GAMs with integrated model selection using penalized regression splines and applications to environmental modelling.

Simon N. Wood [a], Nicole H. Augustin [b]

[a] *Mathematical Institute, North Haugh, St Andrews, Fife KY16 9SS, UK.*

[b] *Freiburg Centre for Data Analysis and Modelling, University of Freiburg, Eckerstr. 1, 79104 Freiburg, Germany.*

Abstract

Generalized Additive Models (GAMs) have been popularized by the work of Hastie and Tibshirani (1990) and the availability of user friendly GAM software in Splus. However, whilst it is flexible and efficient, the GAM framework based on backfitting with linear smoothers presents some difficulties when it comes to model selection and inference. On the other hand, the mathematically elegant work of Wahba (1990) and co-workers on Generalized Spline Smoothing (GSS) provides a rigorous framework for model selection (Gu and Wahba, 1991) and inference with GAMs constructed from smoothing splines: but unfortunately these models are computationally very expensive with operations counts that are of cubic order in the number of *data*. A "middle way" between these approaches is to construct GAMs using penalized regression splines (see e.g. Wahba 1980, 1990; Eilers and Marx 1998, Wood 2000). In this paper we develop this idea and show how GAMs constructed using penalized regression splines can be used to get most of the practical benefits of GSS models, including well founded model selection and multi-dimensional smooth terms, with the ease of use and low computational cost of backfit GAMs. Inference with the resulting methods also requires slightly fewer approximations than are employed in the GAM modelling software provided in Splus. This paper presents the basic mathematical and numerical approach to GAMs implemented in the R package mgcv, and includes two environmental examples using the methods as implemented in the package.

Key words: smoothing, additive, model, selection, GCV

1 Introduction

Consider a univariate response y_i from some exponential family distribution where $\mu_i \equiv E(y_i)$ is determined by some explanatory variables x_{1i}, x_{2i} etc. Replacing the strictly parametric GLM model structure:

$$g(\mu_i) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots \tag{1}$$

with the "less-parametric" GAM structure:

$$g(\mu_i) = \beta_0 + s_1(x_{1i}) + s_2(x_{2i}) + \dots \tag{2}$$

adds a great deal of flexibility to the family of possible models, but adds quite substantially to the problem of model selection, and to a lesser extent inference (the s_i are unknown smooth functions, while g is a known monotonic differentiable "link function" and the β_i are parameters to be estimated). In particular if the s_i are estimated using linear smoothers and backfitting then the effective degrees of freedom of the smooths have to be selected, and this is not straightforward. The reason for the difficulty stems from the very feature that makes back-fitting so appealing - that is that it provides a way of estimating all the smooth terms in the model, making use only of algorithms suitable for estimating single smooth terms individually. The problem arises because the criteria (e.g. AIC, GCV) that one would like to apply in order to select the effective degrees of freedom of the model, measure properties of the *whole model*, not of single smooth terms. Hence while back-fitting makes estimation of smooth terms straightforward *given* the bandwidth of the smoothers, estimation of that bandwidth is hard to integrate into a back-fitting approach. Of course it is always possible to do model selection by searching through a large space of models using some stepwise procedure, but when each smooth term could have one of a fairly large number of alternative degrees of freedom, this can become very tedious.

These problems have been recognised for some time, are discussed e.g. in Hastie and Tibshirani (1990) and are the motivation for techniques like the approximate GCV method BRUTO. However, in the context of GSS models the model selection problem has actually been solved by Gu and Wahba (1991) who developed an algorithm for estimating multiple smoothing parameters using the GCV score for whole GSS models, including GAMs constructed from smoothing splines. Unfortunately the calculations involved in the GSS approach are cubic in n, the number of data being modelled, and this presents a practical barrier to the use of these methods in many applied contexts. (Of course spline based GAMs can be *fitted* much more efficiently using back-fitting - but then the model selection using the full GCV score becomes very inefficient).

The $O(n^3)$ operations count of Gu and Wahba's method results from the fact that GSS models necessarily have as many parameters as there are data to be modelled, although generally fewer effective degrees of freedom, of course. If the models had fewer parameters then the calculations would be faster. This suggests using penalized regression splines (e.g. Eilers & Marx, 1996, Marx & Eilers, 1988, Wahba 1980, 1990) in place of full splines to represent the GAM, thereby reducing the parameter count, but unfortunately this changes the problem structure sufficiently that Gu and Wahba's method is no longer usable (since it relies on the rather special structure of the full spline smoothing problem). However, with some effort it is possible to generalize Gu and Wahba's method to cover a much larger class of generalized ridge regression problems than just those resulting from full spline smoothing. The generalization is reported in Wood (2000) and permits smoothing parameter selection for GAMs formulated using penalized regression splines.

The purpose of this paper is to document exactly how GAMs can be constructed using penalized regression splines in a way that allows (i) integrated model selection via GCV and related criteria, (ii) straightforward incorporation of multi-dimensional smooths in GAMs and (iii) relatively well founded inference using the resulting models. The paper is structured as follows. The next section reviews basic material on modelling with basis functions, and is intended to give the reader a feeling for the ideas which lead naturally to penalized regression splines. Then the construction of GAMs using penalized regression splines is discussed, including material on how model selection can be performed using GCV, and how to obtain confidence intervals. The final theoretical section discusses issues to do with multi-dimensional smoothing. We take the approach of introducing most of the material in the context of linear models (and hence additive models) and only subsequently covering generalized linear modelling and hence generalized additive models. The paper finishes by applying GCV selected penalized regression spline GAMs to European Mackerel Egg Survey data and beech canker data.

2 Modelling with basis functions

In this paper GAMs are constructed using basis functions, so this section is intended to provide an introduction to modelling with basis functions. Readers familiar with this material may want to skim or skip it.

It is often desirable to include a smooth function of a covariate or covariates into a model without being very specific about the exact form of the function. That is, it may be appropriate to include terms like $f(x)$ or $g(x, z)$ in the specification of a model. To do this in practice so that f or g can be estimated, requires a practical means of representing functions like f and g: basis functions can provide this.

Consider representing a function of one variable, $f(x)$, say. Let $\{b_j(x) : i = j \ldots m\}$ be a set of functions that are chosen to have convenient properties, and to have no unknown parameters. $f(x)$ can be represented as:

$$f(x) = \sum_{j=1}^{m} \alpha_j b_j(x) \tag{3}$$

where the α_j are m unknown coefficients. So $f(x)$ is made up of a linear combination of the basis functions $b_j(x)$, and estimating f is now equivalent to finding the α_j.

To see how a function might be estimated in practice, consider the following simple model:

$$y_i \sim N(f(x_i), \sigma^2)$$

and suppose that there are n observations (y_i, x_i). The model can be estimated by minimising:

$$\begin{aligned} \sum_{i=1}^{n} (f(x_i) - y_i)^2 &= \sum_{i=1}^{n} \left(\sum_{j=1}^{m} \alpha_j b_j(x_i) - y_i \right)^2 \\ &= \sum_{i=1}^{n} \left(\mathbf{b}(x_i)^T \boldsymbol{\alpha} - y_i \right)^2 \end{aligned}$$

where $\boldsymbol{\alpha}$ is the vector of coefficients α_i, and $\mathbf{b}(x_i)$ is the vector containing each basis function evaluated at x_i (i.e. $\mathbf{b}(x_i) = [b_1(x_i), b_2(x_i), \ldots, b_m(x_i)]^T$). It is straightforward to see that this is a standard linear model fitting problem. First define:

$$\mathbf{X} = \begin{bmatrix} b_1(x_1) & b_2(x_1) & . & . & b_m(x_1) \\ b_1(x_2) & b_2(x_2) & . & . & b_m(x_2) \\ b_1(x_3) & b_2(x_3) & . & . & b_m(x_3) \\ . & . & . & . & . \\ . & . & . & . & . \\ b_1(x_n) & b_2(x_n) & . & . & b_m(x_n) \end{bmatrix} = \begin{bmatrix} \mathbf{b}(x_1)^T \\ \mathbf{b}(x_2)^T \\ \mathbf{b}(x_3)^T \\ . \\ . \\ \mathbf{b}(x_n)^T \end{bmatrix}$$

and now recall that:

$$\sum_{i=1}^{n} z_i^2 \equiv \mathbf{z}^T \mathbf{z} \equiv \|\mathbf{z}\|^2$$

(i.e. $\|\cdot\|^2$ is the usual Euclidian norm). We have:

$$\sum_{i=1}^{n}\Big(\mathbf{b}(x_i)^T\boldsymbol{\alpha}-y_i\Big)^2 = (\mathbf{X}\boldsymbol{\alpha}-\mathbf{y})^T(\mathbf{X}\boldsymbol{\alpha}-\mathbf{y}) \\ = \|\mathbf{X}\boldsymbol{\alpha}-\mathbf{y}\|^2$$

and minimisation of this last expression yields the least squares estimates of $\boldsymbol{\alpha}$.

2.1 Example: polynomial regression

Basis functions are usually chosen for their theoretical properties, or for practical reasons, or for a mixture of both. The familiar polynomial basis is an example of a basis that is very easy to use, but has poor approximation theoretic and numerical stability properties. For example, suppose that we want to use a 5 dimensional polynomial basis to represent $f(x)$. Suitable basis functions are: $b_1(x) = 1$, $b_2(x) = x$, $b_3(x) = x^2$, $b_4(x) = x^3$ and $b_5(x) = x^4$. Then:

$$f(x) = \sum_{j=1}^{5}\alpha_j b_j(x)$$

is just an elaborate way of writing:

$$f(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3 + \alpha_5 x^4$$

Fitting this model by least squares is a matter of minimising:

$$\left\| \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ 1 & x_n & x_n^2 & x_n^3 & x_n^4 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ . \\ . \\ \alpha_5 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix} \right\|^2 \quad \text{(i.e. } \|\mathbf{X}\boldsymbol{\alpha}-\mathbf{y}\|^2\text{)}$$

The polynomial basis is often a poor choice: as the dimension of the basis increases the basis functions become increasingly co-linear. This tends to lead to highly correlated parameter estimators, which in turn lead to high estimator variance and numerical problems. Furthermore, high order polynomials have a tendency to oscillate wildly if there are wide gaps in the x_i's.

