a.** The UK black smoke network monitoring network at its largest in 1967. Symbol sizes are proportional to annual average log black smoke. **b.** Annual average log black smoke against year. Black dots are averages over space, with dot size proportional to network size. Grey dots are station averages. **c.** Daily averages for 1967, across all stations shown in a. **d.** All daily measurements for the longest running site, shown as a grey '+' in a.

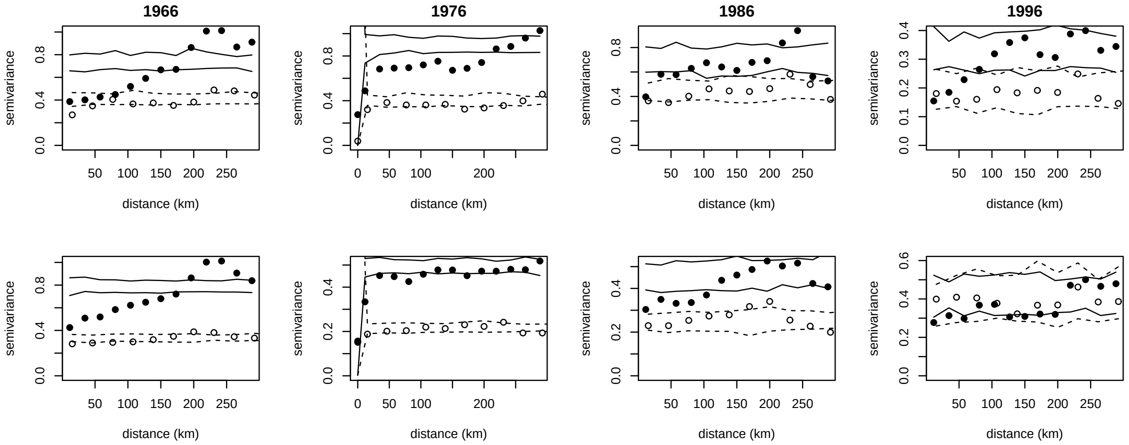

Figure 2: Semivariograms for the 40th (top row) and 200th (bottom row) days of years 1966, 76, 86 and 96, checking for residual spatial auto-correlation. Each plot shows the empirical semivariogram for the log black smoke measurements as black dots, with the corresponding reference bands under zero auto-correlation 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.

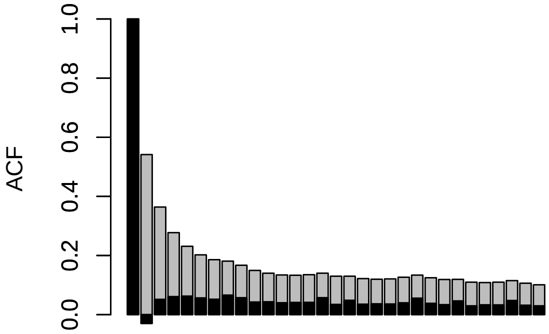

Figure 3: Aggregate ACF for model (1) residuals assuming independent residuals in grey, with the equivalent for the standardized residuals assuming AR1 residuals, overlaid in black. While not perfect, the AR1 model greatly reduces the un-modelled temporal autocorrelation.

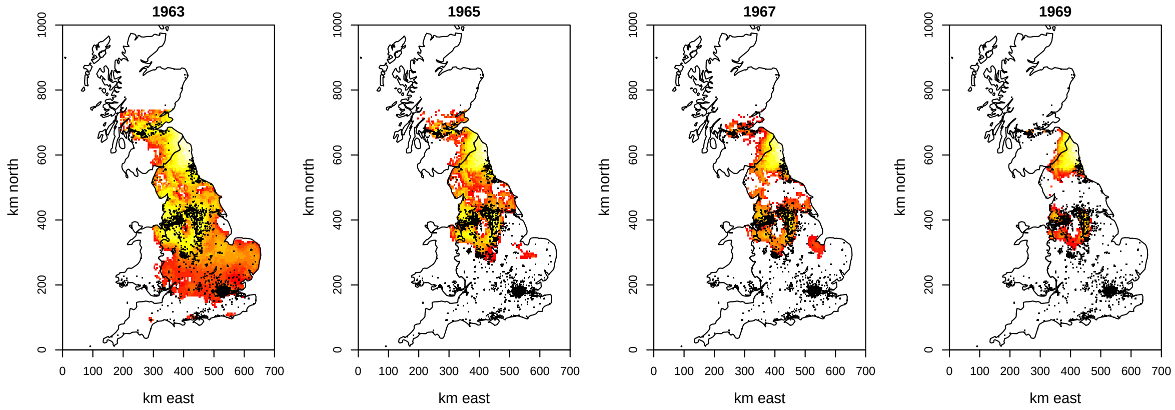

Figure 4: Image plots of log predicted number of days exceeding the EU daily exposure threshold for town centre locations for several years in the 1960s. By 1975 there were essentially no exceedance days predicted.

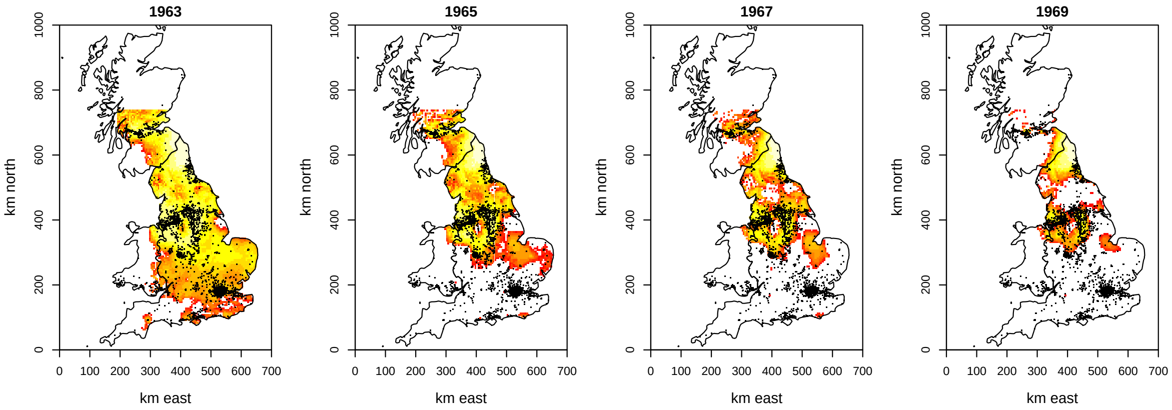

Figure 5: Image plots of log average probability of exceeding the EU daily exposure threshold for town centre locations for several years in the 1960s. Red is -6 corresponding to less than one exceedance day expected per year, while the top of the scale is 0.