CHAPTER 4

Some GAM theory

In the last chapter, it was demonstrated how the problem of estimating a generalized additive model, becomes the problem of estimating smoothing parameters and model coefficients for a penalized likelihood maximization problem, once a basis for the smooth functions has been chosen, together with associated measures of function wiggliness. In practice the penalized likelihood maximization problem is solved by penalized iteratively re-weighted least squares (P-IRLS), while the smoothing parameters can be estimated using cross validation or related criteria. The purpose of this chapter is to justify and extend the methods introduced in chapter 3, and to add some distribution theory to facilitate confidence interval calculation and hypothesis testing. Table 4.1 lists the main elements of the approach, and where they can be found within the chapter.

The methods discussed in this chapter are almost all built around penalized regression smoothers, based on splines. This type of smoother goes back at least as far as Wahba (1980) and Parker and Rice (1985). The suggestion of representing GAMs using spline like penalized regression smoothers was made in section 9.3.6 of Hastie and Tibshirani (1990) and was given renewed impetus by Marx and Eilers (1998), but it is not the only possibility, as will briefly be covered at the chapter's end.

The chapter starts by introducing several different penalized regression smoothers useful for practical work, including smooth functions of several covariates. Since these are all spline based, some discussion of why splines are useful smoothers is also presented. There follows a short explanation of how these can be assembled into an estimable GAM, and the P-IRLS estimation scheme is then justified, before moving on to the important topic of smoothing parameter estimation. Having covered model representation and estimation, a Bayesian model useful for deriving confidence intervals is then introduced, before considering practical performance of such intervals, and the calculation of approximate p-values for model terms. Some further topics of theoretical interest are then touched on before finishing with a very brief presentation of some key ideas underpinning two alternative frameworks for GAM estimation and inference. A review of the matrix algebra used in this chapter is provided in Appendix A.

What	How	Where
Turn GAM into penalized GLM with coefficients $\boldsymbol{\beta}$ and smoothing parameters $\boldsymbol{\lambda}$	Choose bases and wiggliness measures for the smooth terms	4.1, 4.2
Select $\boldsymbol{\lambda}$	By GCV, UBRE or AIC	4.5
	using efficient, robust Newton methods	4.6, 4.7
Estimate $\boldsymbol{\beta}$	By P-IRLS	4.3
Find confidence intervals/ credible intervals for (functions of) $\boldsymbol{\beta}$	Use Bayesian smoothing model	4.8, 4.9
Test hypotheses about GAMs	Use frequentist approximations, or GLM methods on unpenalized GAM	4.8.5, 4.10.1

Table 4.1 *The main components of the framework for generalized additive modelling covered in this chapter, and where they can be found.*

4.1 Smoothing bases

For simplicity of presentation, only one very simple type of penalized regression smoother was presented in Chapter 3. For practical work a variety of alternative smoothers are available, and this section introduces a useful subset of the possibilities, starting with smooths of one covariate, and then moving on to smooths of one or more covariates. Since all the smooths presented are based on splines (although the tensor product smooths need not be), the section starts by addressing the question: what's so special about splines?

4.1.1 Why splines?

Almost all the smooths considered in this book are based in some way on splines, so it is worth spending a little time on the theoretical properties that make these functions so appealing for penalized regression. Rather than attempt full generality, the flavour of the theoretical ideas can be gleaned by considering some properties of cubic splines, first in the context of interpolation, and then of smoothing.

Natural cubic splines are smoothest interpolators

Consider a set of points $\{x_i, y_i : i = 1, \ldots, n\}$ where $x_i < x_{i+1}$. The *natural cubic spline*, $g(x)$, interpolating these points, is a function made up of sections of cubic polynomial, one for each $[x_i, x_{i+1}]$, which are joined together so that the whole

spline is continuous to second derivative, while $g(x_i) = y_i$ and $g''(x_1) = g''(x_n) = 0$. Figure 3.3 illustrates such a cubic spline.

Of all functions that are continuous on $[x_1, x_n]$, have absolutely continuous first derivatives and interpolate $\{x_i, y_i\}$, $g(x)$ is the one that is is smoothest in the sense of minimizing:

$$J(f) = \int_{x_1}^{x_n} f''(x)^2 dx.$$

Green and Silverman (1994) provide a neat proof of this, based on the original work of Schoenberg (1964). Let $f(x)$ be an interpolant of $\{x_i, y_i\}$, other than $g(x)$, and let $h(x) = f(x) - g(x)$. We seek an expression for $J(f)$ in terms of $J(g)$.

$$\begin{aligned}\int_{x_1}^{x_n} f''(x)^2 dx &= \int_{x_1}^{x_n} \{g''(x) + h''(x)\}^2 dx \\ &= \int_{x_1}^{x_n} g''(x)^2 dx + 2\int_{x_1}^{x_n} g''(x)h''(x)dx + \int_{x_1}^{x_n} h''(x)^2 dx\end{aligned}$$

and integrating the second term on the second line, by parts, yields

$$\begin{aligned}\int_{x_1}^{x_n} g''(x)h''(x)dx &= g''(x_n)h'(x_n) - g''(x_1)h'(x_1) - \int_{x_1}^{x_n} g'''(x)h'(x)dx \\ &= -\int_{x_1}^{x_n} g'''(x)h'(x)dx \\ &= -\sum_{i=1}^{n-1} g'''(x_i^+) \int_{x_i}^{x_{i+1}} h'(x)dx \\ &= -\sum_{i=1}^{n-1} g'''(x_i^+) \{h(x_{i+1}) - h(x_i)\} \\ &= 0,\end{aligned}$$

where equality of lines 1 and 2 follows from the fact that $g''(x_1) = g''(x_n) = 0$. Equality of lines 2 and 3 results from the fact that $g(x)$ is made up of sections of cubic polynomial, so that $g'''(x)$ is constant over any interval (x_i, x_{i+1}). The final equality to zero follows from the fact that both $f(x)$ and $g(x)$ are interpolants, and are hence equal at x_i, implying that $h(x_i) = 0$.

So we have shown that

$$\int_{x_1}^{x_n} f''(x)^2 dx = \int_{x_1}^{x_n} g''(x)^2 dx + \int_{x_1}^{x_n} h''(x)^2 dx \geq \int_{x_1}^{x_n} g''(x)^2 dx$$

with equality only if $h''(x) = 0$ for $x_1 < x < x_n$. However, $h(x_1) = h(x_n) = 0$, so in fact we have equality if and only if $h(x) = 0$ on $[x_1, x_n]$. In other words any interpolant that is not identical to $g(x)$ will have a higher integrated squared second derivative. So there is a well defined sense in which the cubic spline is the smoothest possible interpolant through any set of data.

The smoothest interpolation property is not the only good property of cubic spline

interpolants. In de Boor (1978, Chapter 5) a number of results are presented showing that cubic spline interpolation is optimal, or at least very good, in various respects. For example, if a 'complete' cubic spline, g, is used to approximate a function, $\tilde{f}$, by interpolating a set of points $\{x_i, \tilde{f}(x_i) : i = 1, \ldots, n\}$ and matching $\tilde{f}'(x_1)$ and $\tilde{f}'(x_n)$ then if $\tilde{f}(x)$ has 4 continuous derivatives:

$$\max|\tilde{f} - g| \leq \frac{5}{384}\max(x_{i+1} - x_i)^4 \max|\tilde{f}''''|,$$

and this can be shown to be the best achievable.

These properties of spline interpolants, suggest that splines ought to provide a good basis for representing smooth terms in statistical models. Whatever the true underlying smooth function is, a spline ought to be able to approximate it closely, and if we want to construct models from smooth functions of covariates, then representing those functions from smoothest approximations is intuitively appealing.

Cubic smoothing splines

In statistical work, y_i is usually measured with noise, and it is generally more useful to smooth x_i, y_i data, rather than interpolating them. To this end, rather than setting $g(x_i) = y_i$, it might be better to treat the $g(x_i)$ as n free parameters of the cubic spline, and to estimate them in order to minimize

$$\sum_{i=1}^{n}\{y_i - g(x_i)\}^2 + \lambda \int g''(x)^2 dx,$$

where λ is a tuneable parameter, used to control the relative weight to be given to the conflicting goals of matching the data and producing a smooth g. The resulting $g(x)$ is a *smoothing spline* (Reinsch, 1967). In fact, of *all functions*, f, that are continuous on $[x_1, x_n]$, and have absolutely continuous first derivatives, $g(x)$ is the function minimizing:

$$\sum_{i=1}^{n}\{y_i - f(x_i)\}^2 + \lambda \int f''(x)^2 dx. \tag{4.1}$$

The proof is easy. Suppose that some other function, $f^*(x)$, minimized (4.1). In that case we could interpolate $\{x_i, f^*(x_i)\}$ using a cubic spline, $g(x)$. Now $g(x)$ and $f^*(x)$ have the same sum of squares term in (4.1), but by the properties of interpolating splines, $g(x)$ must have the lower integrated squared second derivative. Hence $g(x)$ yields a lower (4.1) than $f^*(x)$, and a contradiction, unless $f^* = g$.

So, the cubic spline basis arises naturally from the specification of the smoothing objective (4.1), in which, what is meant by model fit is defined precisely, what is meant by smoothness is defined precisely, and the basis for representing smooth functions is not chosen in advance, but rather emerges from seeking the function minimizing (4.1).

Smoothing splines, then, seem to be somewhat ideal smoothers. The only substantial problem, is the fact that they have as many free parameters as there are data to be

smoothed. This seems wasteful, given that, in practice, λ will almost always be high enough that the resulting spline is much smoother than n degrees of freedom would suggest. Indeed, in section 4.10.4 we will see that many degrees of freedom of a spline are often suppressed completely by the penalty. For univariate smoothing with cubic splines, the large number of parameters turns out not to be problematic, but as soon as we try to deal with more covariates, the computational expense becomes severe.