2.2 Example: the cubic spline basis

The spline bases have good theoretical and practical properties (see e.g. Wahba, 1990; de Boor 1978). For functions of one variable the cubic spline basis is popular. There are a number of alternative sets of basis functions that can be used as cubic spline basis functions. We give one of the the simplest to understand, although there are alternatives with better numerical stability properties. Again, consider representing $f(x)$, but now let $\{x_j^* : j = 1 \dots m\}$, be a set of points in the range of x, sometimes known as the "knots" of the spline. Representing $f(x)$ using cubic splines amounts to representing it using sections of cubic polynomial joined at the knots so that they are continuous up to and including second derivative. Mathematically, this is achieved by letting $b_j(x) = |x - x_j^*|^3$ for $j = 1, \dots, m$, $b_{m+1}(x) = 1$, $b_{m+2}(x) = x$ and:

$$f(x) = \sum_{j=1}^{m+2} \alpha_j b_j(x).$$

$f(x)$ represented in this way is a "natural" cubic spline provided that the constraints $\sum_{j=1}^{m} \alpha_j = 0$ and $\sum_{j=1}^{m} \alpha_j x_j^* = 0$ are imposed on the coefficients. The "natural" spline constraint means that the spline has zero second derivative outside the interval $[x_1^*, x_m^*]$, which is a sensible requirement since it reduces the dangers associated with extrapolation.

There are three spline based approaches to modelling n data points (x_i, y_i) with a model of the form:

$$E(y_i) = f(x_i)$$

These are smoothing splines (in the Wahba, 1990, sense), regression splines or penalized regression splines: they can all be constructed using the same sort of basis functions, but differ in where the knots are placed and how model complexity is controlled. For the moment consider only a regression spline, which is fitted by finding the α_i that minimise:

$$\sum_{i=1}^{n} (f(x_i) - y_i)^2.$$

It should be clear by now that this reduces to an ordinary least squares problem:

$$\text{minimise} \left\| \begin{bmatrix} |x_1 - x_1^*|^3 & |x_1 - x_2^*|^3 & |x_1 - x_3^*|^3 & . & . & 1 & x_1 \\ |x_2 - x_1^*|^3 & |x_2 - x_2^*|^3 & |x_2 - x_3^*|^3 & . & . & 1 & x_2 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ |x_n - x_1^*|^3 & |x_n - x_2^*|^3 & |x_n - x_3^*|^3 & . & . & 1 & x_n \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ . \\ . \\ . \\ \alpha_{m+2} \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ . \\ . \\ y_n \end{bmatrix} \right\|^2$$

subject to the "natural" spline constraints:

$$\begin{bmatrix} 1 & 1 & 1 & . & . & . & 1 & 0 & 0 \\ x_1^* & x_2^* & x_3^* & . & . & . & x_m^* & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ . \\ . \\ . \\ \alpha_{m+2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Solving least squares problems subject to linear constraints will be covered in section 2.5.

Regression spline modelling is appealingly simple, but in practice the choice of knot locations, x_j^*, can have a substantial influence on modelling results. Hence, the need to choose knot locations can be a serious complicating factor in regression spline modelling (Hastie and Tibshirani 1990, chapter 9, provides a helpful discussion of this problem). One standard method for avoiding knot placement problems is to use smoothing splines or penalized regression splines (both options are discussed, for example, in Wahba, 1990 or Green and Silverman, 1994). With both of these alternatives a relatively large number of knots is used, but excessively wiggly fitted models are avoided by applying a "wiggliness" penalty to the model fitting objective. Having a large number of knots means that the fitted model is quite insensitive to the exact choice of knot locations, but the penalty can be used to avoid the danger of over-fitting that would otherwise accompany the use of many knots. In the smoothing spline case the number of knots is actually the number of unique covariate (x) values, while in the penalized regression spline case a smaller number of knots is chosen, usually to keep down computational cost. The next two sections show how to construct appropriate penalties, and how to use those penalties in fitting.

2.3 Linear operations on $f(x)$ and wiggliness measures

The purpose of this section is to show how measures of function wiggliness can be constructed that are easy to interpret and straightforward to evaluate computationally. The presentation is quite general, to emphasise that such measures can be obtained for a wide variety of bases. Given the measures presented in this section, penalized regression splines will be described in the following section.

In order to develop measures of the wiggliness of $f(x)$, it is helpful to first consider linear operations on f. In particular, consider differentiation and integration of f (as defined by (3)). Clearly:

$$f'(x) = \sum_{j=1}^{m} \alpha_j b_j'(x) \quad f''(x) = \sum_{j=1}^{m} \alpha_j b_j''(x) \text{ and } \int f(x)dx = \sum_{j=1}^{m} \alpha_j \int b_j(x)dx$$

i.e. differentials or integrals of f w.r.t. x are linear in α_i.

Given this linearity, it is possible to construct a penalty on f which will be large if f is very wiggly and small if it is nearly flat and that has a convenient representation in terms of the α_i. One popular possibility is:

$$J(f) = \int [f''(x)]^2 dx$$

To see how this can be calculated for a given f with a given basis, write:

$$f''(x) = \sum_{j=1}^{m} \alpha_j b_j''(x) = \mathbf{b}''(x)^T \boldsymbol{\alpha}$$

where $\mathbf{b}''(x)$ is the vector of second derivatives of the basis functions evaluated at x. Since $f''(x)$ is a scalar it is equal to its own transpose, so:

$$[f''(x)]^2 = \boldsymbol{\alpha}^T \mathbf{b}''(x)^T \mathbf{b}''(x) \boldsymbol{\alpha} = \boldsymbol{\alpha}^T \mathbf{S}(x) \boldsymbol{\alpha}$$

where

$$\mathbf{S}(x) = \begin{bmatrix} b_1''(x)^2 & b_1''(x)b_2''(x) & b_1''(x)b_3''(x) & . & b_1''(x)b_m''(x) \\ b_2''(x)b_1''(x) & b_2''(x)^2 & b_2''(x)b_3''(x) & . & b_2''(x)b_m''(x) \\ b_3''(x)b_1''(x) & b_3''(x)b_2''(x) & b_3''(x)^2 & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ b_m''(x)b_1''(x) & b_m''(x)b_2''(x) & . & . & .b_m''(x)^2 \end{bmatrix}$$

Hence:

$$J(f) = \boldsymbol{\alpha}^T \int \mathbf{S}(x)dx\boldsymbol{\alpha} = \boldsymbol{\alpha}^T\mathbf{H}\boldsymbol{\alpha}, \text{ say.}$$

So, given a basis, we can always evaluate the coefficient matrix $\mathbf{H}$ which allows the penalty $J(f)$ to be written as a quadratic form in the parameter vector $\boldsymbol{\alpha}$ (where the $m \times m$ matrix $\mathbf{H}$ does not depend on $\boldsymbol{\alpha}$). In the case of the spline bases the elements of $\mathbf{H}$ can be found in Green and Silverman (1994) or Wahba (1990), for example. The utility of the result is that it provides a practical way of applying wiggliness penalties as part of model fitting, as is shown in the next section.

Notice that other wiggliness measures can be developed using the same general approach. For example $\int[f'(x)]^2dx$ or $\int[f'''(x)]^2dx$ could be treated in the same way as $J(f)$. Another possibility, advocated by Eilers and Marx (1996), is to obtain an approximate $\mathbf{H}$ based on a discrete approximate wiggliness penalty: if the correct version of the spline basis is chosen (the b-spline representation, see de Boor 1978) this method has the advantage of being very easy to program.

2.4 Combining basis and wiggliness penalty

When modelling with basis functions it is possible to control the wiggliness of the fitted model by controlling the number of basis functions used, but as was discussed in section 2.2, this can cause difficulties. Specifically, if the number of basis functions is large enough to be able to closely approximate the unknown underlying true function, then it is likely that the model will overfit data that contain any noise. Conversely, if the number of basis functions is chosen to be low enough to avoid this overfitting, it is likely that the basis will be too restrictive to closely approximate the underlying truth. These problems can be alleviated by using a relatively large number of basis functions, but avoiding overfit by imposing a penalty during model fitting that is designed to ensure that the fitted model is smooth.

For example, the model:

$$E(y_i) = f(x_i)$$

where f is a smooth function, could be estimated my minimising:

$$\sum_{i=1}^{n}(f(x_i) - y_i)^2 + \lambda \int [f''(x)]^2 dx \tag{4}$$

where λ is a smoothing parameter that controls the trade-off between closely matching the data and having a smooth model. Choosing a basis for f allows a design matrix $\mathbf{X}$ and a penalty matrix $\mathbf{H}$ to be calculated (as described previously). So the fitting problem can be written:

$$\text{minimise } \|\mathbf{X}\boldsymbol{\alpha} - \mathbf{y}\|^2 + \lambda\boldsymbol{\alpha}^T\mathbf{H}\boldsymbol{\alpha} \tag{5}$$

Given λ, this is straightforward to solve: the objective can be re-written as:

$$(\mathbf{X}\boldsymbol{\alpha} - \mathbf{y})^T(\mathbf{X}\boldsymbol{\alpha} - \mathbf{y}) + \lambda\boldsymbol{\alpha}^T\mathbf{H}\boldsymbol{\alpha} = \boldsymbol{\alpha}^T[\mathbf{X}^T\mathbf{X} + \lambda\mathbf{H}]\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{y} + \mathbf{y}^T\mathbf{y}$$

and this can be minimised by differentiating w.r.t. $\boldsymbol{\alpha}$ and setting the resulting system of equations to zero to get:

$$\hat{\boldsymbol{\alpha}} = [\mathbf{X}^T\mathbf{X} + \lambda\mathbf{H}]^{-1}\mathbf{X}^T\mathbf{y}$$

How to deal with any constraints on the problem will be deferred until section 2.5, while the issue of how to estimate λ will be covered in section 3.1.

