21
More general roughness penalties

21.1 Introduction

A theme central to this book has been the use of roughness penalties to incorporate smoothing, whether in the context of using discrete data to define a smooth function in Chapter 5, functional principal components analysis in Chapter 9, or imposing regularity on estimated regression functions in the chapters on the functional linear model.

At the same time, the previous three chapters have dealt with the mathematical properties of linear differential operators L and with techniques for estimating them from data. Principal differential analysis provides a method of estimating low-dimensional functional variation in a sense analogous to principal components analysis, but by estimating an mth order differential operator L rather than a projection.

Moreover, we have seen that by coupling L with a suitable set of constraints on the m linearly independent functions ξ_j satisfying $L\xi_j = 0$, we can *partition* the space of smooth functions into two parts. This is achieved by defining a constraint operator B such that $B\xi_j \neq 0$, and the only function satisfying $Bx = Lx = 0$ is $x = 0$. Then any function x having m derivatives can be expressed uniquely as

$$x = \xi + e \text{ where } L\xi = 0 \text{ and } Be = 0. \tag{21.1}$$

We might call this the *partitioning principle*.

It is time to put these two powerful ideas together, to see what practical value there is in using the partitioning principle to define a roughness

penalty. We want to go beyond the standard practice of defining roughness in terms of $L = D^2$, and even beyond the slightly more general $L = D^m$, to consider what the advantages might be of using an arbitrary operator L, perhaps in conjunction with some constraints captured in the companion operator B. Specifically, when the goal is smoothing the data, we propose using the criterion

$$\texttt{PENSSE}(x) = \sum_j^n [y_j - x(t_j)]^2 + \lambda \times \texttt{PEN}_L(x), \tag{21.2}$$

where

$$\texttt{PEN}_L(x) = \int (Lx)^2(t)\, dt.$$

We begin with some examples.

21.1.1 The lip movement data

Consider the lip movement data introduced in Chapter 19 and plotted in Figure 21.1. We are interested in how these trajectories, all based on observations of a speaker saying "bob," vary from one replication to another. But in the experiment, the syllable was embedded in the phrase, "Say bob again," and it is clear that the lower lip enters and leaves the period during which the syllable is being formed at different heights. This is nuisance variation that we would be happy to eliminate.

Moreover, there was particular interest in the acceleration or second derivative of the lip, suggesting that we should penalize the fourth derivative by spline smoothing with $L = D^4$. Any cubic polynomial trend in the records is ignored if we do that. Now we want to define the *shape* component u and *endpoint* component ξ of each record x in such a way that the behavior of the record at the beginning and end of the interval of observation (normalized to be [0,1]) has minimal impact on the interior and more interesting portion of the curve. One way of achieving this objective is to require the shape components to satisfy the constraints

$$u(0) = Du(0) = 0 \text{ and } u(1) = Du(1) = 0.$$

This means that the constraint is defined by the boundary constraint operator B_B, defined as

$$B_B x = \begin{bmatrix} x(0) \\ Dx(0) \\ x(1) \\ Dx(1) \end{bmatrix}, \tag{21.3}$$

and the shape component u satisfies $B_B u = 0$.

We now have our two linear operators $L = D^4$ and $B = B_B$ in hand, and they are complementary in the sense that $\ker B \cap \ker L = 0$. That is,

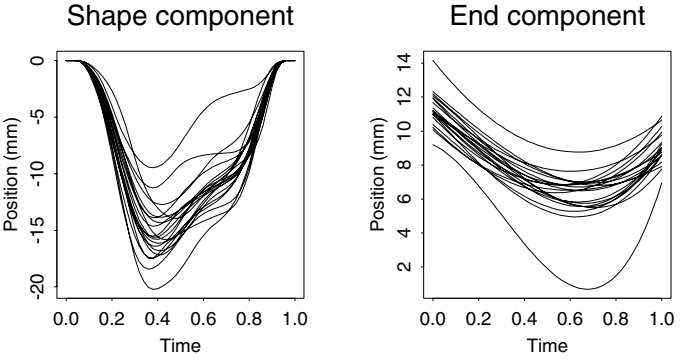

Figure 21.1. The right panel displays the 20 cubic polynomials ξ that match the lip position and derivative values at 0 and 1 for the smoothed versions of the curves in Figure 19.1. The left panel shows the shape components u that have zero endpoint positions and derivatives.

we have now unambiguously split any lip position record x into $x = \xi + u$, where $Bu = 0$, and ξ, a cubic polynomial because $L\xi = D^4\xi = 0$, picks up the endpoint variation by fitting the record's function and derivative values at both 0 and 1. Figure 21.1 displays the endpoint and shape components for all 20 records.

21.1.2 The weather data

We noted in the introduction that a rather large part of the mean daily or monthly temperature curve for any weather station can be captured by the simple function

$$T(t) = c_1 + c_2 \sin(\pi t/6) + c_3 \cos(\pi t/6) \tag{21.4}$$

and the same may be said for the log precipitation profiles. Functions of this form can be annihilated by the operator

$$L = (\pi/6)^2 D + D^3.$$

We could propose smoothing data using the criterion (21.2), where

$$\texttt{PEN}_L(x) = \int (Lx)^2(t)\, dt = \int [(\pi/6)^2 Dx(t) + D^3 x(t]^2(t)\, dt,$$

while paying attention to the periodic character of the data. What would we gain from this? For one thing, as we have already noted in Section 18.4.3, this procedure is likely to have considerable advantages in the estimation of curves x from raw data.