An obvious compromise between retaining the good properties of splines, and computational efficiency, is to use penalized regression splines, as introduced in Chapter 3. At its simplest, this involves constructing a spline basis (and associated penalties) for a much smaller data-set than the one to be analyzed, and then using that basis (plus penalties) to model the original data set. The covariate values in the smaller data set should be arranged to nicely cover the distribution of covariate values in the original data set. This penalized regression spline idea is presented in Wahba (1980) and Parker and Rice (1985), for example. In the rest of this section, some spline based penalized regression smoothers will be presented, starting with univariate smoothers, and then moving on to smooths of several variables.

4.1.2 Cubic regression splines

The basis used in Chapter 3 was one way of defining a cubic regression spline basis, but there are other ways of defining such smoothers, which have some advantages in terms of interpretability of the parameters. One approach is to parameterize the spline in terms of its values at the knots.

Consider defining a cubic spline function, $f(x)$, with k knots, $x_1 \dots x_k$. Let $\beta_j = f(x_j)$ and $\delta_j = f''(x_j)$. Then the spline can be written as

$$f(x) = a_j^-(x)\beta_j + a_j^+(x)\beta_{j+1} + c_j^-(x)\delta_j + c_j^+(x)\delta_{j+1} \text{ if } x_j \leq x \leq x_{j+1} \quad (4.2)$$

where the basis functions a_j^-, a_j^+, c_j^- and c_j^+ are defined in table 4.2. The conditions that the spline must be continuous to second derivative, at the x_j, and should have zero second derivative at x_1 and x_k, can be shown to imply (exercise 1) that

$$\mathbf{B}\boldsymbol{\delta}^- = \mathbf{D}\boldsymbol{\beta}. \quad (4.3)$$

where $\boldsymbol{\delta}^- = (\delta_2, \dots, \delta_{k-1})^\mathsf{T}$ (since $\delta_1 = \delta_k = 0$) and $\mathbf{B}$ and $\mathbf{D}$ are defined in table 4.2.

Defining $\mathbf{F}^- = \mathbf{B}^{-1}\mathbf{D}$, and

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} \\ \mathbf{F}^- \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{0}$ is a row of zeros, we have that $\boldsymbol{\delta} = \mathbf{F}\boldsymbol{\beta}$. Hence, the spline can be re-written entirely in terms of $\boldsymbol{\beta}$ as

$$f(x) = a_j^-(x)\beta_j + a_j^+(x)\beta_{j+1} + c_j^-(x)\mathbf{F}_j\boldsymbol{\beta} + c_j^+(x)\mathbf{F}_{j+1}\boldsymbol{\beta} \text{ if } x_j \leq x \leq x_{j+1},$$

Basis functions for a cubic spline

$a_j^-(x) = (x_{j+1} - x)/h_j \qquad c_j^-(x) = [(x_{j+1} - x)^3/h_j - h_j(x_{j+1} - x)]/6$

$a_j^+(x) = (x - x_j)/h_j \qquad c_j^+(x) = [(x - x_j)^3/h_j - h_j(x - x_j)]/6$

Non-zero matrix elements — non cyclic spline

$D_{i,i} = 1/h_i$ $B_{i,i} = (h_i + h_{i+1})/3$	$D_{i,i+1} = -1/h_i - 1/h_{i+1}$	$D_{i,i+2} = 1/h_{i+1}$ $i = 1 \ldots k-2$
$B_{i,i+1} = h_{i+1}/6$	$B_{i+1,i} = h_{i+1}/6$	$i = 1 \ldots k-3$

Non-zero matrix elements — cyclic spline

$\tilde{B}_{i-1,i} = \tilde{B}_{i,i-1} = h_{i-1}/6$ $\tilde{D}_{i-1,i} = \tilde{D}_{i,i-1} = 1/h_{i-1}$	$\tilde{B}_{i,i} = (h_{i-1} + h_i)/3$ $\tilde{D}_{i,i} = -1/h_{i-1} - 1/h_i$	$i = 2 \ldots k-1$
$\tilde{B}_{1,1} = (h_{k-1} + h_1)/3$ $\tilde{D}_{1,1} = -1/h_1 - 1/h_{k-1}$	$\tilde{B}_{1,k-1} = h_{k-1}/6$ $\tilde{D}_{1,k-1} = 1/h_{k-1}$	$\tilde{B}_{k-1,1} = h_{k-1}/6$ $\tilde{D}_{k-1,1} = 1/h_{k-1}$

Table 4.2 *Definitions of basis functions and matrices used to define a cubic regression spline.* $h_j = x_{j+1} - x_j$.

which can be re-written, once more, as

$$f(x) = \sum_{i=1}^{k} b_i(x)\beta_i$$

by implicit definition of new basis functions $b_i(x)$: figure 4.1 illustrates the basis. Hence, given a set of x values, at which to evaluate the spline, it is easy to obtain a model matrix mapping $\boldsymbol{\beta}$ to the evaluated spline. It can further be shown (e.g. Lancaster and Šalkauskas, 1986, or exercise 2) that

$$\int_{x_1}^{x_k} f''(x)^2 dx = \boldsymbol{\beta}^\mathsf{T}\mathbf{D}^\mathsf{T}\mathbf{B}^{-1}\mathbf{D}\boldsymbol{\beta}$$

i.e. $\mathbf{S} \equiv \mathbf{D}^\mathsf{T}\mathbf{B}^{-1}\mathbf{D}$ is the penalty matrix for this basis.

Notice that in addition to having directly interpretable parameters, this basis does not require any re-scaling of the predictor variables before it can be used to construct a GAM, although, as with the chapter 3 basis, we do have to choose the locations of the knots x_j. See Lancaster and Šalkauskas (1986) for more details about this basis.

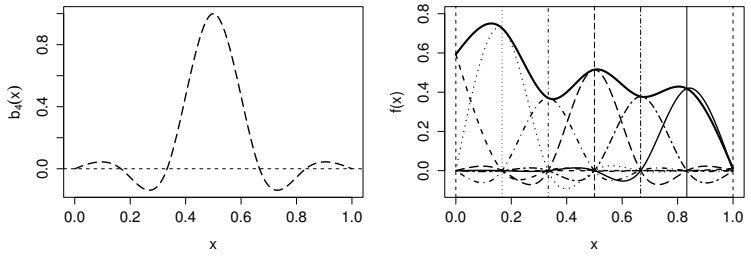

Figure 4.1 *The left hand panel illustrates one basis function,*$b_4(x)$*, for a cubic regression spline of the type discussed in section 4.1.2: this basis function takes the value one at one knot of the spline, and zero at all other knots (such basis functions are sometimes called 'cardinal basis functions'). The right hand panel shows how such basis functions are combined to represent a smooth curve. The various curves of medium thickness show the basis functions,* $b_j(x)$*, of a cubic regression spline, each multiplied by its associated coefficient* β_j*: these scaled basis functions are summed to get the smooth curve illustrated by the thick continuous curve. The vertical thin lines show the knot locations.*

4.1.3 A cyclic cubic regression spline

It is quite often appropriate for a model smooth function to be 'cyclic', meaning that the function has the same value and first few derivatives at its upper and lower boundaries. For example, in most applications, it would not be appropriate for a smooth function of time of year to change discontinuously at the year end. The penalized cubic regression spline, of the previous section, can be modified to produce such a smooth. The spline can still be written in the form (4.2), but we now have that $\beta_1 = \beta_k$ and $\delta_1 = \delta_k$. In this case then, we define vectors $\boldsymbol{\beta}^{\mathsf{T}} = (\beta_1, \ldots, \beta_{k-1})$ and $\boldsymbol{\delta}^{\mathsf{T}} = (\delta_1, \ldots, \delta_{k-1})$. The conditions that the spline must be continuous to second derivative at each knot, and that $f(x_1)$ must match $f(x_k)$, up to second derivative, are equivalent to

$$\tilde{\mathbf{B}}\boldsymbol{\delta} = \tilde{\mathbf{D}}\boldsymbol{\beta}$$

where $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{D}}$ are defined in table 4.2. Similar reasoning to that employed in the previous section implies that the spline can be written as

$$f(x) = \sum_{i=1}^{k-1} \tilde{b}_i(x)\beta_i,$$

by appropriate definition of the basis functions $\tilde{b}_i(x)$: figure 4.2 illustrates this basis. A second derivative penalty also follows:

$$\int_{x_1}^{x_k} f''(x)^2 dx = \boldsymbol{\beta}^{\mathsf{T}}\tilde{\mathbf{D}}^{\mathsf{T}}\tilde{\mathbf{B}}^{-1}\tilde{\mathbf{D}}\boldsymbol{\beta}.$$

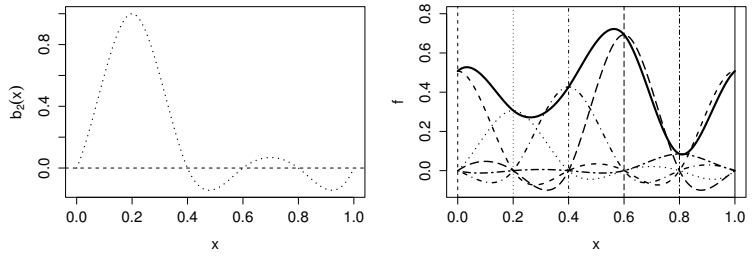

Figure 4.2 *The left hand panel illustrates one basis function, $b_2(x)$, for a cyclic cubic regression spline of the type discussed in section 4.1.3: this basis function takes the value one at one knot of the spline, and zero at all other knots - notice how the basis function values and first two derivatives match at $x = 0$ and $x = 1$. The right hand panel shows how such basis functions are combined to represent a smooth curve. The various curves of medium thickness show the basis functions, $b_j(x)$, of a cubic regression spline, each multiplied by its associated coefficient β_j: these scaled basis functions are summed to get the smooth curve illustrated by the thick continuous curve. The vertical thin lines show the knot locations.*