The treatment given above is quite general, and could be used with a variety of bases and penalties (including e.g. the P-splines of Eilers and Marx 1996). However in the rest of this paper we will consider only spline bases. The reason for this is partly related to a special property of the spline bases. Consider trying to find the function minimising (4) out of all functions — not just those that can be represented using a particular set of basis functions. It turns out that this function exists, and furthermore that it can be represented with a finite dimensional basis: the function is a natural cubic spline, with a knot at each x_i value. This result can be generalized to penalties of different orders and smooth functions of any number of variables as is discussed in section 3.4. It is this optimality property that suggests that the spline bases are a natural choice for representing smooth functions.

2.5 Constraints

The spline basis used in previous sections involves an m dimensional basis represented using $m + 2$ basis functions/ parameters and 2 linear equality constraints on the parameters. We have chosen to present this basis because it makes the model and the penalties quite simple to write down, and also because it is a special case of the more general spline bases covered in section 3.4. However, its use does involve fitting subject to constraints, which we therefore cover in this section. Another reason for discussing constrained fitting at this stage is that in general constraints are required in order to ensure identifiability of GAM models.

To motivate the discussion consider the constraints given at the end of section 2.2. These can be written compactly as:

$$\mathbf{C}\boldsymbol{\alpha} = \mathbf{0}$$

It is necessary to find a way of representing $\boldsymbol{\alpha}$ which ensures that it always meets this constraint, but imposes no further un-necessary restriction on $\boldsymbol{\alpha}$. Suppose that $\mathbf{C}$ is a $q \times m$ matrix, where $q < m$. It is always possible to find an orthogonal matrix $\mathbf{Q}$ such that:

$$\mathbf{C}\mathbf{Q} = [\mathbf{0}_{q,m-q}, \mathbf{T}]$$

where $\mathbf{T}$ is a $q \times q$ matrix and $\mathbf{0}_{q,m-q}$ is a $q \times (m-q)$ matrix of zeroes (the factorisation is like a QR factorization — with which it can be replaced — and is performed using Householder rotations, see Watkins 1991, for example). Now partition $\mathbf{Q}$ into two parts: an $m \times (m-q)$ part $\mathbf{Z}$ and an $m \times q$ part $\mathbf{Y}$, so that:

$$\mathbf{Q} = [\mathbf{Z}, \mathbf{Y}]$$

This means that:

$$\mathbf{C}\mathbf{Z} = \mathbf{0} \text{ and } \mathbf{C}\mathbf{Y} = \mathbf{T}$$

Which in turn means that if we let $\boldsymbol{\alpha} = \mathbf{Z}\boldsymbol{\alpha}_z$, where $\boldsymbol{\alpha}_z$ is an $m-q$ vector of unknown parameters, then $\mathbf{C}\boldsymbol{\alpha} = \mathbf{0}$, for any $\boldsymbol{\alpha}_z$.

Now consider solving (5) subject to $\mathbf{C}\boldsymbol{\alpha} = \mathbf{0}$. The problem can be re-written as:

$$\text{minimise } \|\mathbf{X}\mathbf{Z}\boldsymbol{\alpha}_z - \mathbf{y}\|^2 + \lambda\boldsymbol{\alpha}_z^T\mathbf{Z}^T\mathbf{H}\mathbf{Z}\boldsymbol{\alpha}_z$$

and solved for $\boldsymbol{\alpha}_z$ exactly as discussed in section 2.4, but using $\mathbf{XZ}$ in place of $\mathbf{X}$ and $\mathbf{Z}^T\mathbf{H}\mathbf{Z}$ in place of $\mathbf{H}$. Obviously, $\hat{\boldsymbol{\alpha}} = \mathbf{Z}\hat{\boldsymbol{\alpha}}_z$.

3 GAMs built from penalized regression splines

We are now in a position to see how Generalized Additive Models can be constructed using penalized regression splines. All the essential points are covered by considering a GAM with two smooth terms to be fitted to Gaussian data with an identity link (generalization to other exponential family members will be covered at the end of this section). Consider modelling data (y_i, x_{1i}, x_{2i}) using the model:

$$E(y_i) = \mu_i = \beta_0 + f_1(x_{1i}) + f_2(x_{2i}) \quad y_i \sim N(\mu_i, \sigma^2)$$

where f_1 and f_2 are smooth functions, which could be represented using penalized cubic regression splines so that:

$$f_1(x) = \sum_{i=j}^{q_1} \beta_j b_{1j}(x) \qquad f_2(x) = \sum_{j=1}^{q_2} \beta_{j+q_1} b_{2j}(x)$$

where the $b_{1j}(\cdot)$ and $b_{2j}(\cdot)$ are cubic spline basis functions for f_1 and f_2 respectively. The fitting objective for this model will be:

$$\text{minimise} \sum_{i=1}^{n}(y_i - \beta_0 - f_1(x_{1i}) - f_2(x_{2i}))^2 + \lambda_1 \int \left(\frac{\partial^2 f_1}{\partial x_1^2}\right)^2 dx_1 + \lambda_2 \int \left(\frac{\partial^2 f_2}{\partial x_2^2}\right)^2 dx_2$$

subject to any constraints associated with the bases (e.g. the “natural spline” constraints) plus any constraints needed to ensure that the model is identifiable (for example that the functions must have zero mean).

Using the basis given in section 2.2 this GAM fitting problem becomes:

$$\begin{array}{rl} \text{minimise} & \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|^2 + \lambda_1 \boldsymbol{\beta}'\mathbf{H}_1\boldsymbol{\beta} + \lambda_2 \boldsymbol{\beta}'\mathbf{H}_2\boldsymbol{\beta} \\ \text{subject to} & \mathbf{C}\boldsymbol{\beta} = \mathbf{0} \end{array}$$

where: $\mathbf{X} =$

$$\begin{bmatrix} 1 & |x_{11} - x_{11}^*|^3 & |x_{11} - x_{12}^*|^3 & . & . & |x_{11} - x_{1,q_1-2}^*|^3 & 1 & x_{11} & |x_{21} - x_{21}^*|^3 & . & . & |x_{21} - x_{2,q_2-2}^*|^3 & 1 & x_{21} \\ 1 & |x_{12} - x_{11}^*|^3 & |x_{12} - x_{12}^*|^3 & . & . & |x_{12} - x_{1,q_1-2}^*|^3 & 1 & x_{12} & |x_{22} - x_{21}^*|^3 & . & . & |x_{22} - x_{2,q_2-2}^*|^3 & 1 & x_{22} \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . & . & . & . \\ 1 & |x_{1n} - x_{11}^*|^3 & |x_{1n} - x_{12}^*|^3 & . & . & |x_{1n} - x_{1,q_1-2}^*|^3 & 1 & x_{1n} & |x_{2n} - x_{21}^*|^3 & . & . & |x_{2n} - x_{2,q_2-2}^*|^3 & 1 & x_{2n} \end{bmatrix}$$

$$\mathbf{H}_1 = \begin{bmatrix} 0 & 0 & 0 & . & . & 0 & 0 & . & . & 0 \\ 0 & |x_{11}^* - x_{11}^*|^3 & |x_{12}^* - x_{11}^*|^3 & . & . & |x_{1,q_1-2}^* - x_{11}^*|^3 & 0 & . & . & 0 \\ 0 & |x_{11}^* - x_{12}^*|^3 & |x_{12}^* - x_{12}^*|^3 & . & . & |x_{1,q_1-2}^* - x_{12}^*|^3 & 0 & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & \\ 0 & |x_{11}^* - x_{1,q_1-2}^*|^3 & |x_{12}^* - x_{1,q_1-2}^*|^3 & . & . & |x_{1,q_1-2}^* - x_{1,q_1-2}^*|^3 & 0 & . & . & 0 \\ 0 & 0 & 0 & . & . & 0 & 0 & . & . & 0 \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & \\ 0 & 0 & 0 & . & . & 0 & 0 & . & . & 0 \end{bmatrix}$$

$$\mathbf{H}_2 = \begin{bmatrix} 0\ .\ .\ 0 & 0 & 0 & .\ . & 0 & 0\ 0 \\ .\ .\ .\ . & . & . & .\ . & . & \\ .\ .\ .\ . & . & . & .\ . & . & .\ . \\ 0\ .\ .\ 0 & 0 & 0 & .\ . & 0 & 0\ 0 \\ 0\ .\ .\ 0 & |x^*_{21} - x^*_{21}|^3 & |x^*_{22} - x^*_{21}|^3 & .\ . & |x^*_{2,q_2-2} - x^*_{21}|^3 & 0\ 0 \\ 0\ .\ .\ 0 & |x^*_{21} - x^*_{22}|^3 & |x^*_{22} - x^*_{22}|^3 & .\ . & |x^*_{2,q_2-2} - x^*_{22}|^3 & 0\ 0 \\ .\ .\ .\ . & . & . & .\ . & . & \\ .\ .\ .\ . & . & . & .\ . & . & .\ . \\ 0\ .\ .\ 0 & |x^*_{21} - x^*_{2,q_2-2}|^3 & |x^*_{22} - x^*_{2,q_2-2}|^3 & .\ . & |x^*_{2,q_2-2} - x^*_{2,q_2-2}|^3 & 0\ 0 \\ 0\ .\ .\ 0 & 0 & 0 & .\ . & 0 & 0\ 0 \\ 0\ .\ .\ 0 & 0 & 0 & .\ . & 0 & 0\ 0 \end{bmatrix}$$

and the constraint matrix $\mathbf{C}$ is defined as:

$$\begin{bmatrix} 0 & 1 & .\ . & 1 & 0 & 0 & 0 & .\ . & 0 & 0 & 0 \\ 0 & x^*_{11} & .\ . & x^*_{1,q_1-2} & 0 & 0 & 0 & .\ . & 0 & 0 & 0 \\ 0 & 0 & .\ . & 0 & 0 & 0 & 1 & .\ . & 1 & 0 & 0 \\ 0 & 0 & .\ . & 0 & 0 & 0 & x^*_{21} & .\ . & x^*_{2,q_2-2} & 0 & 0 \\ 0 & \sum |x_{1i} - x^*_{11}|^3 & .\ . & \sum |x_{1i} - x^*_{1,q_1-2}|^3 & n & \sum x_{1i} & 0 & .\ . & 0 & 0 & 0 \\ 0 & 0 & .\ . & 0 & 0 & 0 & \sum |x_{2i} - x^*_{21}|^3 & .\ . & \sum |x_{2i} - x^*_{2,q_2-2}|^3 & n & \sum x_{2i} \end{bmatrix}$$

where x^*_{kj} is the j^{th} "knot" location for the k^{th} spline and summations are for $i = 1 \ldots n$.