At the same time, the function L`Temp` in this example is interesting in itself, and Ramsay and Dalzell (1991) refer to this as the *harmonic acceleration* of temperature. They show by functional principal components

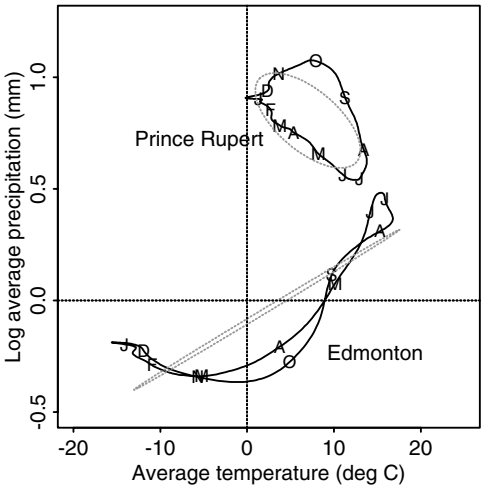

Figure 21.2. The solid cycles are the smoothed daily temperature and log precipitation data, plotted against each other, for two Canadian weather stations. The dotted curves are the estimated cycles based on strictly sinusoidal variation, taking the first three terms of the Fourier expansion of each observed temperature and log precipitation curve. Letters indicate the middle of each month.

and linear regression analyses that $L\texttt{Temp}$, and the harmonic acceleration of log precipitation, contain a great deal of information about the peculiarities of weather at any station. In order to identify the component e uniquely, though, we must choose a matching integral constraint operator B_I, and for this application they chose

$$B_I x = \begin{bmatrix} \int x(t)\,dt \\ \int x(t)\sin(\pi t/6)\,dt \\ \int x(t)\cos(\pi t/6)\,dt \end{bmatrix},$$

corresponding to the first three Fourier coefficients of the observed curves. The three functions ξ_i that span $\ker L$ are then 1, $\sin(\pi t/6)$ and $\cos(\pi t/6)$. Given any curve x, the partition (21.1) is achieved by setting the component $\boldsymbol{\xi}$ to be the first three terms in the Fourier expansion of x.

The solid curves in Figure 21.2 show, for two weather stations, plots of smoothed daily temperature against smoothed daily log precipitation through the year. The shifted sinusoidal components $\xi_j(t)$ for temperature and for log precipitation respectively become ellipses when plotted against each other and yield the dotted curves in the figure.

21.2 The optimal basis for spline smoothing

In Chapter 3 we reviewed the classic technique of representing functions by fitting a basis function expansion to the data. We took pains to point out that not all bases are equal: A good basis has basis functions which mimic the general features that we know apply to the data, such as periodicity, asymptotic linearity, and so on. When we get these features right, we can expect to do a good job with a smaller number of basis functions.

We also pointed out that when the number n of data points is large, computing an expansion in $O(n)$ operations is critical, and in order to achieve this, the basis functions should at least be nonzero only locally, or have compact support. The B-spline basis is especially attractive from this perspective.

In Section 5.6, we extended the basis function expansion concept to employ a partitioned basis (ϕ, ψ) along with a penalty on the size of the component expanded in terms of the basis functions ψ. But two properties, relevance to the data and convenience of computation, remain essential.

We now bring these elements together: Use the partitioning principle to define a set of basis functions that are optimal with respect to smoothing, provide a recipe for an $O(n)$ smoothing algorithm, and also show how these can be put into compact support form to give the appropriate analogue of B-splines. Further details are available in Heckman and Ramsay (2000).

We begin with a theorem that states that the optimal basis for spline smoothing in the context of operators (B, L) is defined by the reproducing kernel k_2 defined in Chapter 20.

Optimal Basis Theorem:

For any $\lambda > 0$, the function x minimizing the spline smoothing criterion (21.2) defined by a linear differential operator L of order m has the expansion

$$x(t) = \sum_{j=1}^{m} d_j \xi_j(t) + \sum_{i=1}^{n} c_i k_2(t_i, t). \tag{21.5}$$

Equation (21.5) can be put a bit more compactly. As before, let $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_m)'$; define another vector function

$$\tilde{k}(t) = \{k_2(t_1, t), k_2(t_2, t), \ldots, k_2(t_n, t)\}'.$$

Then the optimal basis theorem says that the function x has to be of the form $x = \mathbf{d}'\boldsymbol{\xi} + \mathbf{c}'\tilde{k}$, where $\mathbf{d}$ is a vector of m coefficients d_j and $\mathbf{c}$ is the corresponding vector of n coefficients c_i in (21.5). We give a proof of the optimal basis theorem, but as usual any reader prepared to take this on trust should simply skip to the next section.

Proof:

Suppose x^* is any function having square-integrable derivatives up to order m. The strategy for the proof is to construct a function $\tilde{x}$ of the form

(21.5) such that

$$\texttt{PENSSE}(\tilde{x}) \leq \texttt{PENSSE}(x^*)$$

with equality only if $\tilde{x} = x^*$. It then follows at once that we need never look beyond functions of the form (21.5) if we want to minimize the spline smoothing criterion PENSSE.

First of all, write $x^* = u^* + e^*$ where $u^* \in \ker L$ and $e^* \in \ker B$. Let $\mathcal{K}$ be the subspace of $\ker B$ spanned by the n functions $k_2(t_i, \cdot)$, and let $\tilde{e}$ be the projection of e^* onto $\mathcal{K}$ in the L-inner product. This means that $e^* = \tilde{e} + e^\perp$, where

$$\tilde{e} = c'\tilde{k}$$

for some vector c, and the residual $e^\perp$ in $\ker B$ satisfies the orthogonality relation

$$\langle e, e^\perp \rangle_L = \int (Le)(Le^\perp) = 0 \text{ for all } e \text{ in } \mathcal{K}. \tag{21.6}$$

We now define our function $\tilde{x} = u^* + \tilde{e}$, meaning that $\tilde{x}$ is necessarily of the required form (21.5), and $x^* - \tilde{x}$ is equal to the residual $e^\perp$.