4.1.4 P-splines

Yet another way to represent cubic splines (and indeed splines of higher or lower order), is by use of the B-spline basis. The B-spline basis is appealing because the basis functions are strictly local — each basis function is only non-zero over the intervals between $m + 3$ adjacent knots, where $m + 1$ is the order of the basis (e.g. $m = 2$ for a cubic spline*). To define a k parameter B-spline basis, we need to define $k+m+1$ knots, $x_1 < x_2 < \ldots < x_{k+m+1}$, where the interval over which the spline is to be evaluated lies within $[x_{m+2}, x_k]$ (so that the first and last $m+1$ knot locations are essentially arbitrary). An $(m + 1)^{\text{th}}$ order spline can then be represented as

$$f(x) = \sum_{i=1}^{k} B_i^m(x)\beta_i,$$

where the B-spline basis functions are most conveniently defined recursively as follows:

$$B_i^m(x) = \frac{x - x_i}{x_{i+m+1} - x_i} B_i^{m-1}(x) + \frac{x_{i+m+2} - x}{x_{i+m+2} - x_{i+1}} B_{i+1}^{m-1}(x) \quad i = 1, \ldots k$$

and

$$B_i^{-1}(x) = \begin{cases} 1 & x_i \leq x < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

(see e.g. de Boor, 1978; Lancaster and Šalkauskas, 1986). For example, the following R code can be used to evaluate single B-spline basis functions at a series of x values:

* The somewhat inconvenient definition of order is for compatibility with the notation usually used for normal splines.

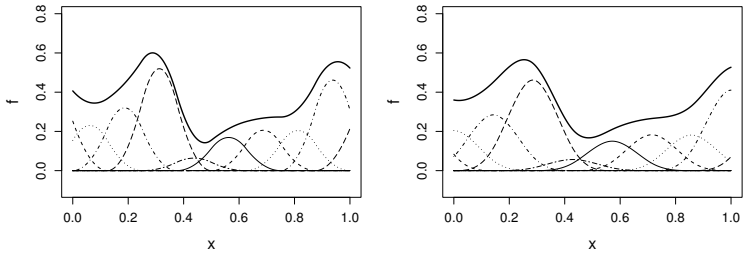

Figure 4.3 *Illustration of the representation of a smooth curve by rank 10 B-spline bases. The left plot shows a B spline basis with* $m = 1$*. The thin curves show B-spline basis functions multiplied by their associated coefficients, each is non-zero over only 3 intervals. The sum of the coefficients multiplied by the basis functions gives the spline itself, represented by the thicker continuous curve. The right panel is the same, but for a basis for which* $m = 2$*: in this case each basis function is non-zero over 4 adjacent intervals. In both panels the knot locations are where each basis function peaks.*

```
bspline <- function(x,k,i,m=2)
# evaluate ith b-spline basis function of order m at the values
# in x, given knot locations in k
{ if (m==-1) # base of recursion
  { res <- as.numeric(x<k[i+1]&x>=k[i])
  } else     # construct from call to lower order basis
  { z0 <- (x-k[i])/(k[i+m+1]-k[i])
    z1 <- (k[i+m+2]-x)/(k[i+m+2]-k[i+1])
    res <- z0*bspline(x,k,i,m-1)+ z1*bspline(x,k,i+1,m-1)
  }
  res
}
```

Figure 4.3 illustrates the representation of functions using B-spline bases of two different orders.

B-splines were developed as a very stable basis for large scale spline interpolation (see de Boor, 1978, for further details), but for most statistical work with low rank penalized regression splines, you would have to be using very poor numerical methods before the enhanced stability of the basis became noticeable. The real statistical interest in B-splines has resulted from the work of Eilers and Marx (1996) in using them to develop what they term *P-splines*.

P-splines are low rank smoothers using a B-spline basis, usually defined on evenly spaced knots, and a *difference penalty* applied directly to the parameters, β_i, to control function wiggliness. How this works is best seen by example. If we decide to penalize the squared difference between adjacent β_i values then the penalty would

be

$$\mathcal{P} = \sum_{i=1}^{k-1} (\beta_{i+1} - \beta_i)^2 = \beta_1^2 - 2\beta_1\beta_2 + 2\beta_2^2 - 2\beta_2\beta_3 + \ldots + \beta_k^2,$$

and it is straightforward to see that this can be written

$$\mathcal{P} = \boldsymbol{\beta}^{\mathsf{T}} \begin{bmatrix} 1 & -1 & 0 & . & . \\ -1 & 2 & -1 & . & . \\ 0 & -1 & 2 & . & . \\ . & . & . & . & . \\ . & . & . & . & . \end{bmatrix} \boldsymbol{\beta}.$$

Such penalties are very easily generated in R. For example the penalty matrix for $\mathcal{P}$ can be generated by:

```
k<-6                                # example basis dimension
P <- diff(diag(k),differences=1)    # sqrt of penalty matrix
S <- t(P)%*%P                       # penalty matrix
```

Higher order penalties are produced by increasing the `differences` parameter. The only lower order penalty is the identity matrix.

P-splines are extremely easy to set up and use, and allow a good deal of flexibility, in that any order of penalty can be combined with any order of B-spline basis, as the user sees fit. Their disadvantage is that the simplicity is somewhat diminished if uneven knot spacing is required, and that the penalties are less easy to interpret in terms of the properties of the fitted smooth, than the more usual spline penalties. See exercises 7 to 9, for further coverage of P-splines.

4.1.5 Thin plate regression splines

The bases covered so far are each useful in practice, but are open to some criticisms.

1. It is necessary to choose knot locations, in order to use each basis: this introduces an extra degree of subjectivity into the model fits.
2. The bases are only useful for representing smooths of one predictor variable.
3. It is not clear to what extent the bases are better or worse than any other basis that might be used.

In this section, an approach is developed which goes some way to addressing these issues, by producing knot free bases, for smooths of any number of predictors, that are in a certain limited sense 'optimal': the thin plate regression splines.

Thin plate splines

Thin plate splines (Duchon, 1977) are a very elegant and general solution to the problem of estimating a smooth function of multiple predictor variables, from noisy

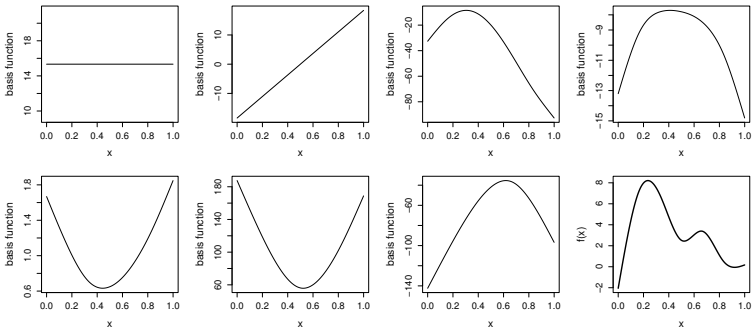

Figure 4.4 *Illustration of a thin plate spline basis for representing a smooth function of one variable fitted to 7 data with penalty order $m = 2$. The first 7 panels (starting at top left) show the basis functions, multiplied by coefficients, that are summed to give the smooth curve in the lower right panel. The first two basis functions span the space of functions that are completely smooth, according to the wiggliness measure. The remaining basis functions represent the wiggly component of the smooth curve: these latter functions are shown after absorbtion of the thin plate spline constraints $\mathbf{T}^{\mathsf{T}}\boldsymbol{\delta} = \mathbf{0}$ into the basis.*

observations of the function, at particular values of those predictors. Consider then, the problem of estimating the smooth function $g(\mathbf{x})$, from n observations $(y_i, \mathbf{x}_i)$ such that

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

where ϵ_i is a random error term and where $\mathbf{x}$ is a d - vector ($d \leq n$). Thin-plate spline smoothing estimates g by finding the function $\hat{f}$ minimizing:

$$\|\mathbf{y} - \mathbf{f}\|^2 + \lambda J_{md}(f) \tag{4.4}$$

where $\mathbf{y}$ is the vector of y_i data and $\mathbf{f} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n))^{\mathsf{T}}$. $J_{md}(f)$ is a penalty functional measuring the 'wiggliness' of f, and λ is a smoothing parameter, controlling the tradeoff between data fitting and smoothness of f. The wiggliness penalty is defined as

$$J_{md} = \int \ldots \int_{\Re^d} \sum_{\nu_1 + \ldots + \nu_d = m} \frac{m!}{\nu_1! \ldots \nu_d!} \left(\frac{\partial^m f}{\partial x_1^{\nu_1} \ldots \partial x_d^{\nu_d}} \right)^2 dx_1 \ldots dx_d.^{\dagger} \tag{4.5}$$

Further progress is only possible if m is chosen so that $2m > d$, and in fact for 'visually smooth' results it is preferable that $2m > d+1$. Subject to the first of these

† The general form of the penalty is somewhat intimidating, so an example is useful. In the case of a smooth of two predictors with wiggliness measured using second derivatives, we have

$$J_{22} = \int \int \left(\frac{\partial^2 f}{\partial x_1^2} \right)^2 + \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 f}{\partial x_2^2} \right)^2 dx_1 dx_2.$$

restrictions, it can be shown that the function minimizing (4.4) has the form,

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} \delta_i \eta_{md}(\|\mathbf{x} - \mathbf{x}_i\|) + \sum_{j=1}^{M} \alpha_j \phi_j(\mathbf{x}), \tag{4.6}$$

where $\boldsymbol{\delta}$ and $\boldsymbol{\alpha}$ are vectors of coefficients to be estimated, $\boldsymbol{\delta}$ being subject to the linear constraints that $\mathbf{T}^{\mathsf{T}}\boldsymbol{\delta} = \mathbf{0}$ where $T_{ij} = \phi_j(\mathbf{x}_i)$. The $M = \binom{m+d-1}{d}$ functions, ϕ_i, are linearly independent polynomials spanning the space of polynomials in $\Re^d$ of degree less than m. The ϕ_i span the space of functions for which J_{md} is zero, i.e. the 'null space' of J_{md}: those functions that are considered 'completely smooth'. For example, for $m = d = 2$ these functions are $\phi_1(\mathbf{x}) = 1$, $\phi_2(\mathbf{x}) = x_1$ and $\phi_3(\mathbf{x}) = x_2$. The remaining basis functions used in (4.6) are defined as