The first 4 rows of $\mathbf{C}$ are the constraints required by the cubic spline bases being used for each term. The final two rows of $\mathbf{C}$ impose side constraints on the smooths so that for $k = 1, 2$, $\sum_i f_k(x_{ki}) = 0$ — these ensure identifiability of the model (simply omitting the constant terms from the representation of the 2 smooths does not work well in practice).

So, given smoothing parameters, λ_1 and λ_2, which directly control the effective degrees of freedom per smooth term, this model is straightforward to fit using the same approach taken in the single penalty case in section 2.4. Furthermore, given this representation of the GAM as a constrained generalized ridge regression problem the smoothing parameters can also be estimated reasonably efficiently using GCV, as we will discuss in section 3.1. It is not difficult to generalize to more than two smooth terms, or to models involving thin plate spline terms of the sort discussed in section 3.4 (which can include terms with different orders of penalty).

The above model is really an AM rather than a GAM, but generalization is straightforward. Consider the model:

$$g(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta} \quad \text{where} \quad y_i \sim \text{exponential family},$$

$\mathbf{X}$ is the design matrix for an additive model constructed in the manner described above, and g is a smooth monotonic "link" function. In the same way as the least squares objective was penalized above, in order to ensure smooth function estimates, the log likelihood can be penalized in the this more general setting. Specifically, if $l(\boldsymbol{\beta})$ is the log likelihood for the above GAM, then we would fit the GAM by minimising the negative penalized log-likelihood:

$$-l(\boldsymbol{\beta}) + \frac{1}{2}(\lambda_1\boldsymbol{\beta}'\mathbf{H}_1\boldsymbol{\beta} + \lambda_2\boldsymbol{\beta}'\mathbf{H}_2\boldsymbol{\beta})$$

Minimisation of the negative penalized log likelihood for this model can be achieved by iteratively re-weighted least squares. Letting $V(\mu_i)$ be the variance of y_i implied by a mean μ_i and using the superscript "$^{[k]}$" to denote the estimate of a quantity at the k^{th} iteration, we can define pseudodata:

$$\mathbf{z}^{[k]} = \mathbf{X}\boldsymbol{\beta}^{[k]} + \boldsymbol{\Gamma}^{[k]}(\mathbf{y} - \boldsymbol{\mu}^{[k]})$$

where $\boldsymbol{\Gamma}^{[k]}$ is a diagonal matrix such that $\Gamma_{ii}^{[k]} = g'(\mu_i^{[k]})$. Also define a diagonal weight matrix, $\mathbf{W}$, where:

$$W_{ii} = \left[\Gamma_{ii}^{[k]}\sqrt{V(\mu_i^{[k]})}\right]^{-1}.$$

Penalized maximum likelihood estimation is performed by iterative solution of:

$$\begin{array}{rl} \text{minimise} & \|\mathbf{W}^{[k]}(\mathbf{X}\boldsymbol{\beta} - \mathbf{z}^{[k]})\|^2 + \lambda_1\boldsymbol{\beta}'\mathbf{H}_1\boldsymbol{\beta} + \lambda_2\boldsymbol{\beta}'\mathbf{H}_2\boldsymbol{\beta} \\ \text{subject to} & \mathbf{C}\boldsymbol{\beta} = \mathbf{0} \end{array}$$

for $\boldsymbol{\beta}^{[k+1]}$. In other words, given smoothing parameters, GAM fitting amounts to penalized likelihood maximization by iterative least squares (O'Sullivan 1986; Hastie and Tibshirani, 1990; Wahba, 1990 and Green and Silverman 1994 all provide further information). Smoothing parameter estimation by GCV can be included in the scheme by applying GCV estimation of $\boldsymbol{\lambda}$ to the weighted least squares problem produced at each stage of the iterative least squares method. The method for doing this is covered in the next section.

Notice that the total computational cost of estimating these models, given $\boldsymbol{\lambda}$ is modest relative to the full spline models of Wahba (1990) and co-workers. If n is the number of data modelled, and q the total number of parameters used to represent the model (that is the length of $\boldsymbol{\beta}$) then the solution of the least squares problem produced at each stage of iteration costs $O(nq^2)$ operations. For a full spline model this cost would be $O(n^3)$, but when modelling with penalized regression splines it is usual for q to be substantially less than n.

3.1 Choosing $\boldsymbol{\lambda}$, the parameter(s) controlling the amount of smoothing

The unresolved question from the previous section, is how to choose $\boldsymbol{\lambda}$? In this section we describe a computationally efficient approach using Generalized Cross Validation (GCV). We again start by discussing the simple Additive Model case and add Generalization at the end of the section.

GCV can be motivated by first considering Ordinary Cross Validation (OCV). OCV works like this: imagine leaving out one of your data points, fitting the model to the remainder and calculating the squared difference between the left out datum and the fitted model; now repeat this calculation for each data point in the data set and hence obtain the average squared difference between missing data and model fitted to the remaining data. This average squared difference is the ordinary cross validation score. Low values indicate a good model, while high values suggest a poor model. In the context of linear models and penalized linear models of the sort covered here, it is possible to perform OCV in a relatively efficient manner without having to actually re-fit the model for each left out datum. This is achieved by writing the OCV score as a weighted sum of the model residuals, where the weights are calculable directly from the original fit to all the data (see e.g. Wahba 1990, section 4.2). The GCV score is obtained by replacing all the individual weights in this summation by the average weight. This yields the score

$$V = \frac{n\|\mathbf{y} - \mathbf{A}\mathbf{y}\|^2}{[n - tr(\mathbf{A})]^2}$$

where $\mathbf{A}$ is the "hat matrix" or "influence matrix" for the model being fitted. That is the matrix such that:

$$\hat{\boldsymbol{\mu}} = \mathbf{A}\mathbf{y}$$

In the notation of sections 2.4 and/or 3 (and neglecting weights and constraints):

$$\mathbf{A} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \sum_i \lambda_i \mathbf{H}_i)^{-1}\mathbf{X}^T \tag{6}$$

The term tr($\mathbf{A}$) is the estimated degrees of freedom of the model. This is by analogy with ordinary linear regression where the the hat matrix is $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$, the trace of which is well known to be the number of (identifiable) parameters in the model. Note then that the GCV score is proportional to the estimated variance of the data per estimated residual degree of freedom. V is a function of the smoothing parameters, so the idea is that the "best" $\boldsymbol{\lambda}$ vector will be the one minimising V

There are two reasons for working with the GCV score, rather than with the OCV score. One reason is computational: in the multiple smoothing parameter context it is possible to produce very efficient methods based on GCV, but none are yet known for OCV, and the task of producing one appears formidable. The second reason is theoretical. The model fitting problems for GAMs constructed using penalized regression splines are examples of generalized ridge regression problems. However there are generalized ridge regression problems in which the behaviour of the OCV score is different when the problem is written down using different, but exactly equivalent, bases (see e.g. Wahba 1990, section 4.3). Golub, Heath and Wahba (1979) suggest GCV as a suitably invariant fix for this problem. Some quite good theoretical properties have been obtained for GCV — in particular GCV estimated smoothing parameters can be shown to minimise mean square prediction error in the large sample limit (see Wahba 1990, section 4.4 and references therein).

Although, it is possible to produce efficient GCV methods, actually doing so is quite taxing. The difficulty arises because V is expensive to calculate directly: the cost of direct calculation of V is of the same order as the cost of model fitting: i.e. around $O(nq^2)$, where n is the number of data and q the total number of parameters. Hence attempting to find the smoothing parameters by direct grid search rapidly becomes very costly as the number of smoothing parameters increases. This is because each new trial set of smoothing parameters will require an $O(nq^2)$ calculation to obtain the GCV score (although in the AM setting this can be reduced to $O(q^3)$).

If there is only one smoothing parameter it is possible to perform some transformations of the problem up front so that subsequent evaluations of V for different λ values are very efficient (see Wahba, 1990 for some strategies), but unfortunately this is not possible with more than one smoothing parameter. In the multiple smoothing parameter context the most efficient strategy known to date is as follows.

Firstly, the multiple smoothing parameter model fitting problem is re-written with an extra (and strictly redundant) "overall" smoothing parameter controlling the tradeoff between model fit and overall smoothness, while retaining smoothing parameters multiplying each individual penalty which now control only the *relative* weights given to the different penalties. The following steps are then iterated:

- Given the current estimates of the relative smoothing parameters, estimate the overall smoothing parameter using highly efficient single smoothing parameter methods.
- Given the overall smoothing parameter, update the logarithms of the relative smoothing parameters simultaneously using Newton's method (backed up by steepest descent).

Working with the logs of smoothing parameters in the second step avoids the need to use constrained optimization to force the relative smoothing parameters to remain positive. It also allows meaningful step length limits to be set in the second step. This approach was first proposed by Gu and Wahba (1991) for full spline smoothing models, in which context it is of $O(n^3)$ computational cost (n being the number of data). Their method used special structure of the full spline smoothing problem and can not be used directly with the penalized regression spline based models described in this paper. Wood (2000) provides the generalization of the Gu and Wahba method to more general problem classes, including GAMs built from penalized regression splines, and we refer the reader to that paper for full details of the method.