To show that $\texttt{PENSSE}(\tilde{x}) \leq \texttt{PENSSE}(x^*)$, note first that, by the defining property of the reproducing kernel, for each i,

$$x^*(t_i) - \tilde{x}(t_i) = e^\perp(t_i) = \langle k_2(t_i, \cdot), e^\perp \rangle_L = 0$$

by property (21.6), since $k_2(t_i, \cdot)$ is of course a member of $\mathcal{K}$ and so is L-orthogonal to $e^\perp$.

Therefore x^* and $\tilde{x}$ agree at the arguments t_i, and so

$$\texttt{PENSSE}(x^*) - \texttt{PENSSE}(\tilde{x}) = \lambda\{\texttt{PEN}_L(x^*) - \texttt{PEN}_L(\tilde{x})\};$$

the residual sum of squares of the y_i is the same about each of the two functions x^* and $\tilde{x}$. Since $Lx^* = Le^*$ and $L\tilde{x} = L\tilde{e}$, we have

$$\begin{aligned} \texttt{PEN}_L(x^*) - \texttt{PEN}_L(\tilde{x}) &= \texttt{PEN}_L(e^*) - \texttt{PEN}_L(\tilde{e}) \\ &= \langle \tilde{e} + e^\perp, \tilde{e} + e^\perp \rangle_L - \langle \tilde{e}, \tilde{e} \rangle_L \\ &= \langle e^\perp, e^\perp \rangle_L + 2\langle \tilde{e}, e^\perp \rangle_L = \langle e^\perp, e^\perp \rangle_L \end{aligned}$$

since $\tilde{e}$ is in $\mathcal{K}$ and is therefore L-orthogonal to $e^\perp$. Therefore $\texttt{PEN}_L(e^*) \geq \texttt{PEN}_L(\tilde{e})$, and consequently $\texttt{PENSSE}(x^*) \geq \texttt{PENSSE}(\tilde{x})$. Equality holds only if $e^\perp \in \ker L$; since we already know that $e^\perp \in \ker B$, this implies that $e^\perp = 0$ and that $x^* = \tilde{x}$. This completes the proof of the theorem.

21.3 An $O(n)$ algorithm for L-spline smoothing

21.3.1 The need for a good algorithm

In principle, the optimal basis theorem should tell us exactly how to proceed. Since we know that the required function is of the form $x = d'u + c'\tilde{k}$,

we need only express $\texttt{PENSSE}(x)$ in terms of c and d and minimize to find the best values of c and d. How would this work out?

Let $\mathbf{K}$ be the matrix with values $k_2(t_i, t_j)$. From equation (20.14) it follows that

$$\texttt{PEN}_L(x) = \langle c'\tilde{k}, c'\tilde{k}\rangle_L = c'\mathbf{K}c.$$

The vector of values $x(t_i)$ is $\mathbf{U}d + \mathbf{K}c$, where $\mathbf{U}$ is the matrix with values $\xi_j(t_i)$. Hence, at least in principle, we can find x by minimizing the quadratic form

$$\texttt{PENSSE}(x) = (y - \mathbf{U}d - \mathbf{K}c)'(y - \mathbf{U}d - \mathbf{K}c) + \lambda c'\mathbf{K}c \tag{21.7}$$

to find the vectors c and d.

Unfortunately the matrix $\mathbf{K}$ is in practice usually extremely badly conditioned, that is to say, the ratio of its largest eigenvalue to its smallest explodes. A practical consequence of this is that the computations required to minimize the quadratic form (21.7) are likely to be unstable or impossible.

Furthermore, in smoothing long sequences of observations, it is critical to devise a smoothing procedure that requires a number of arithmetic operations that does not grow too quickly as the length of the sequence increases. For example, the handwriting data has $n = 1401$ and so an algorithm that was $O(n^2)$ would be impracticable and an $O(n^3)$ algorithm virtually impossible with current computing power. By adopting a somewhat different approach, we can set out an algorithm that requires only $O(n)$ operations, and furthermore avoids the numerical problems inherent in the direct minimization of (21.7).

The algorithm we use is based on the theoretical paper of Anselone and Laurent (1967), but is also known as the Reinsch algorithm because of the application to the cubic polynomial smoothing case ($L = D^2$) by Reinsch (1967, 1970). It was subsequently extended by Hutchison and de Hoog (1985). We do not attempt a full exposition of the rationale for this algorithm here, but Heckman and Ramsay (2000) and Ramsay, Heckman and Silverman (1997) can be consulted for details.

The algorithm requires the computation of values of two types of function that we have already encountered:

1. $\xi_j, j = 1, \ldots, m$: a set of m linearly independent functions satisfying $L\xi_j = 0$, that is, spanning $\ker L$. As before, we refer to these collectively as the vector-valued function $\boldsymbol{\xi}$.
2. k_2: the reproducing kernel function defined in Chapter 18 for the subspace of functions e satisfying $B_I e = 0$, where B_I is the initial value constraint operator.

The functions $\boldsymbol{\xi}$ and k_2 are the user-supplied components of the algorithm and are, of course, defined by the particular choice of operator L used in the smoothing application.