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

Now defining matrix $\mathbf{E}$ by $E_{ij} \equiv \eta_{md}(\|\mathbf{x}_i - \mathbf{x}_j\|)$, the thin plate spline fitting problem becomes,

$$\text{minimize } \|\mathbf{y} - \mathbf{E}\boldsymbol{\delta} - \mathbf{T}\boldsymbol{\alpha}\|^2 + \lambda\boldsymbol{\delta}^{\mathsf{T}}\mathbf{E}\boldsymbol{\delta} \text{ subject to } \mathbf{T}^{\mathsf{T}}\boldsymbol{\delta} = \mathbf{0}, \tag{4.7}$$

with respect to $\boldsymbol{\delta}$ and $\boldsymbol{\alpha}$. Wahba (1990) or Green and Silverman (1994) provide further information about thin-plate splines, and figure 4.4 illustrates a thin plate spline basis in one dimension.

The thin plate spline, $\hat{f}$, is something of an ideal smoother: it has been constructed by defining exactly what is meant by smoothness, exactly how much weight to give to the conflicting goals of matching the data and making $\hat{f}$ smooth, and finding the *function* that best satisfies the resulting smoothing objective. Notice that in doing this we did not have to choose knot positions or select basis functions, both of these emerged naturally from the mathematical statement of the smoothing problem. In addition, thin plate splines can deal with any number of predictor variables, and allow the user some flexibility to select the order of derivative used in the measure of function wiggliness. So, at first sight it might seem that the problems listed at the start of this section are all solved, and thin plate spline bases and penalties should be used to represent all the smooth terms in the model.

The problem with thin plate splines is computational cost: these smoothers have as many unknown parameters as there are data (strictly, number of unique predictor combinations), and, except in the single predictor case, the computational cost of model estimation is proportional to the cube of the number of parameters. This is a very high price to pay for using such smooths. Given that the effective degrees of freedom estimated for a model term is usually a small proportion of n, it seems wasteful to use so many parameters to represent the term, and this begs the question of whether a low rank approximation could be produced which is as close as possible to the thin plate spline smooth, without incurring prohibitive computational cost.

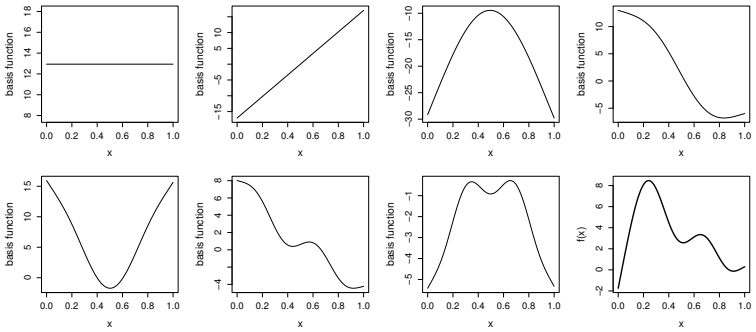

Figure 4.5 *Illustration of a rank 7 thin plate regression spline basis for representing a smooth function of one variable, with penalty order $m = 2$. The first 7 panels (starting at top left) show the basis functions, multiplied by coefficients, that are summed to give the smooth curve in the lower right panel. The first two basis functions span the space of functions that are completely smooth, according to the wiggliness measure. The remaining basis functions represent the wiggly component of the smooth curve: notice how these functions become successively more wiggly while generally tending to contribute less and less to the overall fit.*

Thin plate regression splines

Thin plate regression splines are based the idea of truncating the space of the wiggly components of the thin plate spline (the components with parameters $\boldsymbol{\delta}$), while leaving the components of 'zero wiggliness' unchanged (the $\boldsymbol{\alpha}$ components). Let $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T}$ be the eigen-decomposition of $\mathbf{E}$, so that $\mathbf{D}$ is a diagonal matrix of eigenvalues of $\mathbf{E}$ arranged so that $|D_{i,i}| \geq |D_{i-1,i-1}|$ and the columns of $\mathbf{U}$ are the corresponding eigenvectors. Now let $\mathbf{U}_k$ denote the matrix consisting of the first k columns of $\mathbf{U}$ and $\mathbf{D}_k$ denote the top right $k \times k$ submatrix of $\mathbf{D}$. Restricting $\boldsymbol{\delta}$ to the columns space of $\mathbf{U}_k$, by writing $\boldsymbol{\delta} = \mathbf{U}_k\boldsymbol{\delta}_k$, means that (4.7) becomes

$$\text{minimise } \|\mathbf{y} - \mathbf{U}_k\mathbf{D}_k\boldsymbol{\delta}_k - \mathbf{T}\boldsymbol{\alpha}\|^2 + \lambda\boldsymbol{\delta}_k^\mathsf{T}\mathbf{D}_k\boldsymbol{\delta}_k \text{ subject to } \mathbf{T}^\mathsf{T}\mathbf{U}_k\boldsymbol{\delta}_k = \mathbf{0}$$

w.r.t. $\boldsymbol{\delta}_k$ and $\boldsymbol{\alpha}$. The constraints can be absorbed in the usual manner, described in section 1.8.1. We first find any orthogonal column basis, $\mathbf{Z}_k$, such that $\mathbf{T}^\mathsf{T}\mathbf{U}_k\mathbf{Z}_k = \mathbf{0}$. One way to do this is to form the QR decomposition of $\mathbf{U}_k^\mathsf{T}\mathbf{T}$: the final M columns of the orthogonal factor give a $\mathbf{Z}_k$ (see sections 1.8.1 and A.6). Restricting $\boldsymbol{\delta}_k$ to this space, by writing $\boldsymbol{\delta}_k = \mathbf{Z}_k\tilde{\boldsymbol{\delta}}$, yields the unconstrained problem that must be solved to fit the rank k approximation to the smoothing spline:

$$\text{minimise } \|\mathbf{y} - \mathbf{U}_k\mathbf{D}_k\mathbf{Z}_k\tilde{\boldsymbol{\delta}} - \mathbf{T}\boldsymbol{\alpha}\|^2 + \lambda\tilde{\boldsymbol{\delta}}^\mathsf{T}\mathbf{Z}_k^\mathsf{T}\mathbf{D}_k\mathbf{Z}_k\tilde{\boldsymbol{\delta}}$$

with respect to $\tilde{\boldsymbol{\delta}}$ and $\boldsymbol{\alpha}$. This has a computational cost of $O(k^3)$. Having fitted the model, evaluation of the spline at any point is easy: simply evaluate $\boldsymbol{\delta} = \mathbf{U}_k\mathbf{Z}_k\tilde{\boldsymbol{\delta}}$ and use (4.6).

Now, the main problem is how to find $\mathbf{U}_k$ and $\mathbf{D}_k$ sufficiently cheaply. A full eigen-decomposition of $\mathbf{E}$ requires $O(n^3)$ operations, which would somewhat limit the utility of the TPRS approach. Fortunately the method of Lanczos iteration can be employed to find $\mathbf{U}_k$ and $\mathbf{D}_k$ at the substantially lower cost of $O(n^2k)$ operations. See Appendix A, section A.11, for a suitable Lanczos algorithm.

Properties of thin plate regression splines

It is clear that thin plate regression splines avoid the problem of knot placement, are relatively cheap to compute, and can be constructed for smooths of any number of predictor variables, but what of their optimality properties? The thin-plate splines are optimal in the sense that no smooth function will better minimize (4.4), but to what extent is that optimality inherited by the TPRS approximation? To answer this it helps to think about what would make a good approximation. An ideal approximation would probably result in the minimum possible perturbation of the fitted values of the spline, at the same time as making the minimum possible change to the 'shape' of the fitted spline. It is difficult to see how both these aims could be achieved, for all possible response data, without first fitting the full thin plate spline. But if the criteria are loosened somewhat to minimizing the worst possible changes in shape and fitted value then progress can be made, as follows.

The basis change and truncation can be thought of as replacing $\mathbf{E}$, in the norm in (4.7), by the matrix $\hat{\mathbf{E}} = \mathbf{E}\mathbf{U}_k\mathbf{U}_k^{\mathsf{T}}$, while replacing $\mathbf{E}$, in the penalty term of (4.7), by $\tilde{\mathbf{E}} = \mathbf{U}_k^{\mathsf{T}}\mathbf{U}_k\mathbf{E}\mathbf{U}_k\mathbf{U}_k^{\mathsf{T}}$. Now since the fitted values of the spline are given by $\mathbf{E}\hat{\boldsymbol{\delta}} + \mathbf{T}\boldsymbol{\alpha}$, the worst possible change in fitted values could be measured by:

$$\hat{e}_k = \max_{\boldsymbol{\delta}\neq\mathbf{0}} \frac{\|(\mathbf{E} - \hat{\mathbf{E}}_k)\boldsymbol{\delta}\|}{\|\boldsymbol{\delta}\|}.$$

(dividing by $\|\boldsymbol{\delta}\|$ is necessary since the upper norm otherwise has a maximum at infinity.) The 'shape' of the spline is measured by the penalty term in (4.7), so a suitable measure of the worst possible change in the shape of the spline caused by the truncation might be:

$$\tilde{e}_k = \max_{\boldsymbol{\delta}\neq\mathbf{0}} \frac{\boldsymbol{\delta}^{\mathsf{T}}(\mathbf{E} - \tilde{\mathbf{E}}_k)\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|^2}.$$

It turns out to be quite easy to show that $\hat{e}_k$ and $\tilde{e}_k$ are simultaneously minimized by the choice of $\mathbf{U}_k$, as the truncated basis for $\boldsymbol{\delta}$. i.e. there is no matrix of the same dimension as $\mathbf{U}_k$ which would lead to lower $\hat{e}_k$ or $\tilde{e}_k$, if used in place of $\mathbf{U}_k$ (see Wood, 2003).