The Wood (2000) approach typically converges in 5-15 iterations, with each iteration being $O(nq^2)$ in computational cost. This can be compared, for example, to the smoothing parameter selection strategy proposed by Marx and Eilers (1998) when using P-spline based GAMs — smoothing parameter selection for their 3 term GAM example required 9^3 model fitting steps each costing $O(nq^2)$ operations. So, even with small numbers of smooth terms in the model, our suggested approach is likely to offer quite substantial computational savings relative to direct grid-search based methods. It is also worth comparing the computational cost of our approach with the cost of using GCV to estimate smoothing parameters in a backfit GAM context. In this case the computational cost is higher again, because it is expensive to obtain the $\text{tr}(\mathbf{A})$ term required to evaluate V — using backfitting this term is $O(n^3)$ to calculate, and the term has to be re-calculated for each trial set of smoothing parameters: this is the motivation for approximate GCV methods like BRUTO (see Hastie and Tibshirani, 1990). Again this comparison suggests that our approach offers quite substantial computational savings.

3.1.1 Generalization

GCV can be used with weights, and as part of the iteratively re-weighted least squares methods used to fit GAMs by penalized likelihood maximization. Suppose that the model fit term in the fitting objective is $\|\mathbf{W}(\mathbf{y}-\boldsymbol{\mu})\|^2$, then the GCV score is:

$$V = \frac{n\|\mathbf{W}(\mathbf{y}-\mathbf{A}\mathbf{y})\|^2}{[n - tr(\mathbf{A})]^2}$$

where:

$$\mathbf{A} = \mathbf{X}(\mathbf{X}^T\mathbf{W}^2\mathbf{X} + \sum_k \lambda_k \mathbf{H}_k)^{-1}\mathbf{X}^T\mathbf{W}^2$$

(note that non diagonal $\mathbf{W}$ would cause no technical or practical difficulties here).

Use with the iteratively re-weighted least squares method used to fit GAMs is equally straightforward: the same GCV score as above is used, but at each step the pseudodata $\mathbf{z}$ replaces $\mathbf{y}$ and the weights are given by the iterative weights (see e.g. Gu and Wahba 1993; Wood 2000).

3.2 Confidence intervals on the model, and term-wise effective degrees of freedom

There are a number of ways of calculating confidence bands for the terms making up a GAM. An approach that gives good coverage probabilities (Lonegan in prep) is to use intervals similar to the Bayesian intervals developed by Wahba (1983). They can be justified by making a large sample approximation about the pseudodata at convergence: $\mathbf{z}|\boldsymbol{\beta} \sim N(\mathbf{X}\boldsymbol{\beta}, \mathbf{I}\sigma^2)$ and by assuming a multivariate normal prior on $\boldsymbol{\beta}$ with mean $\mathbf{0}$ and covariance matrix proportional to an appropriate pseudoinverse of $\sum_k \lambda_k \mathbf{H}_k$: see Hastie and Tibshirani, (1990) section 3.6, based on Silverman (1985) (they assume square $\mathbf{X}$, but this is not necessary). Neglecting constraints and assuming uniform weights, the estimated posterior covariance matrix for the parameters is given by $\mathbf{V}_{\hat{\beta}} = \hat{\sigma}^2(\mathbf{X}^T\mathbf{X} + \sum_k \hat{\lambda}_k \mathbf{H}_k)^{-1}$, where $\hat{\sigma}^2 = \|\mathbf{y} - \hat{\boldsymbol{\mu}}\|^2/[tr(\mathbf{I} - \mathbf{A})]$ and all quantities are estimated at convergence. The approximate posterior distribution of $\hat{\boldsymbol{\beta}}$ is multivariate normal, so approximate confidence intervals for the parameters can be obtained. The generalization to the non-uniform weight case useful for GAMs (rather than AMs) is straightforward.

Given an estimate of the parameter covariance matrix it is possible to obtain the variance associated with any smooth term in the model evaluated at any point, since the smooths are linear in the model parameters. As an example consider the simple two term GAM given at the beginning of section 3:

$$\text{var}(\hat{f}_1(x)) = \sum_{i=1}^{q_1}\sum_{j=1}^{q_1} b_{1i}(x) b_{1j}(x) V_{\hat{\beta},i,j}$$

Also by linearity of the smooths in the parameters we expect that any smooth evaluated at any particular covariate value(s) will have an approximately normal distribution - hence confidence intervals on model terms can be obtained.

Further outputs usually needed for a fitted model are the estimated degrees of freedom for each model term. These are obtained from the model “hat” or “influence” matrix defined in equation (6) (again neglecting constraints and weights). Tr($\mathbf{A}$) gives the estimated degrees of freedom for the whole model. To obtain separate e.d.f.s for each smooth model term we must decompose the elements on the leading diagonal of $\mathbf{A}$ into components relating to the different terms within the model. To do this note that the matrix:

$$\mathbf{P} = (\mathbf{X}^T\mathbf{X} + \sum_k \hat{\lambda}_k \mathbf{H}_k)^{-1}\mathbf{X}^T$$

yields the parameter estimates when applied to the (pseudo)data. Hence each row of $\mathbf{P}$ is associated with one parameter, and $\mathbf{A} = \mathbf{XP}$. Now let $\mathbf{P}_k$ be the matrix $\mathbf{P}$ with all rows zeroed except for those associated with the parameters of the k^{th} smooth. Clearly now, $\mathbf{A} = \sum_k \mathbf{XP}_k$. This gives a straightforward way of evaluating the effective degrees of freedom of the k^{th} smooth: it is simply $\text{tr}(\mathbf{XP}_k)$.

3.3 When to drop terms

We have seen how the smoothness of model terms may be estimated given that the terms are included in the model, but have not yet considered how to judge whether a term should be in the model at all. The need to answer this question separately arises because the automatic smoothing parameter selection method can not reduce the degrees of freedom of a term all the way to zero. Once a term has become a straight line or plane (or some other simple polynomial if higher order penalties are used) it has zero wiggliness, so that it is not possible to simplify the model beyond this point by further smoothing parameter changes. As a result the decision to remove a term from the model altogether has to be made in a different way.

The approach that is most consistent with using GCV for smoothing parameter selection is to drop each term from the model in turn, and see if this reduces the GCV score relative to the full model. This approach could be used as the basis for a general backwards selection method.

In practical modelling situations an ad hoc approach is sometimes useful. For this, three questions need to be asked:

(1) Are the estimated degrees of freedom for the term close to their lower limit (e.g. 1 for a univariate smooth with a second derivative based wiggliness penalty)?
(2) Does the confidence region for the smooth everywhere include zero?
(3) Does the GCV score for the model go down if the term is removed from the model?

If the answer to all 3 of these is “yes” then the term should be dropped. If the answer to 2 is “no” then it probably should not be. Other cases will require judgement.

As an example here is some output from the `R` package `mgcv`, which the first author has written to implement the approach to GAMs described in this paper (see `http://cran.r-project.org/`, and Wood 2001). In this case a 4 term model has been fitted to data simulated from a 3 term truth (normal errors, identity link) so that the final covariate is spurious. The results of estimating the model with smoothing parameters selected by GCV is shown in the plot.

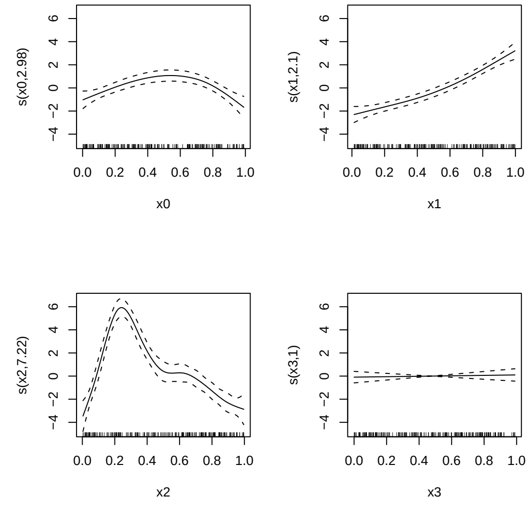
```
> gam.model<-gam(y~s(x0)+s(x1)+s(x2)+s(x3))
> gam.model           # printing fitted model object
Family: gaussian
Link function: identity

Formula:
y ~ s(x0) + s(x1) + s(x2) + s(x3)

Estimated degrees of freedom:
 2.982494 2.096610 7.219753 1.000005
 total =  14.29886

GCV score:  4.326104

> plot(gam.model,pages=1)
```



In this case the smooth of `x3` is a clear candidate to be dropped, and doing so also reduced the GCV score slightly.