The algorithm splits into three phases:

1. an initial setup phase that does not depend on the smoothing parameter λ
2. a smoothing phase in which we smooth the data
3. a summary phase in which we compute performance measures for the smooth

This division of the task is of practical importance because we may want to try smoothing with many values of λ, and do not want to needlessly repeat either the initial setup phase or the final descriptive phase.

21.3.2 Setting up the smoothing procedure

In the initial phase, we define two symmetric $(n-m)\times(n-m)$ band-structured matrices $\mathbf{H}$ and $\mathbf{C}'\mathbf{C}$ where m is the order of operator L. Both matrices are band-structured with band width at most $2m+1$, which means that all entries more than m positions away from the main diagonal are zero. Because of symmetry, these band-structured matrices require only $(n-m)(m+1)$ storage locations.

We start by explaining how to construct the matrix $\mathbf{C}$. For each $i = 1,\ldots,n-m$, define the $(m+1)\times m$ matrix $\mathbf{U}^{(i)}$ to have (l,j) element $\xi_j(t_{i+l})$, for $l = 0,\ldots,m$. Thus $\mathbf{U}^{(i)}$ is the submatrix of $\mathbf{U}$ consisting only of rows $i, i+1,\ldots,i+l$. Find the QR decomposition (as discussed in Section A.3.3)

$$\mathbf{U}^{(i)} = \mathbf{Q}^{(i)}\mathbf{R}^{(i)},$$

where the matrix $\mathbf{Q}^{(i)}$ is square, of order $m+1$, and orthonormal, and where the matrix $\mathbf{R}^{(i)}$ is $(m+1)\times m$ and upper triangular. Let the vector $c^{(i)}$ be the last column of $\mathbf{Q}^{(i)}$; this vector is orthogonal to all the columns of $\mathbf{U}^{(i)}$. In fact any vector having this property will do, and in special cases the vector can be found by some other method. For polynomial spline smoothing, for instance, coefficients defining divided differences are used.

Now define the $n\times(n-m)$ matrix $\mathbf{C}$ so that its ith column has the $m+1$ values $c^{(i)}$ starting in row i; elsewhere the matrix contains zeroes. In practice, the argument sequence $t_1,\ldots,t_n$ is often equally spaced, and in this case it frequently happens that all the coefficient vectors $c^{(i)}$ are the same, and hence need be computed only once. The band structure of $\mathbf{C}$ immediately implies that $\mathbf{C}'\mathbf{C}$ has the required band structure, and can be found in $O(n)$ operations for fixed m.

The other setup-phase matrix $\mathbf{H}$ is the $(n-m)\times(n-m)$ symmetric matrix

$$\mathbf{H} = \mathbf{C}'\mathbf{K}\mathbf{C}, \tag{21.8}$$

where $\mathbf{K}$ is the matrix of values $k_2(t_i, t_j)$. It turns out that $\mathbf{H}$ is also band-structured with band width $2m-1$. This is a consequence of the expression (20.15) for the reproducing kernel, which yields the following two-part expression:

$$k_2(t_i, t) = \begin{cases} u(t_i)'\mathbf{F}(t)u(t) & \text{for } t_i \geq t \\ u(t_i)'\mathbf{F}(t_i)u(t) & \text{for } t_i \leq t, \end{cases} \tag{21.9}$$

for a certain matrix function $\mathbf{F}(t)$. This in turn implies that

$$\mathbf{K}_{ij} = \{\mathbf{U}\mathbf{F}(t_j)u(t_j)\}_i \quad \text{for } i \geq j. \tag{21.10}$$

Suppose $k \geq j$. Because $\mathbf{C}_{ik}$ is zero for $i < k$,

$$(\mathbf{C}'\mathbf{K})_{kj} = \sum_{i=k}^{n} \mathbf{C}_{ik}\mathbf{K}_{ij} = \sum_{i=k}^{n} \mathbf{C}_{ik}\{\mathbf{U}\mathbf{F}(t_j)u(t_j)\}_i,$$

substituting (21.10); notice that $i \geq j$ for all i within the range of summation $k \leq i \leq n$. It follows that for $k \geq j$ we have

$$(\mathbf{C}'\mathbf{K})_{kj} = \{\mathbf{C}'\mathbf{U}\mathbf{F}(t_j)u(t_j)\}_i = 0.$$

So $\mathbf{C}'\mathbf{K}$ is strictly upper-triangular. Because of the band structure of $\mathbf{C}$ this means that the matrix $\mathbf{H} = (\mathbf{C}'\mathbf{K})\mathbf{C}$ has zero entries for positions m or more below the main diagonal, and by symmetry $\mathbf{H}$ has the stated band structure.

21.3.3 The smoothing phase

The actual smoothing consists of two steps:

1. Compute the vector z, of length $n-m$, that solves

$$(\mathbf{H} + \lambda\mathbf{C}'\mathbf{C})z = \mathbf{C}'y, \tag{21.11}$$

   where the vector y contains the values to be smoothed.

2. Compute the vector of n values $\hat{y}_i = x(t_i)$ of the smoothing function x at the n argument values using

$$\hat{y} = y - \lambda\mathbf{C}z. \tag{21.12}$$

Because of the band structure of $(\mathbf{H} + \lambda\mathbf{C}'\mathbf{C})$ and of $\mathbf{C}$, both of these steps can be computed in $O(n)$ operators, and references on efficient matrix computation such as Golub and van Loan (1989) can be consulted for details.