Note that $\hat{e}_k$ and $\tilde{e}_k$ are really formulated in too large a space. Ideally we would impose the constraints $\mathbf{T}^{\mathsf{T}}\boldsymbol{\delta} = \mathbf{0}$ on both, but in that case different bases minimize the two criteria. This in turn leads to the question of whether it would not be better to concentrate on just one of the criteria, but this is unsatisfactory, as it leads to results that depend on how the original thin plate spline problem is parameterized.

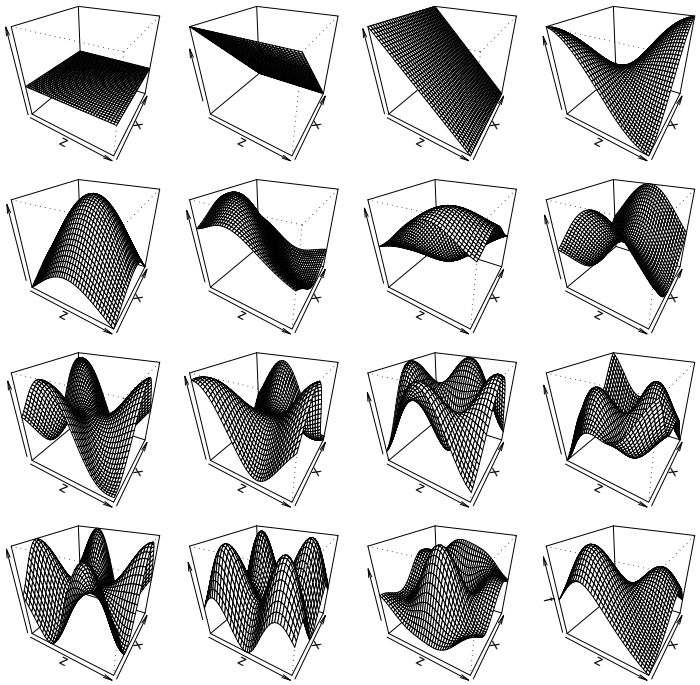

Figure 4.6 *Illustration of a rank 15 thin plate regression spline basis for representing a smooth function of two variables, with penalty order* $m = 2$*. The first 15 panels (starting at top left) show the basis functions, multiplied by coefficients, that are summed to give the smooth surface in the lower right panel. The first three basis functions span the space of functions that are completely smooth, according to the wiggliness measure,* J_{22}*. The remaining basis functions represent the wiggly component of the smooth curve: notice how these functions become successively more wiggly.*

Furthermore, these results can be extremely poor for some parameterizations. For example, if the thin plate spline. is parameterized in terms of the fitted values, then the $\hat{e}_k$ optimal approximation is not smooth. Similarly, very poor fitted values result from an $\tilde{e}_k$ optimal approximation to a thin plate spline., if that thin plate spline is parameterized so that the penalty matrix is an identity matrix, with some leading diagonal entries zeroed.

To sum up: thin plate regression splines are probably the best that can be hoped for in terms of approximating the behaviour of a thin plate spline using a basis of any given low rank. They have the nice property of avoiding having to choose 'knot locations', and are reasonably computationally efficient, if Lanczos iteration is used to find the truncated eigen-decomposition of $\mathbf{E}$. They also retain the rotational invari-

ance (isotropy) of full thin plate spline. Figures 4.5 and 4.6 provide examples of the bases functions that result from adopting a t.p.r.s approach.

Knot based approximation

If one is prepared to forgo optimality, and choose knot locations, then a simpler approximation is available, which avoids the truncated eigen-decomposition. If knot locations $\{\mathbf{x}_i^* : i = 1 \dots k\}$ are chosen, then the spline can be approximated by

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{k} \delta_i \eta_{md}(\|\mathbf{x} - \mathbf{x}_i^*\|) + \sum_{j=1}^{M} \alpha_j \phi_j(\mathbf{x}) \tag{4.8}$$

where $\boldsymbol{\delta}$ and $\boldsymbol{\alpha}$ are estimated by minimizing

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \boldsymbol{\beta}^\mathsf{T} \mathbf{S} \boldsymbol{\beta} \text{ subject to } \mathbf{C}\boldsymbol{\beta} = \mathbf{0}$$

w.r.t. $\boldsymbol{\beta}^\mathsf{T} = (\boldsymbol{\delta}^\mathsf{T}, \boldsymbol{\alpha}^\mathsf{T})$. $\mathbf{X}$ is an $n \times k + M$ matrix such that

$$X_{ij} = \begin{cases} \eta_{md}(\|\mathbf{x}_i - \mathbf{x}_j^*\|) & j = 1, \dots, k \\ \phi_{j-k}(x_i) & j = k+1, \dots, k+M. \end{cases}$$

$\mathbf{S}$ is a $(k + M) \times (k + M)$ matrix with zeroes everywhere except in its upper left $k \times k$ block where $S_{ij} = \eta_{md}(\|\mathbf{x}_i^* - \mathbf{x}_j^*\|)$. Finally, $\mathbf{C}$ is an $M \times (k + M)$ matrix such that

$$C_{ij} = \begin{cases} \phi_i(\mathbf{x}_j^*) & j = 1, \dots, k \\ 0 & j = k+1, \dots, k+M. \end{cases}$$

This approximation goes back at least to Wahba (1980). Some care is required to choose the knot locations carefully. In one dimension it is usual to choose quantiles of the empirical distribution of the predictor, or even spacing, but in more dimensions matters are often more difficult. One possibility is to take a random sample of the observed predictor variable combinations, another to take a 'spatially stratified' sample of the predictor variable combinations. Even spacing is sometimes appropriate, or more sophisticated space filling schemes can be used: Ruppert et al. (2003) provide a useful discussion of the alternatives.

4.1.6 Shrinkage smoothers

A disadvantage of the smooths discussed so far, is that no matter how large their associated smoothing parameter becomes, the smooth is never completely eliminated in the sense of having all its parameters estimated to be zero. On the contrary, some functions are treated as completely smooth by the penalty, and hence functions of this class are always completely un-penalized. From the point of view of model selection with GAMs it would be more convenient if smooths could be zeroed by adjustment of smoothing parameters. One way to do this would be to add an extra penalty, with associated smoothing parameter which acted only on the unpenalized functions, but this would open up the possibility of penalizing the smooth components of a function

more than the wiggly components, which seems unsatisfactory, as well as requiring an extra smoothing parameter per smooth. A fairly crude alternative, is simply to add a small multiple of the identity matrix to the penalty matrix of the smooth, i.e.

$$\mathbf{S} \rightarrow \mathbf{S} + \epsilon \mathbf{I}$$

so that the penalty will now shrink all parameters to zero if its associated smoothing parameter is large enough. If ϵ is small enough, the identity part of the penalty will have almost no impact when a function is 'wiggly': only once it becomes close to 'completely smooth' will the identity component start to become important, and really start shrinking the parameters towards zero.

4.1.7 Choosing the basis dimension

When using penalized regression splines the modeller chooses the basis dimension as part of the model building process. Typically, this substantially reduces the computational burden of modelling, relative to full spline methods, and recognizes the fact that, usually, something is seriously wrong if a statistical model really *requires* as many coefficients as there are data. Working with fixed basis dimensions also makes it rather trivial to demonstrate large sample consistency, and other properties, of the smoothing methods, but only at the cost of a slightly artificial assumption that the truth is really in the space spanned by the reduced basis.

The main challenge introduced, by this low rank approach, is that a basis dimension has to be chosen. In the context of spline smoothing, Kim and Gu (2004) showed that the basis size should scale as $n^{2/9}$, where n is the number of data. Based on simulation they suggested using $10n^{2/9}$ as the basis dimension, but it is hard to see how one can really know what the constant of proportionality should be, without knowing the truth that is being estimated. Chapter 5 includes several examples where the rule appears to give too small a basis dimension, for example in section 5.6.2. Wood (2006) also suggests that the basis dimension should depend on the number of covariates of a smooth, as well as the sample size.

In practice, then, choice of basis dimension is something that probably has to remain a part of model specification. However, it is important to note that the exact size of basis dimension is really not that critical. The basis dimension is only setting an upper bound on the flexibility of a term: it is the smoothing parameter that controls the actual effective degrees of freedom. Hence the model fit is usually rather insensitive to the basis dimension, provided that it is not set restrictively low for the application concerned. The only caveat to this point is the slightly subtle one, that a function space with basis dimension 20 will contain a larger space of functions with EDF 5 than will a function space of dimension 10 (the numbers being arbitrary): it is this fact that causes model fit to retain some sensitivity to basis dimension, even if the appropriate EDF for a term is well below the basis dimension.

In practice, the modeller needs to decide roughly how large a basis dimension is fairly certain to provide adequate flexibility, in any particular application, and use that.