3.4 Multi-dimensional smoothing with thin-plate spline like terms

So far we have only covered one dimensional basis functions. This section will look at multidimensional basis functions: in particular the thin-plate splines. Consider the problem of estimating the smooth function $f(\mathbf{x})$ where $\mathbf{x}$ is a d

- vector, from n observations $(y_i, \mathbf{x}_i)$ such that:

$$y_i = f(\mathbf{x}_i) + \epsilon_i$$

where ϵ_i is a random error term. f can be represented using a basis based on the thin plate splines which are the natural generalisation of cubic splines to any number of dimensions and almost any order of wiggliness penalty. If we now choose a set of points spread "nicely" over the region covered by the covariates: $\{\mathbf{x}_j^* : j = 1. \dots, m\}$, say , then f can be represented as:

$$f(\mathbf{x}) = \sum_{j=1}^{m} \delta_j \eta_{wd}(\|\mathbf{x} - \mathbf{x}_j^*\|) + \sum_{j=1}^{M} \alpha_j \phi_j(\mathbf{x}) \tag{7}$$

$\boldsymbol{\delta}$ and $\boldsymbol{\alpha}$ are unknown parameters subject to the constraint that $\mathbf{T}^T\boldsymbol{\delta} = \mathbf{0}$, where the $m \times M$ matrix $\mathbf{T}$ has elements $T_{ij} = \phi_j(\mathbf{x}_i^*)$. The $M = \begin{pmatrix} w+d-1 \\ d \end{pmatrix}$ functions ϕ_i are linearly independent polynomials spanning the space of polynomials in $\Re^d$ of degree less than w. w is the order of the derivatives in the measure that will be used to define "wiggliness" of f and must satisfy $2w > d$. The basis functions η are defined as follows:

$$\eta_{wd}(r) = \begin{cases} \frac{(-1)^{w+1+d/2}}{2^{2w-1}\pi^{d/2}(w-1)!(w-d/2)!} r^{2w-d} \log(r) & d \text{ even} \\ \\ \frac{\Gamma(d/2-w)}{2^{2w}\pi^{d/2}(w-1)!} r^{2w-d} & d \text{ odd} \end{cases}$$

The natural measure of wiggliness to use with this basis is:

$$J_{wd}(f) = \underset{\Re^d}{\int \dots \int} \sum_{\nu_1+\dots+\nu_d=w} \frac{w!}{\nu_1! \dots \nu_d!} \left(\frac{\partial^w f}{\partial x_1^{\nu_1} \dots \partial x_d^{\nu_d}} \right)^2 dx_1 \dots dx_d$$

and the way to estimate f is to find $\boldsymbol{\alpha}$ and $\boldsymbol{\delta}$ to minimise:

$$\|\mathbf{y} - \mathbf{f}\|^2 + \lambda J_{md}(f)$$

where $\mathbf{y}$ is the vector of y_i data, $\mathbf{f} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_n))^T$ and λ is a smoothing parameter. Subject to the constraint $\mathbf{T}^T\boldsymbol{\delta} = \mathbf{0}$ it turns out that:

$$J_{wd} = \boldsymbol{\delta}^T\mathbf{E}\boldsymbol{\delta}$$

where $E_{ij} = \eta_{wd}(\|\mathbf{x}_i^* - \mathbf{x}_j^*\|)$. Given the basis, it is also easy to get a design matrix for the problem of fitting f, which makes it straightforward to write the fitting problem in the same form as the regression spline problems that we have already met. Similarly, it is straightforward to incorporate these thin plate spline like terms into penalized regression spline based GAMs, and to estimate their smoothing parameters along side the the other smoothing parameters as part of model fitting. Note that full thin plate splines in the GSS sense have $\mathbf{x}_i^* = \mathbf{x}_i$ for $i = 1, \ldots n$, i.e. one η for each data point. More complete treatments of thin plate splines can be found in Duchon (1977), Wahba (1990) and Green and Silverman (1994), with the latter being the most accessible.

3.4.1 Thin plate spline for 2 covariates

To make the rather general formulae given above less intimidating, consider the simple example of 2 covariates $\mathbf{x}_1$ and $\mathbf{x}_2$, so that $d = 2$, and again let $m = 2$. Plugging these constants into the general form for the penalty yields:

$$J = \int\int \left(\frac{\partial^2 g}{\partial x_1^2}\right)^2 + 2\left(\frac{\partial^2 g}{\partial x_1 \partial x_2}\right)^2 + \left(\frac{\partial^2 g}{\partial x_2^2}\right)^2 dx_1 dx_2$$

There are now $M = 3$, ϕ_i: $\phi_1(x_1, x_2) = 1$, $\phi_2(x_1, x_2) = x_1$ and $\phi_3(x_1, x_2) = x_2$, So that the constraint matrix becomes:

$$\mathbf{T}^T = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ x_{11}^* & x_{12}^* & x_{13}^* & \ldots & x_{1m}^* \\ x_{21}^* & x_{22}^* & x_{23}^* & \ldots & x_{2m}^* \end{bmatrix}$$

and in this case:

$$\eta(r) = \frac{1}{8\pi} r^2 \log(r)$$

3.4.2 Problems with multi-dimensional terms

There are two problems with the approach taken in this section. The first is the way in which the basis is constructed, by selecting a “nicely” distributed set of points $\{\mathbf{x}_j^* : j = 1, \ldots, m\}$. It can be quite difficult to choose points that adequately cover the region covered by the covariates, particularly as d increases and if the data are irregularly spread over an irregularly shaped region. It would be better to use a basis that avoids this knot placement problem. Wood (2002) suggests an optimal basis that does just that.

The second problem is that the thin plate splines are isotropic smoothers. In some cases this is sensible, but in others the relative scaling of different covariates is arbitrary, but will still affect how the wiggliness measure penalizes wiggliness in different directions. Ideally in these cases one would estimate one smoothing parameter for each covariate: i.e. one overall smoothing parameter

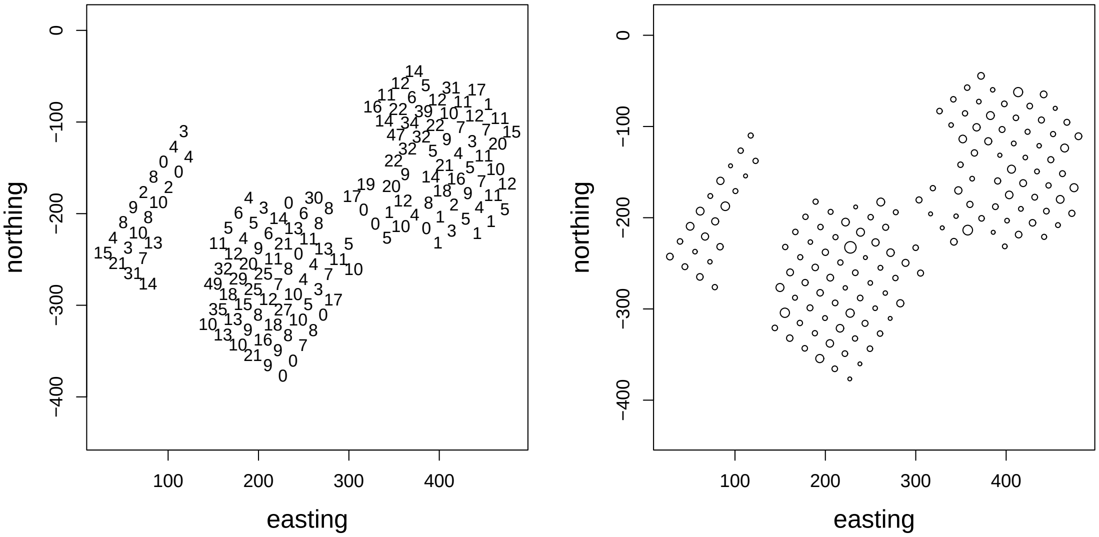


Fig. 1. The left hand plot shows the raw percentage infection rates of beech trees. The right hand plot shows the corresponding residuals for the model fit discussed in the text. Circle sizes are proportional to magnitude of residuals.

and a set of relative scaling parameters. Wood (2000) shows how this can be done, but practical implementation is still at an early stage of development.

4 Modelling beech canker

In this section we analyse data from a trial which was part of an international beech provenance trial of the Federal Research Centre for Forestry and Forest Products, Grosshansdorf (Muhs, 1991), and from an additional investigation of shelter wood in its vicinity (Metzler et al, in press). The trial is situated in the forest district Bad Wildbad in the northern Black Forest on a plateau with shallow soil on sandstone. In its vicinity are a few old shelter beech trees infected with *Nectria* canker. Other tree stands nearby are dominated by Norway spruce.

Between 1987 and 1988 100 young beech trees were planted on each of 149 plots of size 10 x 10 m. After the first year on average 61% of the trees survived. In November 2000 each tree of the 149 plots was examined for *Nectria* canker. The percentage infection rates per plot are shown in the left panel of

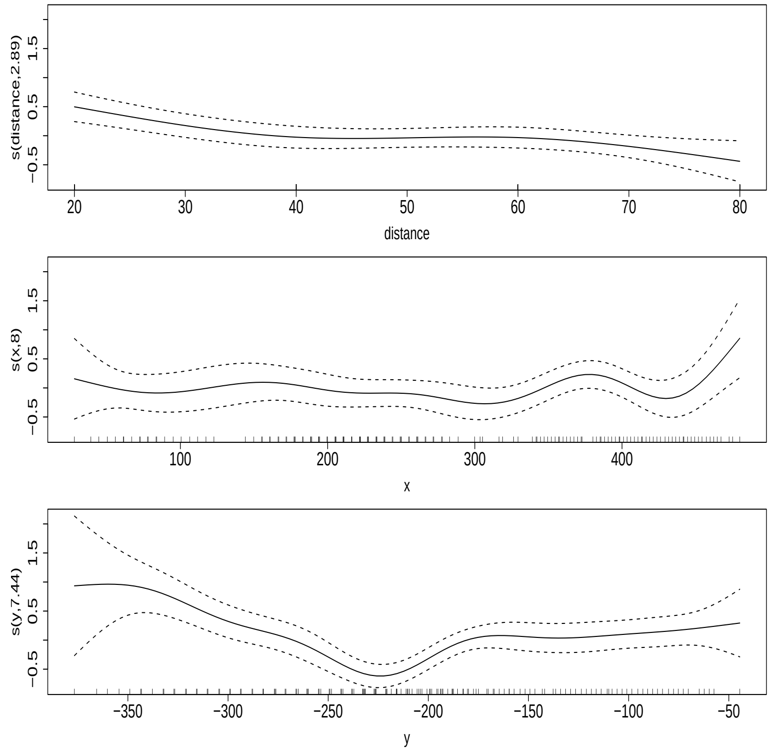


Fig. 2. Estimated terms describing the dependence of number of infected trees per plot on distance, x and y. Estimates (solid) and 95% Bayesian confidence intervals (dashed), with covariate values as a rug plot along the bottom of the plot are shown.