21.3.4 The performance assessment phase

The vector of smoothed values $\hat{y}$ and the original values y that were smoothed are related as follows:

$$\hat{y} \quad = \quad y - \lambda\mathbf{C}(\mathbf{H} + \lambda\mathbf{C}'\mathbf{C})^{-1}\mathbf{C}'y$$

$$= \quad \{\mathbf{I} - \lambda\mathbf{C}(\mathbf{H} + \lambda\mathbf{C}'\mathbf{C})^{-1}\mathbf{C}'\}y. \tag{21.13}$$

The matrix $\mathbf{S}$ defined by

$$\mathbf{S} = \mathbf{I} - \lambda\mathbf{C}(\mathbf{H} + \lambda\mathbf{C}'\mathbf{C})^{-1}\mathbf{C}' \tag{21.14}$$

is often called the *hat matrix*, and in effect defines a linear transformation that maps the unsmoothed data into its smooth image by

$$\hat{y} = \mathbf{S}y.$$

Various measures of performance depend on the diagonal values in $\mathbf{S}$. Of these, the most popular are currently

$$\texttt{GCV} = \texttt{SSE}/(1 - n^{-1}\text{trace}\,\mathbf{S})^2, \tag{21.15}$$

where

$$\texttt{SSE} = \sum_{i=1}^{n}[y_i - x(t_i)]^2 = \|y - \hat{y}\|^2$$

and

$$\texttt{CV} = \sum_{i=1}^{n}[\{y_i - x(t_i)\}/\{1 - s_{ii}\}]^2, \tag{21.16}$$

where s_{ii} is the ith diagonal entry of $\mathbf{S}$. We can compute both measures GCV and CV in $O(n)$ operations given the band-structured nature of the matrices defining $\mathbf{S}$, using methods developed by Hutchison and de Hoog (1985).

One of the main applications of these two criteria, both of which are types of discounted error sums of squares, is as a guide for choosing the value of the smoothing parameter λ. It is relatively standard practice to look for the value that minimizes one of these two criteria, just as various variable selection procedures attempt to minimize discounted error sums of squares in standard regression analysis. Interestingly, the GCV measure was originally introduced by Craven and Wahba (1979) as an approximation to the CV criterion that could be computed in $O(n)$ operations; now CV is also available in n operations, but GCV still tends to be preferred in practice for other reasons. For example, various simulation studies have indicated that GCV tends to be a better basis for choosing the smoothing parameter λ, possibly because GCV makes use of smoothing itself by replacing the variable values $1 - s_{ii}$ by the average $1 - n^{-1}\text{trace}\,\mathbf{S}$.

Also of great value is a measure of the effective number of degrees of freedom of the smoothing operation. Two measures are

$$\texttt{DF}_1 = \text{trace}\,\mathbf{S} \text{ and } \texttt{DF}_2 = \text{trace}\,\mathbf{S}'\mathbf{S} = \text{trace}\,\mathbf{S}^2. \tag{21.17}$$

These dimensionality measures were introduced and discussed by Buja et al. (1989). It can be shown that in the limit as $\lambda \to \infty$, both measures become simply m, and similarly, as $\lambda \to 0$, both measures converge to

n. In between, they give slightly different impressions of how much of the variation in the original unsmoothed data remains in the smoothed version.

21.3.5 Other $O(n)$ algorithms

There is an intimate connection between the theory of splines and the theory of stochastic differential equations (Wahba, 1978, 1990, Weinert, Bird and Sidhu, 1980). This leads to the possibility of using the Kalman filter, a technique widely used in engineering and other fields to extract an estimate of a signal from noisy data, to compute a smoothing spline. Ansley, Kohn and Wong (1993), using a Kalman filtering algorithm described in Ansley and Kohn (1989), give some examples of computing an L-spline in $O(n)$ operations. However, except for fairly simple cases, this algorithm appears to be difficult to implement, and its description involves substantial mathematical detail. Nevertheless, we feel that it is important to call the reader's attention to this stimulating literature on smoothing by state-space methods.

21.4 A compact support basis for L-splines

In this section our concern is the construction of compact support basis functions from the reproducing kernel basis functions $k_2(t_i, \cdot)$. A basis made up of such functions may, for example, be useful for techniques such as the regularized principal components analysis described in Section 9.4.1, and has many numerical advantages, analogous to those of B-splines.

For any fixed $i = 1, \ldots, n-2m$, consider the sequence of $2m+1$ basis functions based on the reproducing kernel:

$$k_2(t_{i+\ell}, \cdot), \ell = 0, \ldots, 2m.$$

Let $b_\ell^{(i)}, \ell = 0, \ldots, 2m$ be a corresponding sequence of weights defining a new basis function

$$\psi_i = \sum_{\ell=0}^{2m} b_\ell^{(i)} k_2(t_{i+\ell}, \cdot). \tag{21.18}$$

The properties we are seeking are

$$\psi_i(t) = 0, t \le t_i \text{ and } \psi_i(t) = 0, t \ge t_{i+2m}.$$

But from the first line of (21.9), we see the first of these is achieved if

$$\sum_{\ell=0}^{2m} b_\ell^{(i)} \xi_{j_1}(t_{i+\ell}) = 0, \ j_1 = 1, \ldots, m \tag{21.19}$$

and at the same time the second line of (21.9) indicates that the second property is satisfied if

$$\sum_{\ell=0}^{2m} b_\ell^{(i)} [\sum_{j_1=1}^{m} \xi_{j_1}(t_{i+\ell}) f_{j_1,j_2}(t_{i+\ell})] = 0, \ j_2 = 1, \ldots, m, \tag{21.20}$$

where $f_{j_1,j_2}(t_{i+\ell})$ is entry (j_1, j_2) of $\mathbf{F}(t_{i+\ell})$.