4.1.8 Tensor product smooths

A major feature of the thin plate (regression) spline approach of section 4.1.5 is the isotropy of the wiggliness penalty: wiggliness in all directions is treated equally, with the fitted spline entirely invariant to rotation of the co-ordinate system for the predictor variables. For example, suppose we were to measure air pollution at a fixed set of points in Southern England, measuring the location of the points relative to the UK national grid and modelling pollution levels as a smooth function of the two spatial co-ordinates. Now suppose that the locations were instead measured on the French grid, and the modelling exercise repeated: the model fit would be identical (provided that the earth is flat).

This isotropy is often considered to be desirable when modelling things as a smooth function of geographic co-ordinates[‡], but it has some disadvantages. Chief among them is the difficulty of knowing how to scale predictors relative to one another, when both are arguments of the same smooth, but they are measured in fundamentally different units. For example, consider a smooth function of a single spatial co-ordinate and time: the implied relative importance of smoothness in time versus smoothness in space, is very different between a situation in which the units are metres and hours, compared to that in which the units are light-years and nanoseconds. One pragmatic approach is to scale all predictors into the unit square, as is often done in loess smoothing, but this is essentially arbitrary. A more satisfactory approach uses *tensor product smooths*.

Tensor product bases

The basic approach of this section is to start from smooths of single covariates, represented using any basis with associated quadratic penalty measuring 'wiggliness' of the smooth. From these 'marginal smooths' a 'tensor product' construction is used to build up smooths of several variables. See de Boor (1978) for an important early reference on tensor product spline bases.

The methods developed here can be used to construct smooth functions of *any* number of covariates, but the simplest introduction is via the construction of a smooth function of 3 covariates, x, z and v, the generalization then being trivial. The process starts by assuming that we have low rank bases available, for representing smooth functions f_x, f_z and f_v of each of the covariates. That is we can write:

$$f_x(x) = \sum_{i=1}^{I} \alpha_i a_i(x), \quad f_z(z) = \sum_{l=1}^{L} \delta_l d_l(z) \quad \text{and} \quad f_v(v) = \sum_{k=1}^{K} \beta_k b_k(v),$$

where the α_i, δ_l and β_k are parameters, and the $a_i(x)$, $d_l(z)$ and $b_k(v)$ are known basis functions.

[‡] Although it's possible to overstate the case for doing this: in many applications at many locations North-South is not the same as East-West.

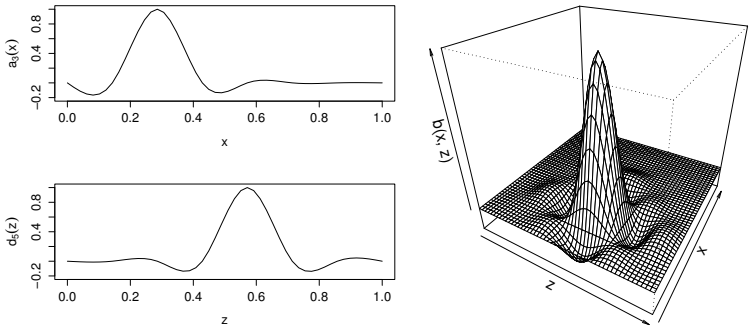

Figure 4.7 *How the product of two marginal basis functions for smooth functions of x and z, separately, results in a basis function for a smooth function of x and z together. The two left panels show the 3rd and 5th basis functions for rank 8 cubic regression spline smooths of x and z respectively. The right hand plot shows $a_3(x)b_5(z)$, one of 64 similar basis functions of the tensor product smooth derived from these two marginal smooths.*

Now consider how the smooth function of x, f_x, could be converted into a smooth function of x and z. What is required is for f_x to vary smoothly with z, and this can be achieved by allowing its parameters, α_i, to vary smoothly with z. Using the basis already available for representing smooth functions of z we could write:

$$\alpha_i(z) = \sum_{l=1}^{L} \delta_{il} d_l(z)$$

which immediately gives

$$f_{xz}(x, z) = \sum_{i=1}^{I} \sum_{l=1}^{L} \delta_{il} d_l(z) a_i(x).$$

Figure 4.7 illustrates this construction. Continuing in the same way, we could now create a smooth function of x, z and v by allowing f_{xz} to vary smoothly with v. Again, the obvious way to do this is to let the parameters of f_{xz} vary smoothly with v, and following the same reasoning as before we get

$$f_{xzv}(x, z, v) = \sum_{i=1}^{I} \sum_{l=1}^{L} \sum_{k=1}^{K} \beta_{ilk} b_k(v) d_l(z) a_i(x).$$

For any particular set of observations of x, z and v, there is a simple relationship between the model matrix, $\mathbf{X}$, evaluating the tensor product smooth at these observations, and the model matrices $\mathbf{X}_x$, $\mathbf{X}_z$ and $\mathbf{X}_v$ that would evaluate the marginal smooths at the same observations. If $\otimes$ is the usual Kronecker product (see section A.4), then it is easy to show that, given appropriate ordering of the β_{ilk} into a vector

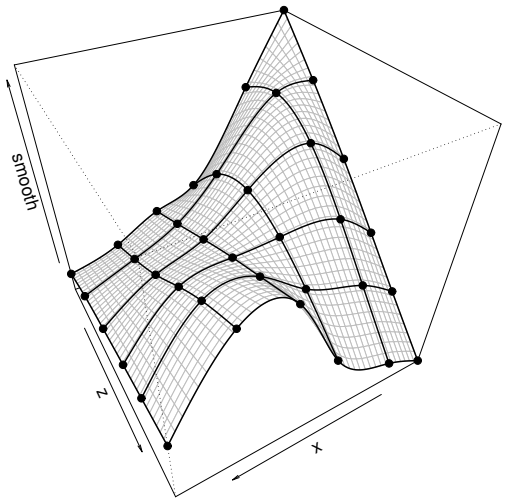

Figure 4.8 *Illustration of a tensor product smooth of two variables x and z, constructed from 2 rank 6 marginal bases. Following section 4.1.8 a tensor product smooth can always be parameterized in terms of the values of the function at a set of 'knots' spread over the function domain on a regular mesh: i.e. in terms of the heights of the $\bullet$'s shown. The basis construction can be thought of as follows: start with a smooth of x parameterized in terms of function values at a set of 'knots'; to make the smooth of x vary smoothly with z, simply allow each of its parameters to vary smoothly with z: this can be done by representing each parameter using a smooth of z, also parameterized in terms of function values at a set of 'knots'. Exactly the same smooth arises if we reverse the roles of x and z in this construction. The tensor product smooth penalty in the x direction, advocated in section 4.1.8, is simply the sum of the the marginal wiggliness measure for the smooth of x applied to the thick black curves parallel to the x axes: the z penalty is similarly defined in terms of the marginal penalty of the smooth of z applied to the thick black curves parallel to the z axis.*

$\boldsymbol{\beta}$, the i^{th} row of $\mathbf{X}$ is simply:

$$\mathbf{X}_i = \mathbf{X}_{xi} \otimes \mathbf{X}_{zi} \otimes \mathbf{X}_{vi}.$$

Clearly (i) this construction can be continued for as many covariates as are required; (ii) the result is independent of the order in which we treat the covariates and (iii) the covariates can themselves be vector covariates. Figure 4.8 attempts to illustrate the tensor product construction for a smooth of two covariates.

Tensor product penalties

Having derived a 'tensor product' basis for representing smooth functions, it is also necessary to have some way of measuring function 'wiggliness', if the basis is to be useful for representing smooth functions in a GAM context. Again, it is possible to start from wiggliness measures associated with the marginal smooth functions, and again the three covariate case provides sufficient illustration. Suppose then, that each marginal smooth has an associated functional that measures function wiggliness, and can be expressed as a quadratic form in the marginal parameters. That is

$$J_x(f_x) = \boldsymbol{\alpha}^\mathsf{T}\mathbf{S}_x\boldsymbol{\alpha}, \quad J_z(f_z) = \boldsymbol{\delta}^\mathsf{T}\mathbf{S}_z\boldsymbol{\delta} \text{ and } J_v(f_v) = \mathcal{B}^\mathsf{T}\mathbf{S}_v\mathcal{B}.$$

The $\mathbf{S}_\bullet$ matrices contain known coefficients, and $\boldsymbol{\alpha}$, $\boldsymbol{\delta}$ and $\mathcal{B}$ are vectors of coefficients of the marginal smooths. An example of a penalty functional is the cubic spline penalty, $J_x(f_x) = \int(\partial^2 f_x/\partial x^2)^2 dx$. Now let $f_{x|zv}(x)$ be $f_{xvz}(x,z,v)$ considered as a function of x only, with z and v held constant, and define $f_{z|xv}(z)$ and $f_{v|xz}(v)$ similarly. A natural way of measuring wiggliness of f_{xzv} is to use:

$$J(f_{xzv}) = \lambda_x \int_{z,v} J_x(f_{x|zv})dzdv + \lambda_z \int_{x,v} J_z(f_{z|xv})dxdv + \lambda_v \int_{x,z} J_v(f_{v|xz})dxdz$$

where the $\lambda_\bullet$ are smoothing parameters controlling the tradeoff between wiggliness in different directions, and allowing the penalty to be invariant to the relative scaling of the covariates. As an example, if cubic spline penalties were used as the marginal penalties, then

$$J(f) = \int_{x,z,v} \lambda_x \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + \lambda_z \left(\frac{\partial^2 f}{\partial z^2}\right)^2 + \lambda_v \left(\frac{\partial^2 f}{\partial v^2}\right)^2 dxdzdv.$$

Hence, if the marginal penalties are easily interpretable, in terms of function shape, then so is the induced penalty. Numerical evaluation of the integrals in J is straightforward. As an example consider the penalty in the x direction. The function $f_{x|zv}(x)$ can be written as

$$f_{x|zv}(x) = \sum_{i=1}^{I} \alpha_i(z,v)a_i(x),$$

and it is always possible to find the matrix of coefficients $\mathbf{M}_{z,v}$ such that $\boldsymbol{\alpha}(z,v) = \mathbf{M}_{zv}\boldsymbol{\beta}$ where $\boldsymbol{\beta}$ is the vector of β_{ilk} arranged in some appropriate order. Hence

$$J_x(f_{x|zv}) = \boldsymbol{\alpha}(z,v)^\mathsf{T}\mathbf{S}_x\boldsymbol{\alpha}(z,v) = \boldsymbol{\beta}^\mathsf{T}\mathbf{M}_{zv}^\mathsf{T}\mathbf{S}_x\mathbf{M}_{zv}\boldsymbol{\beta}$$

and so

$$\int_{z,v} J_x(f_{x|zv})dzdv = \boldsymbol{\beta}^\mathsf{T} \int_{z,v} \mathbf{M}_{zv}^\mathsf{T}\mathbf{S}_x\mathbf{M}_{zv}dzdv\boldsymbol{\beta}.$$

The last integral can be performed numerically, and it is clear that the same approach can be applied to all components of the penalty. However, a simple reparameterization can be used to provide an approximation to the terms in the penalty, which performs well in practice, and avoids the need for explicit numerical integration.