Figure 1. To investigate the spread of the disease from the infected shelter wood equidistance lines ($dist$) were drawn around the diseased shelterwood with radius groups 20, 40, 60 and 80m. In addition wind dispersal zones (wdz) were calculated according to the pattern of a wind rose of the prevailing wind direction, as outlined in Metzler et al (2002). There are four wind dispersal zones of which zone 1 is the nearest to the infected shelterwood. For investigating the effect of the wind dispersal zones we fit a logistic model to the probability of infection on plot i.

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + wdz_i + s_1(dist_i) + s_2(x_i) + s_3(y_i)$$

where β_0 is the intercept and wdz is a factor.

Using the `gam()` function from the `mgcv` package to fit this model yields:

```
> fit<- gam(ratio~as.factor(wdz)+s(distance,4)+s(x)+s(y),
           family=binomial, weights=buche$N,data=buche)
> fit
Family: binomial
Link function: logit
```

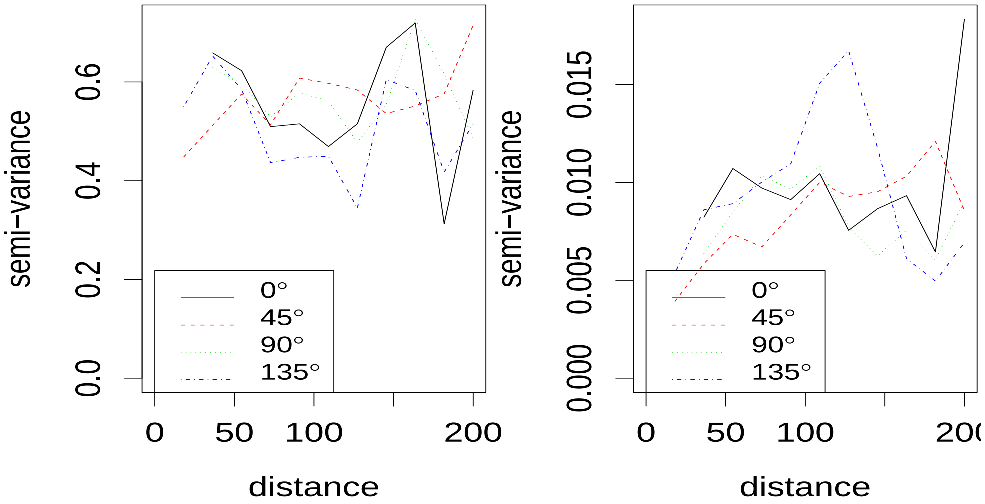


Fig. 3. The left hand plot shows the directional semi-variogram of residuals from the gam fitted to number of infected trees, while the right hand plot is the equivalent on the raw percentage of infected trees

```
Formula:
ratio ~ as.factor(wdz) + s(distance, 4) + s(x) + s(y)

Estimated degrees of freedom:
 2.893629 7.99685 7.435384   total =  22.32586

GCV score:  1.177649
```

Figure 2 shows the fitted smooth functions and Bayesian confidence intervals. It appears that the x coordinate does not have a “significant” effect as the confidence interval includes zero at most values of x. Dropping x also yields a lower GCV score:

```
Family: binomial
Link function: logit

Formula:
ratio ~ as.factor(wdz) + s(distance, 4) + s(y)

Estimated degrees of freedom:
 2.083032 7.75965   total =  13.84268

GCV score:  1.103182
```

We investigate the model fit using the semi-variogram function provided in the R package geoR (Ribiero & Diggle, 2000). The right hand panel of Figure 1 shows the residuals in space. The left panel of figure 3 shows that in all

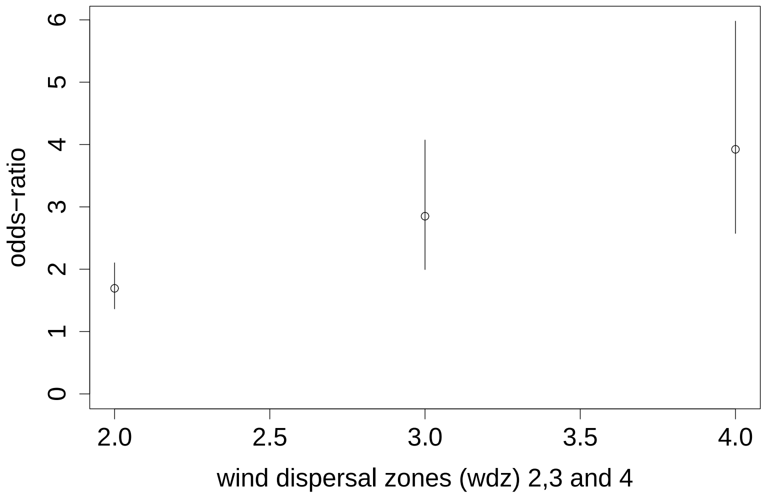


Fig. 4. Odds-ratios: odds of infection among trees in wind dispersal zone 1 ($wdz1$) divided by odds of infection among trees in $wdz2$ to $wdz4$ (○) with 95% confidence limits.

directions the empirical semi-variogram is very close to a horizontal line up to a distance of 200m. Thus we have suceeded fairly well in eliminating spatial autocorrelation compared to the autocorrelation and trend present in the response variable (right hand panel of figure 3), where the semi-variograms at 45° and 135° have a steep slope.

Using the estimated coefficients for wind dispersal zones we calculated odds-ratios as shown with 95% confidence intervals in Figure 4. This shows that for $wdz1$ there is a higher chance of infection than in the other zones: The odds of infection in $wdz1$ is about 1.7 times higher than in $wdz2$, 2.9 times higher than in $wdz3$ and 3.9 times higher than in $wdz4$. The difference in odds of the last is significantly different to the odds of $wdz1$ vs $wdz2$.

5 Modelling Fish Eggs

As a second (and more concise) example we consider a Fisheries problem.

The western stock of Atlantic mackerel (*Scomer scombrus*) is an important fishery resource in European waters and the actual biomass of the stock is estimated every three years for management purposes. The Daily Egg Production Method (DEPM) is one possible method to estimate the biomass. It estimates the total fish biomass indirectly from an estimate of the peak daily egg production using egg plankton survey data from the middle of the spawning season. For the purpose of this example we concentrate on modelling egg production, but for further details on the DEPM see Gunderson (1993).

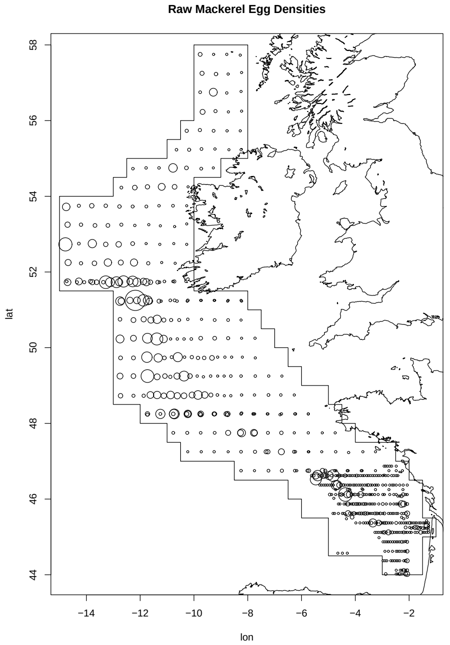


Fig. 5. Sample locations of mackerel egg density data, with symbol sizes proportional to the raw density estimate at the sampling location. The piecewise rectangular boundary is the boundary of the survey area.

When modelling egg production or density the random variable is the observed egg count, obtained by net hauls taken from below the lower depth limit for mackerel eggs (or the sea bed if higher) to the sea surface. In areas of high density only subsamples of eggs are counted. Egg counts are multiplied by a conversion factor involving a multiplication factor for the subsample, sampled water volume, egg mortality, the depth of the water column sampled etc. Although we are interested in egg density, egg count would seem natural to use as the response variable in the model. Nevertheless we prefer to model the data at the density level, due to the fact that we are dealing with subsamples the size of which depend on density.

Hence the response in the fitted model is mackerel egg density per m^2 of sea surface per day (see Figure 5). Possible candidate covariates for modelling the density are longitude, latitude, sea bed depth, sea surface temperature and the distance from the 200m sea bed contour.

Borchers *et al* (1997) modelled these data using backfit GAMs constructed using univariate smoothing splines under the assumption that the effects of the different explanatory variables are additive. The approach taken by Borchers *et al* for model selection was to consider only splines with either 4 degrees of freedom (*df=4*) or one degree of freedom (*df=1*) for covariates and their first order interactions (defined as a the product of the covariates). The covariates first entered the model with *df=4*, and backward stepwise elimination was used

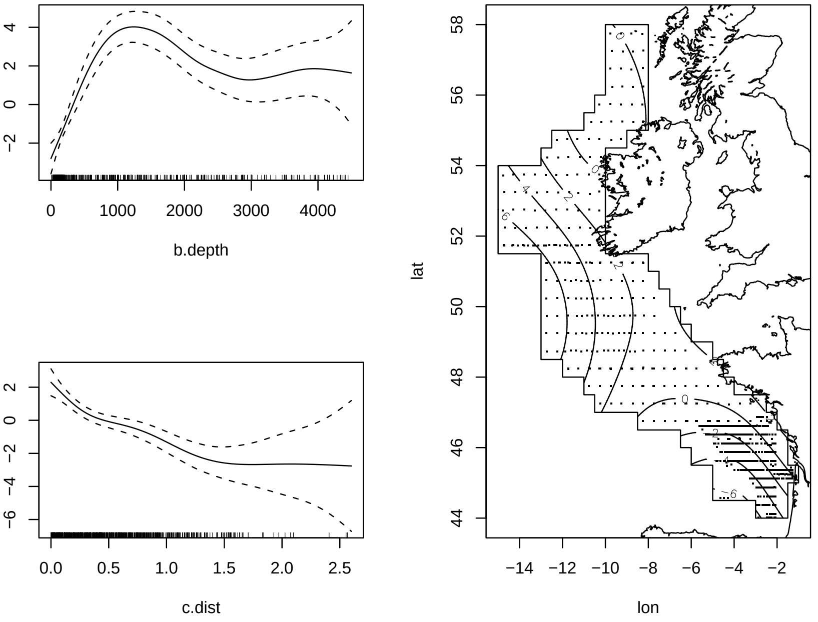


Fig. 6. Estimated terms describing the dependence of mackerel egg densities on longitude, latitude, sea bed depth and distance from the 200m sea bed contour. For both univariate smooths the plot shows estimates (solid) and 95% Bayesian confidence intervals (dashed), with covariate values as a rug plot along the bottom of the plot. The bivariate smooth is shown as a contour plot over the survey area with part of the European coastline superimposed for orientation and sample locations marked.

to select a set of covariates. Selection between smooths with *df=4* and smooths with *df=1* was performed in the next step. Finally, first order interactions of the previously selected covariates were first entered with *df=4*, again using backwards stepwise elimination for model selection, and selection between smooths with *df=4* and *df=1* was performed in the next step. Comparisons between models were made on the basis of approximate F-tests (Hastie and Tibshirani, 1990).