Now these are two sets of m linear constraints on the $2m+1$ coefficients $b_\ell^{(i)}$, and we know that in general we can always find a coefficient vector $b^{(i)}$ that satisfies them. The reason that there are only $2m$ constraints for $2m+1$ coefficients is that the linear constraints can only define the vector $b^{(i)}$ up to a change of scale.

Let the $(2m+1) \times 2m$ matrix $\mathbf{V}^{(i)}$ have in its first m columns the values $\xi_{j_1}(t_{i+\ell}), j_1 = 1, \ldots, m$ and in its second set of m columns the values $\sum_{j_1=1}^{m} \xi_{j_1}(t_{i+\ell}) f_{j_1,j_2}(t_{i+\ell}), j_2 = 1, \ldots, m$. Then the constraints (21.19) and (21.20) can be written in the matrix form

$$(b^{(i)})' \mathbf{V}^{(i)} = 0.$$

As in the calculation of the vectors $c^{(i)}$ in Section 21.3.2, the required vector $b^{(i)}$ is simply the last column of the $\mathbf{Q}$ matrix in the QR decomposition of $\mathbf{V}^{(i)}$.

If the argument values are unequally spaced, this calculation of the coefficient vectors $b^{(i)}$ must be carried for each value of i from 1 to $n-2m$. However, in the frequently encountered case where the t_i values are equally spaced, only one coefficient calculation is required, and the resulting set of coefficients b serves for all $n-2m$ compact support splines ψ_i.

Observant readers may note that we have lost $2m$ basis functions by this approach. We may deal with this difficulty in various ways. One approach is to say that a little bit of fitting power has been lost, but that if n is large, this may have little impact on the smoothing function, and what little impact it has is at the ends of the interval. Alternatively, however, we can use a technique employed in defining polynomial B-splines, and add m additional argument values at each end of the interval. For computational convenience in the equally spaced argument case, we can make these simply a continuation of the sequence in both directions. This augments the basis in order to retain the full fitting power of the original reproducing kernel basis.

21.5 Some case studies

21.5.1 The gross domestic product data

The gross domestic product data introduced in Chapter 18 share with many economic indicators the overall tendency for exponential growth. If we wish

to smooth the de-seasonalized GDP record of the United States displayed in Figure 18.3, the operator $L = -\gamma D + D^2$ annihilates $\xi_1(t) = 1$ and $\xi_2(t) = e^{\gamma t}$, so these are obvious choices for the functions spanning $\ker L$. A reasonable choice for the matching constraint operator is simply B_I, such that $B_I u = \{u(0), Du(0)\}'$, implying that the coefficients of ξ_1 and ξ_2 are specified by the initial value and slope of the smoothed record.

In this case, we might decide to estimate parameter γ by estimating the slope of the relationship between log GDP and time by ordinary regression analysis. Another possibility is to fit all or part of the data by nonlinear least squares regression using the two functions ξ_1 and ξ_2. That is, we minimize the error sum of squares with respect to the coefficients c_1 and c_2 of $c_1\xi_1 + c_2\xi_2$ and with respect to γ which, of course, determines ξ_2. Since for any fixed γ value, the minimizing values of the coefficients can be computed directly by linear least squares, it makes sense to use a one-dimensional function minimizing routine such as Brent's method (Press et al. 1992) to find the optimal γ value; each new value of γ within the iterative method implies a linear regression to get the associated values of c_1 and c_2. The resulting least squares estimate of γ for the U.S. data, based on the values from 1980 to 1989, when the growth was more exponential, is 0.054.

Using this value of γ, we used the method of Section 21.3 to find the L-smoothing spline shown in Figure 21.3. We minimized the GCV criterion to obtain the value $\lambda = 0.053$. The $\mathtt{DF}_1$ measure of equivalent degrees of freedom was 39.6, so we purchased the excellent fit of the spline at the price of a rather large number of degrees of freedom.

By comparison, the cubic smoothing spline that minimizes GCV produces almost identical results in terms of GCV and $\mathtt{DF}_1$ values. This is perhaps not too surprising since the curve is only slightly more exponential than linear. But the results are rather different when we smooth with the fixed value of $\mathtt{DF}_1 = 10$, corresponding to $\lambda = 22.9$. The L-spline fit using this more economical model is just barely visible in Figure 21.3, and $\mathtt{GCV} = 0.00068$. The cubic polynomial spline with $\mathtt{DF}_1 = 10$ yields $\mathtt{GCV} = 0.00084$, and its poorer fit reflects the fact that some of its precious degrees of freedom were used up in fitting the mild exponential trend.

21.5.2 The melanoma data

These data, displayed in Figure 17.5, represent a more complex relationship, with a cyclic effect superimposed on a linear development. The interesting operator is

$$L = \omega^2 D^2 + D^4 \tag{21.21}$$

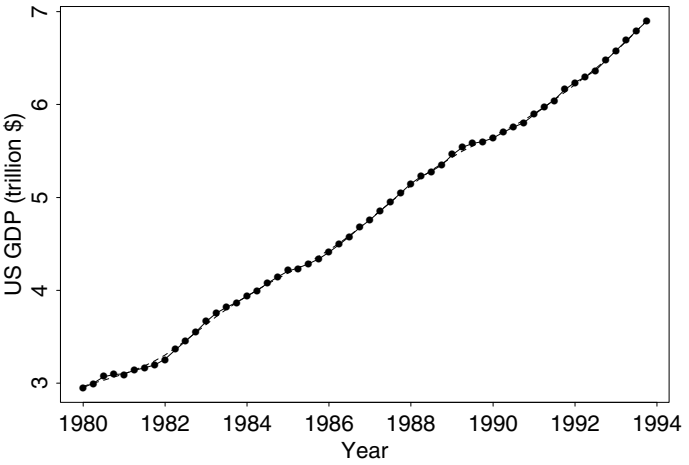

Figure 21.3. The line indicates the spline smooth of the U.S. GDP data using $L = -0.054D + D^2$ and the minimum GCV value for smoothing parameter λ. The dashed line indicates the L-spline fit corresponding to $\mathtt{DF}_1 = 10$.