To see how the approach works, consider the marginal smooth f_x. Let $\{x_i^* : i =$

$1, \ldots, I\}$ be a set of values of x spread evenly through the range of the observed x values. In this case we can always re-parameterize f_x in terms of new parameters

$$\alpha'_i = f_x(x^*_i).$$

Clearly under this re-parameterization $\boldsymbol{\alpha}' = \boldsymbol{\Gamma}\boldsymbol{\alpha}$ where $\Gamma_{ij} = a_i(x^*_j)$. Hence the marginal model matrix becomes $\mathbf{X}'_x = \mathbf{X}_x\boldsymbol{\Gamma}^{-1}$ and the penalty coefficient matrix becomes $\mathbf{S}'_x = \boldsymbol{\Gamma}^{-\mathsf{T}}\mathbf{S}_x\boldsymbol{\Gamma}^{-1}$.

Now suppose that the same sort of re-parameterization is applied to the marginal smooths f_v and f_z. In this case we have that

$$\int_{z,v} J_x(f_{x|zv})dzdv \approx h\sum_{lk} J_x(f_{x|z^*_l v^*_k}),$$

where h is some constant of proportionality related to the spacing of the z^*_l's and v^*_k's. Similar expressions hold for the other integrals making up J. It is straightforward to show that the summation in the above approximation is:

$$J^*_x(f_{xzv}) = \boldsymbol{\beta}^{\mathsf{T}}\tilde{\mathbf{S}}_x\boldsymbol{\beta} \text{ where } \tilde{\mathbf{S}}_x = \mathbf{S}'_x \otimes \mathbf{I}_L \otimes \mathbf{I}_K$$

and $\mathbf{I}_L$ is the rank L identity matrix. Exactly similar definitions hold for the other components of the penalty so that

$$J^*_z(f_{xzv}) = \boldsymbol{\beta}^{\mathsf{T}}\tilde{\mathbf{S}}_z\boldsymbol{\beta} \text{ where } \tilde{\mathbf{S}}_z = \mathbf{I}_I \otimes \mathbf{S}'_z \otimes \mathbf{I}_K$$

and

$$J^*_v(f_{xzv}) = \boldsymbol{\beta}^{\mathsf{T}}\tilde{\mathbf{S}}_v\boldsymbol{\beta} \text{ where } \tilde{\mathbf{S}}_v = \mathbf{I}_I \otimes \mathbf{I}_L \otimes \mathbf{S}'_v.$$

Hence

$$J(f_{xzv}) \approx J^*(f_{xzv}) = \lambda_x J^*_x(f_{xzv}) + \lambda_z J^*_z(f_{xzv}) + \lambda_v J^*_v(f_{xzv}),$$

where any constants, h, have been absorbed into the λ_j. Again, this penalty construction clearly generalizes to any number of covariates. Figure 4.8 attempts to illustrate what the penalties actually measure, for a smooth of two variables.

Given its model matrix and penalties, the coefficients and smoothing parameters of a tensor product smooth can be estimated as GAM components using the methods of sections 4.3 and 4.6 or 4.7. These smooths have the nice property of being invariant to rescaling of the covariates, provided only that the marginal smooths are similarly invariant (which is always the case in practice).

Note that it is possible to omit the reparameterization of the marginal smooths, in terms of function values, and to work with penalties of the form

$$\boldsymbol{\beta}^{\mathsf{T}}\bar{\mathbf{S}}_z\boldsymbol{\beta} \text{ where } \bar{\mathbf{S}}_z = \mathbf{I}_I \otimes \mathbf{S}_z \otimes \mathbf{I}_K$$

for example: Eilers and Marx (2003) successfully used this approach to smooth with respect to two variables using tensor products of B-splines. A potential problem with the approach is that the penalties no-longer have the interpretation in terms of (averaged) function shape, that is inherited from the marginal smooths when re-parameterization is used. Another proposal in the literature is to use single penalties

of the form:

$$\boldsymbol{\beta}^{\mathsf{T}}\mathbf{S}\boldsymbol{\beta} \text{ where } \mathbf{S} = \mathbf{S}_1 \otimes \mathbf{S}_2 \otimes \cdots \otimes \mathbf{S}_d.$$

but this often leads to severe undersmoothing. The reason for the undersmoothing is straightforward: the rank of $\mathbf{S}$ is the product of the ranks of the $\mathbf{S}_j$, and in practice this is often far too low for practical work. For example, consider a smooth of 3 predictors constructed as a tensor product smooth of 3 cubic spline bases, each of rank 5. The resulting smooth would have 125 free parameters, but a penalty matrix of rank 27. This means that varying the weight given to the penalty would only result in the effective degrees of freedom for the smooth varying between 98 and 125: not a very useful range. By contrast, for the same marginal bases, the multiple term penalties would have rank 117, leading to a much more useful range of effective degrees of freedom of between 8 and 125.

4.2 Setting up GAMs as penalized GLMs

As we saw in Chapter 3, a GAM models a response variable, y_i, using a model structure of a form like:

$$g(\mu_i) = \mathbf{X}_i^*\boldsymbol{\theta} + f_1(x_{1i}) + f_2(x_{2i}, x_{3i}) + f_3(x_{4i}) + \cdots \tag{4.9}$$

where $\mu_i \equiv \mathbb{E}(y_i)$ and $y_i \sim$ 'an exponential family distribution'[§]. Here g is a known, monotonic, twice differentiable, link function; $\mathbf{X}_i^*$ is the i^{th} row of a model matrix for any strictly parametric model components, with parameter vector $\boldsymbol{\theta}$; the f_j are smooth functions of the covariates x_j.

To estimate such a model we can specify a basis for each smooth function, along with a corresponding definition of what is meant by smoothness/wiggliness of the function. Starting with the bases, we choose a set of basis functions, b_{ji}, for each function, so that it can be represented as:

$$f_j(x_j) = \sum_{i=1}^{q_j} \beta_{ji} b_{ji}(x_j)$$

where x_j may be a vector quantity and the β_{ji} are coefficients of the smooth, which will need to be estimated as part of model fitting.

Given a basis, it is straightforward to create a model matrix, $\tilde{\mathbf{X}}_\mathbf{j}$, for each smooth. If $\mathbf{f_j}$ is the vector such that $\mathbf{f_j}_i = f_j(x_{ji})$ and $\tilde{\boldsymbol{\beta}}_\mathbf{j} = \left[\beta_{j1}, \beta_{j2}, \ldots, \beta_{jq_j}\right]^{\mathsf{T}}$, then

$$\mathbf{f_j} = \tilde{\mathbf{X}}_\mathbf{j}\tilde{\boldsymbol{\beta}}_\mathbf{j}$$

where $\tilde{\mathbf{X}}_{\mathbf{j},ik} = b_{jk}(x_{ji})$, and the covariate, x_j, may sometimes be a vector quantity.

Typically, (4.9) is not an identifiable model, unless each smooth is subject to a 'centering constraint'. A suitable constraint is that the sum (or mean) of the elements of

[§] although if we take a quasi-likelihood approach we can relax the distributional assumption somewhat and only specify a mean variance relationship for y_i.

$\mathbf{f_j}$ should be zero, which can be written as

$$\mathbf{1}^\mathsf{T}\tilde{\mathbf{X}}_\mathbf{j}\tilde{\boldsymbol{\beta}}_\mathbf{j} = 0.$$

Using the approach taken in section 1.8.1, this constraint can easily be absorbed by re-parameterization. Specifically we find a matrix $\mathbf{Z}$, the $q_j - 1$ columns of which are orthogonal, and which satisfies:

$$\mathbf{1}^\mathsf{T}\tilde{\mathbf{X}}_\mathbf{j}\mathbf{Z} = \mathbf{0}.$$

Now reparameterizing the smooth in terms of $q_j - 1$ new parameters, $\boldsymbol{\beta}_\mathbf{j}$, such that $\tilde{\boldsymbol{\beta}}_\mathbf{j} = \mathbf{Z}\boldsymbol{\beta}_\mathbf{j}$, we obtain a new model matrix for the j^{th} term, $\mathbf{X_j} = \tilde{\mathbf{X}}_\mathbf{j}\mathbf{Z}$, such that $\mathbf{f_j} = \mathbf{X_j}\boldsymbol{\beta}_\mathbf{j}$ automatically satisfies the centering constraint. $\mathbf{Z}$ is never formed explicitly, since it can be represented by a single Householder matrix (see section A.5).