Besides the fact that the above procedure is *ad hoc*, the assumption of having additive latitude and longitude effects may not be very realistic. *A priori* it is odd to model the dependence on longitude and latitude by summing a longitude and a latitude effect: Mackerel are unlikely to know which coordinate system we happen to have chosen. Hence we modelled the dependence on spatial location as an isotropic bivariate function of longitude and latitude. We use GCV for controlling the amount of smoothing as describe above. The model was fitted using the `gam()` function in R package `mgcv`, with the call:

```
> mack.fit<-gam(egg.dens~s(lon,lat,40)+s(b.depth)+s(c.dist)+s(temp.surf))
```

Inspection of residual plots clearly indicated a mean variance problem, which can be largely eliminated by raising the egg density data to the power 0.4.

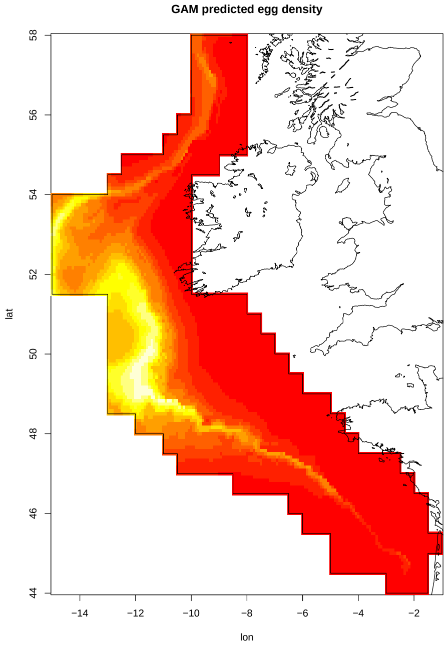


Fig. 7. Image plot of the GAM egg densities across the survey area for the egg data presented in figure 1.

Following the term selection procedure given in section 3.3 it was apparent that sea surface temperature should be removed from the model. Doing this and refitting:

```
> mack.fit<-gam(egg.dens^0.4~s(lon,lat,40)+s(b.depth)+s(c.dist))
```

yielded the model term estimates shown in figure 2 and the fitted model egg densities shown in figure 3. This model yields a GCV score of 3.6. For comparison we also fitted an additive model with the same terms as in Borchers et al (1997), but allowing mgcv to estimate the smoothing parameters

```
> mack.fitadd<-gam(egg.dens^0.4~s(lon)+s(lat)+s(b.depth)+s(c.dist))
```

yields a GCV score of 3.75. Going one step back and fixing the degrees of freedom to 4 increases the GCV score to 3.85.

6 Discussion

In this paper we have provided an introduction to the theory required for estimation, model selection and inference with GAMs constructed using penalized regression splines, and illustrated this theory with two ecological examples. The given theory provides the necessary background to understand the GAM modelling software provided with the R language and environment in package mgcv. But note that mgcv actually uses a different (although exactly equivalent) basis representation to the one given here, and also allows the use of the more sophisticated bases covered in Wood (2002).

As compared to GAMs as implemented in Splus the approach given here has advantages, and disadvantages. The advantages are that model selection is somewhat more convenient and a little less ad hoc, and that inference requires fewer approximations. If the user is happy to use GCV for model selection than the ability to select models automatically and rapidly using this criterion is an advantage, especially in an interactive modelling context, and this might be viewed as an improvement on the backfit GAMs. On the other hand, the disadvantage of the approach is that the class of smoothers usable with the methods is much smaller. For example the approach given here can not be used with LOESS. On the other hand the methods discussed in this paper can be employed with any smoothers that can be represented by a set of basis functions and a wiggliness penalty (e.g. the P-splines of Eilers and Marx), so that a very rich family of models could be produced by employing these methods in conjunction with the "pseudosplines" of Hastie (1996).

As compared with the generalised smoothing spline methods of Wahba and co-workers (as implemented, for example, in the `R` package `gss`), the methods presented here have one major advantage: computational efficiency. On the other hand they suffer the disadvantage of only offering approximations to the GSS models. Relative to the approach to GAMs described in Marx and Eilers (1998) the methods described here have the advantage of increased model selection efficiency, the ability to incorporate multidimensional smooths in a straightforward way and the fact that our penalties are perhaps a little easier to interpret. The relative disadvantages of the approach given here are that the enhanced computational efficiency is not easily extended to other model selection criteria and that the methods are more difficult to implement, although this latter issue is unlikely to be of concern to users.

Perhaps the most interesting open issue for the models discussed here relates to multi-dimensional smooths. Specifically how should the relative scaling of covariates be performed? This is the problem of anisotropic smoothing: for example if we would like to model some response variable as a smooth function of distance along a transect, d and time t, then we need to decide how to scale distance against time, in order that the smoothness of the function is appropriate in time and space. In principle this is a non-linear multiple smoothing parameter estimation problem that can be approached using methods discussed in Wood (2000), but practical implementation of such an approach within a GAM framework is some way off yet.

Acknowledgements

The authors thank E. Kublin, E. Meierjohann and B. Metzler for making the beech canker data available.

References

[1] Borchers, D.L., Buckland, S.T., Priede, I.G. and Ahmadi, S. (1997) Improving the precision of the daily egg production method using generalized additive models. *Can. J. Fish. Aq. Sci* **54**, 2727-2742.

[2] DeBoor, C. (1978) A Practical Guide to Splines. Springer-Verlag. New York.

[3] Gunderson, D. R. (1993) Surveys of Fisheries Resources. John Wiley & Sons, Inc., London.

[4] Duchon, J. (1977) Splines minimizing rotation-invariant semi-norms in Solobev spaces *in Construction Theory of Functions of Several Variables* Springer, Berlin.

[5] Eilers, P.H.C. and Marx, B.D. (1996) Flexible Smoothing with B-splines and Penalties *Statistical Science* **11**, 89-121.

[6] Green, P.J. and Silverman, B.W. (1994) *Nonparametric regression and generalized linear models.* London: Chapman and Hall.

[7] Gu, C and Wahba, G. (1991) Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method. *SIAM J. Sci. Stat. Comp.*, **12(2)**, 383-398.

[8] Gu, C and Wahba, G. (1993) Semiparametric analysis of variance with tensor product thin plate splines. *J. R. Statist. Soc. B* **55(2)**, 353-368.

[9] Hastie. T.J. and Tibshirani, R.J. (1990) *Generalized additive models.* London: Chapman and Hall.

[10] Hastie. T.J. (1996) *Pseudosplines. J. R. Statist. Soc. B* **58(2)**, 379-396.

[11] Lonergan, M.E. (in prep.) Bayesian confidence interval coverage probabilities for cross validated GAMs built from penalized regression splines: a simulation study.

[12] Marx B.D. and P.H.C. Eilers (1998) Direct generalized additive modeling with penalized likelihood. Computational Statistics and Data Analysis.

[13] Metzler, B., Meierjohann, E., Kublin, E. and von Wühlisch, G. (2002). Spatial dispersal of *Nectria ditissima* canker of beech in an international provenance trial. Forest Pathology 32:1–8.

[14] Muhs, H. J. (1991) Die Anlage des interantionalen Bucheherkunftsversuchs 1983 - 1985. Eds Korpel, S. and Paule, L. Proceedings of the 3. IUFRO-Buchensymposium in Zvolen 1988: 85-89. Zvolen: Publisher.

[15] Ribeiro Jr, J. and Diggle, P.J. (2000) geoR: A package for geostatistical analysis. R News 1(2): 15-18.

[16] Silverman, B.W. (1985) Some aspects of the spline smoothing approach to nonparametric regression curve fitting. *J.R. Statist. Soc. B* **47**,1-52.

[17] O'Sullivan, F. Yandell, B.S. and Raynor, W.J. (1986) Automatic smoothing of regression functions in generalized linear models *J. Am. Statist. Ass.* **81**, 96-103.

[18] Wahba, G, (1980) Spline bases, regularization, and generalized cross validation for solving approximation problems with large quantities of noisy data. In *Approximation Theory III* W. Cheney, ed., Academic Press, New York, pp.905-912

[19] Wahba, G. (1983) Bayesian confidence intervals for the cross validated smoothing spline. *J. R. Statist. Soc. B* **45**, 133-150.

[20] Wahba (1990) *Spline models for observational data. CBMS-NSF Reg. Conf. Ser. Appl. Math.*: **59**.

[21] Watkins, D.S. (1991) Fundamentals of Matrix Computation. . *John Wiley and Sons, New York*

[22] Wood, S.N. (2000) Modelling and smoothing parameter estimation with multiple quadratic penalties. *J. R. Statist. Soc. B* **62**, 413-428.

[23] Wood, S.N. (2001) mgcv: GAMs and Generalized Ridge Regression for R. R News 1(2):20-25.

[24] Wood, S.N. (2002) Thin-plate regression splines. submitted *J. R. Statist. Soc. B*