for some appropriate constant ω, since this would annihilate the four functions

$$u(t) = (1, t, \sin \omega t, \cos \omega t)'.$$

Using the techniques of Chapter 18, the reproducing kernel is

$$\begin{aligned} k_2(s,t) \quad = \quad & \omega^{-7}[(\omega s)^2(\omega t/2 - \omega s/6) - \omega t + \omega s + \omega t \cos \omega s \\ & + \omega s \cos \omega t + \sin \omega s - \sin \omega t + \sin(\omega t - \omega s) \\ & - (\sin \omega s \cos \omega t)/2 + s \cos(\omega t - \omega s)/2], \\ & s \le t. \end{aligned} \tag{21.22}$$

We estimated the parameter ω to be 0.650 by the nonlinear least squares approach. This corresponds to a period of 9.66 years, roughly the period of the sunspot cycle affecting solar radiation and consequently melanoma. When we smooth the data with the spline defined by the operator (21.21) and select λ so as to minimize GCV, it turns out that λ becomes arbitrarily large, corresponding to a parametric fit using only the basis functions $\boldsymbol{\xi}$, consuming four degrees of freedom, and yielding $\mathtt{GCV} = 0.076$. The polynomial smoothing spline with order $m = 4$, displayed in Figure 17.5, is a minimum-GCV estimate corresponding to $\mathtt{DF}_1 = 12.0$ and $\mathtt{GCV} = 0.095$. Thus, polynomial spline smoothing required three times the degrees of freedom to produce a fit that was still worse in GCV terms than the L-spline

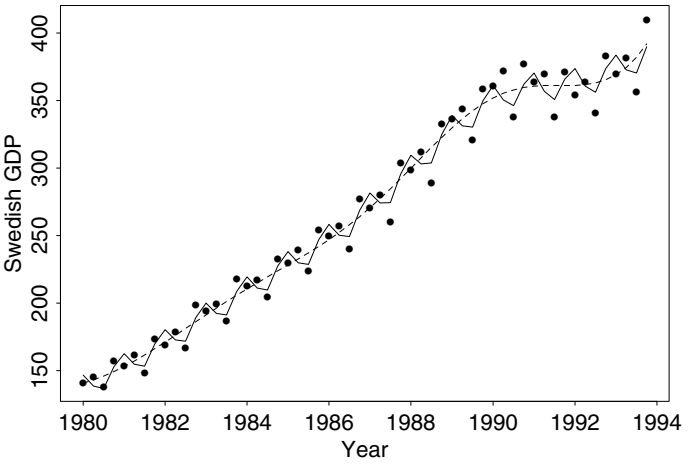

Figure 21.4. The gross domestic product for Sweden with seasonal variation. The solid line is the smooth using operator $L = (-\gamma D+D^2)(\omega^2 I+D^2)$, and the dashed line is the smooth for $L = D^4$, the smoothing parameter being determined by minimizing the GCV criterion in both cases.

smooth. Of the two order-4 methods, the operator (21.21) is much to be preferred to $L = D^4$.

21.5.3 The GDP data with seasonal effects

In the data provided by the U.S. and most other countries, the within-year variation in GDP that is a normal aspect of most economies was removed. But the data for Sweden, displayed in Figure 21.4, does retain this seasonal variation. This suggests composing the operator $-\gamma D + D^2$ used for the U.S. GDP data with the de-seasonalizing operator $\omega^2 I + D^2$ to obtain the composite operator of order four

$$L = (-\gamma D + D^2)(\omega^2 I + D^2) = -\gamma\omega^2 D + \omega^2 D^2 - \gamma D^3 + D^4. \quad (21.23)$$

This annihilates the four linearly independent functions given by the components of

$$u(t) = (1, \exp\gamma t, \sin\omega t, \cos\omega t)'.$$

In this application we know that $\omega = 2\pi$ for time measured in years, and the nonlinear least squares estimate for γ was 0.078.

The minimum GCV L-spline for these data is the solid line in the figure, and corresponds to $\texttt{GCV} = 142.9, \texttt{SSE} = 5298$, and $\texttt{DF}_1 = 10.4$. This fairly low-dimensional spline tracks both the seasonal and long-term

variation rather well. By contrast, the minimum GCV polynomial spline corresponding to $L = D^4$ is shown by the dashed line, and corresponds to $\texttt{GCV} = 193.8, \texttt{SSE} = 8169$, and $\texttt{DF}_1 = 7.4$. As both the curve itself and the GCV value indicate, the polynomial spline was completely unable to model the seasonal variation, and treated it as noise. On the other hand, reducing the smoothing parameter λ to the point where SSE was reduced to the same value as was attained for the L-spline required $\texttt{DF}_1 = 28.2$, or nearly three times the degrees of freedom. Again we see that building the capacity to model important sources of variation into the operator L pays off handsomely.