Given centered model matrices, for each smooth term, (4.9) can now be re-written as

$$g(\mu_i) = \mathbf{X}_i\boldsymbol{\beta} \tag{4.10}$$

where $\mathbf{X} = [\mathbf{X}^* : \mathbf{X_1} : \mathbf{X_2} : \cdots]$ and $\boldsymbol{\beta}^\mathsf{T} = [\boldsymbol{\theta}^\mathsf{T}, \boldsymbol{\beta}_\mathbf{1}^\mathsf{T}, \boldsymbol{\beta}_\mathbf{2}^\mathsf{T}, \cdots]$. Clearly (4.10) is just a GLM, and we can therefore write down its likelihood, $l(\boldsymbol{\beta})$, say. Equally clearly, if the q_j are large enough that we have a reasonable chance of accurately representing the unknown f_j's, and $\boldsymbol{\beta}$ is estimated by ordinary likelihood maximization, then there is a good chance of substantially overfitting. For this reason, GAMs are usually estimated by penalized likelihood maximization, where the penalties are designed to suppress overly wiggly estimates of the f_j terms.

The most convenient penalties to work with are those which measure function wiggliness as a quadratic form in the coefficients of the function. For example, the wiggliness of the j^{th} function might be measured by $\tilde{\boldsymbol{\beta}}_\mathbf{j}^\mathsf{T}\tilde{\mathbf{S}}_j\tilde{\boldsymbol{\beta}}_\mathbf{j}$, where $\tilde{\mathbf{S}}_j$ is a matrix of known coefficients. Sometimes $\tilde{\mathbf{S}}_j$ may itself be a weighted sum of simpler matrices of known coefficients, where the weights are parameters to be estimated (see section 4.1.8). The centering reparameterization would convert this penalty to the form $\boldsymbol{\beta}_\mathbf{j}^\mathsf{T}\bar{\mathbf{S}}_j\boldsymbol{\beta}_\mathbf{j}$ where $\bar{\mathbf{S}}_j = \mathbf{Z}^\mathsf{T}\tilde{\mathbf{S}}_j\mathbf{Z}$. Notationally it is convenient to re-write the penalty in terms of the full coefficient vector $\boldsymbol{\beta}$, so that it becomes $\boldsymbol{\beta}^\mathsf{T}\mathbf{S}_j\boldsymbol{\beta}$, where $\mathbf{S}_j$ is just $\bar{\mathbf{S}}_j$ padded with zeroes so that $\boldsymbol{\beta}^\mathsf{T}\mathbf{S}_j\boldsymbol{\beta} \equiv \boldsymbol{\beta}_\mathbf{j}^\mathsf{T}\bar{\mathbf{S}}_j\boldsymbol{\beta}_\mathbf{j}$.

Given a wiggliness measure for each function, we can define a penalized likelihood for the model,

$$l_p(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \frac{1}{2}\sum_j \lambda_j\boldsymbol{\beta}^\mathsf{T}\mathbf{S}_j\boldsymbol{\beta}, \tag{4.11}$$

where the λ_j are smoothing parameters, controlling the tradeoff between goodness of fit of the model and model smoothness. Given values for the λ_j, then l_p is maximized to find $\hat{\boldsymbol{\beta}}$, but the λ_j must themselves be estimated.

4.2.1 Variable coefficient models

Hastie and Tibshirani (1993) proposed a class of models, which they dubbed 'variable coefficient models'. These models are basically GAMs, in which the smooths

may be multiplied by some known covariate. An example is

$$g(\mu_i) = \mathbf{X}^*_i\boldsymbol{\theta} + f_1(x_{1i})x_{2i} + f_2(x_{3i}, x_{4i})x_{5i} + f_3(x_{6i})x_{7i} + \cdots$$

Setting up these models for estimation by penalized regression methods is straightforward. Each row of the model matrix for the smooth is multiplied by the corresponding value of the covariate. For example, the formal expression for the model matrix for the term $f_1(x_{1i})x_{2i}$, in the above example, is simply $\mathrm{diag}(\mathbf{x}_2)\mathbf{X_1}$, where $\mathbf{X_1}$ is the model matrix for $f_1(x_{1i})$, and $\mathrm{diag}(\mathbf{x}_2)$ is a diagonal matrix with x_{2i} at the i^{th} position on its leading diagonal (in the terminology of the `mgcv` package, covered in Chapter 5, variables like x_2 are known as 'by' variables). No other modification of the GAM framework presented in this chapter is necessary.

Note that such models make it easy to condition smooths on factors. For example, if a smooth of x should depend on which of two levels of a factor, a, pertains for a particular response observation, we could write a model as

$$g(\mu_i) = f(x_i)z_{1i} + f(x_i)z_{2i},$$

where z_j is an indicator variable for whether the corresponding factor level is j.

4.3 Justifying P-IRLS

As we saw in chapter 3, the GAM penalized likelihood, (4.11), can be maximized by penalized iteratively re-weighted least squares, and in this section some justification for this approach is provided. For notational compactness (4.11) can be re-written as

$$l_p(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \frac{1}{2}\boldsymbol{\beta}^{\mathsf{T}}\mathbf{S}\boldsymbol{\beta}$$

where $\mathbf{S} = \sum_j \lambda_j \mathbf{S}_j$, and for the moment the λ_j are taken as known. To maximize l_p we set its derivatives with respect to the β_j to zero:

$$\frac{\partial l_p}{\partial \beta_j} = \frac{\partial l}{\partial \beta_j} - [\mathbf{S}\boldsymbol{\beta}]_j = \frac{1}{\phi}\sum_{i=1}^{n} \frac{y_i - \mu_i}{V(\mu_i)}\frac{\partial \mu_i}{\partial \beta_j} - [\mathbf{S}\boldsymbol{\beta}]_j = 0,$$

where $[\cdot]_j$ denotes the j^{th} row of a vector. But by the same argument used in section 2.1.2 these equations are exactly those that would have to be solved to maximize the penalized non-linear least squares problem

$$\mathcal{S}_p = \sum_{i=1}^{n} \frac{(y_i - \mu_i)^2}{\mathrm{var}(Y_i)} + \boldsymbol{\beta}^{\mathsf{T}}\mathbf{S}\boldsymbol{\beta},$$

assuming that the $\mathrm{var}(Y_i)$ terms were known. Again following section 2.1.2 it is easy to show that, in the vicinity of some parameter vector estimate $\hat{\boldsymbol{\beta}}^{[k]}$,

$$\mathcal{S}_p \simeq \left\| \sqrt{\mathbf{W}^{[k]}} \left(\mathbf{z}^{[k]} - \mathbf{X}\boldsymbol{\beta} \right) \right\|^2 + \boldsymbol{\beta}^{\mathsf{T}}\mathbf{S}\boldsymbol{\beta}, \tag{4.12}$$

where, if g is the model link function, $\mathbf{z}^{[k]}$ is a vector of pseudodata and $\mathbf{W}^{[k]}$ is a diagonal matrix with diagonal elements $w_i^{[k]}$ then

$$w_i^{[k]} = \frac{1}{V(\mu_i^{[k]})g'(\mu_i^{[k]})^2} \text{ and } z_i = g(\mu_i^{[k]})(y_i - \mu_i^{[k]}) + \mathbf{X}_i\hat{\boldsymbol{\beta}}^{[k]}.$$

Hence given smoothing parameters, the maximum penalized likelihood estimates, $\hat{\boldsymbol{\beta}}$, are obtained by iterating the steps

1. Given the current $\hat{\boldsymbol{\beta}}^{[k]}$ calculate the pseudodata $\mathbf{z}^{[k]}$ and weights $w_i^{[k]}$.
2. Minimize 4.12 w.r.t. $\boldsymbol{\beta}$ to find $\hat{\boldsymbol{\beta}}^{[k+1]}$. Increment k.

to convergence. See O'Sullivan et al. (1986) for an early reference on penalized likelihood maximization for smooth models.

4.4 Degrees of freedom and residual variance estimation

Before covering $\boldsymbol{\lambda}$ estimation, it is helpful to consider the notion of degrees of freedom for a GAM, and this will also lead on naturally to the question of scale parameter estimation. How many degrees of freedom does a fitted GAM have? Clearly, if the smoothing parameters were all set to zero then the degrees of freedom of the model would be the dimension of $\boldsymbol{\beta}$ (less the number of identifiability constraints). At the opposite extreme, if all the smoothing parameters are very high then the model will be quite inflexible and will hence have very few degrees of freedom. One way of measuring the flexibility of the fitted model is to define the *effective degrees of freedom* as $\mathrm{tr}\,(\mathbf{A})$, by analogy with section 1.3.5. It is fairly easy to show that the maximum of $\mathrm{tr}\,(\mathbf{A})$ is just the number of parameters less the number of constraints, and similarly that the minimum values is $\mathrm{rank}(\sum_i \mathbf{S_i})$ less than this. As the smoothing parameters vary, from zero to infinity, the effective degrees of freedom moves smoothly between these limits.

Now the degrees of freedom of the model are, in effect, reduced by the application of the penalties during fitting, and penalties for different model terms will have different smoothing parameters, and will hence penalize their smooth functions differently. It is therefore natural to want to break the effective degrees of freedom down, into effective degrees of freedom for each smooth. In fact one might as well go further still, and try to ascertain the effective degrees of freedom associated with each $\hat{\beta}_i$, separately. Again this is natural, since the penalties generally penalize each element of $\boldsymbol{\beta}$ differently.

To this end, first define[¶] $\mathbf{P} \equiv (\mathbf{X}^\mathsf{T}\mathbf{X}+\mathbf{S})^{-1}\mathbf{X}^\mathsf{T}$, so that $\hat{\boldsymbol{\beta}} = \mathbf{P}\mathbf{y}$ (in the un-weighted additive model case). Hence $\mathrm{tr}\,(\mathbf{A}) = \mathrm{tr}\,(\mathbf{XP})$. Now define $\mathbf{P}_i^0$ to be $\mathbf{P}$ with all its

[¶] Again writing $\mathbf{S} = \sum_j \lambda_j \mathbf{S}_j$.