21.5.4 Smoothing simulated human growth data

One of the triumphs of nonparametric regression techniques has been their capacity to uncover previously unsuspected aspects of growth in skeletal height (Gasser, Müller, Köhler, Molinari and Prader, 1984; Ramsay, Bock and Gasser, 1995). In this illustration, spline smoothing using an estimated differential operator was applied to simulated smoothing data. The objective was to see whether estimating the smoothing operator improves the estimation of the height and height acceleration growth functions over an a priori "off-the-rack" smoother.

To investigate how the performance of the L-spline would compare with a polynomial spline in practice, we simulated data to resemble as much as possible actual human growth curve records. We generated two samples: a training sample of 100 records that was analyzed in a manner representative of actual practice, and a validation sample of 1000 records to see how these analyses would perform on data for which the analyses were not tuned.

The simulated data for both the training and validation samples consisted of growth records generated by using the triple logistic parametric nine-parameter growth model proposed by Bock and Thissen (1980). According to this model, height $h_i(t)$ at age t for individual i is

$$h_i(t) = \sum_{j=1}^{3} c_{ij}/[1 + \exp(-a_{ij}(t - b_{ij}))]. \tag{21.24}$$

This model, although not completely adequate to account for actual growth curves, does capture their salient features rather well. The actual number of parameters in the model turns out to be only eight, since parameter $a_{i,1}$ can be expressed as a function of the other parameters.

We generated each record by first sampling from a population of coefficient vectors having a random distribution estimated from actual data for males in the Fels Growth Study (Roche, 1992). We computed the errorless growth curves (in cm) for the 41 age values ranging from 1 to 21 in half-yearly steps, and generated the simulated data by adding independent normal error with mean 0 and standard deviation 0.5 to these values.

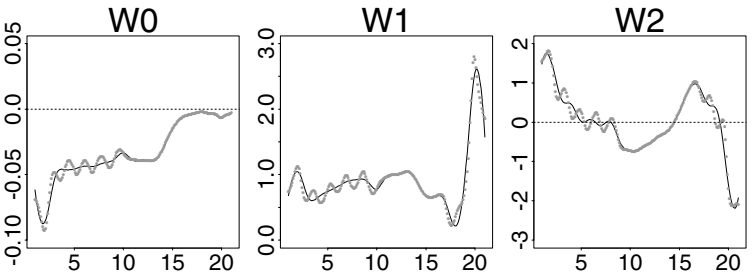

Figure 21.5. The three weight functions w_0, w_1, and w_2 for the operator $L = w_0 I + w_1 D + w_2 D^2 + D^3$: The points indicate the point-wise-approximation, and the solid line indicates the basis function expansion.

These simulated data had roughly the same variability as actual growth measurements.

The first step was to use the training sample to estimate the order three L-spline that comes as near as possible to annihilating the curves. To this end, the first analysis consisted of polynomial spline smoothing of the simulated data to get estimates of the first three derivatives. The smoothing operator used for this purpose was D^5, implying that the smoothing splines were piecewise polynomials of degree 9. This permitted us to control the roughness of the third derivative in much the same way as a cubic smoothing spline controls the roughness of the smoothing function itself. The smoothing parameter was chosen to minimize the `GCV` criterion, and with this amount of replicated data, this value of its minimum is sharply defined. Since our principal differential analysis estimate of the operator L required numerical integration, we also obtained function and derivative estimates at 201 equally-spaced values 1(.1)21.

We estimated a third-order differential operator L using both the point-wise technique and the basis function expansion approach described in Chapter 19. For the latter approach, we used the 23 order 4 B-splines defined by positioning knots at the integer values of age. Figure 21.5 displays the estimated weight functions w_0, w_1, and w_2 for the operator $L = w_0 I + w_1 D + w_2 D^2 + D^3$. Although these are difficult to interpret, we can see that w_0 is close to 0, suggesting that the operator could be simplified by dropping the first term. On the other hand, w_1 is close to one until the age of 15 when the growth function has strong curvature as the pubertal growth spurt ends, and its strong variation after 15 helps the operator to deal with this pronounced curvilinearity. The acceleration weight w_2 varies substantially over the whole range of ages.

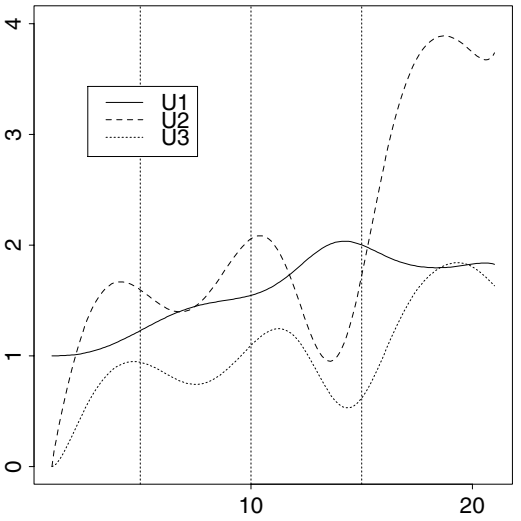

Figure 21.6. The three solutions to the homogeneous equation $Lu = 0$ corresponding to the linear differential operator L estimated for the simulated human growth data.

Figure 21.6 shows three linearly independent solutions ξ_j to $Lu = 0$. Linear combinations of these three functions can produce good approximations to actual growth curves.

The next step was to use the estimated functions ξ_j and the techniques of Chapter 18 to estimate the Green's function G and the reproducing kernel k_2 associated with this operator. We approximated the integrals involved using the trapezoidal rule applied to the values at the 201 argument values.

Now we were ready to carry out the actual smoothing of the training sample data by using the two techniques, L-spline and polynomial spline smoothing, both of order three, just much as one would in practice. For both techniques, we relied on the GCV criterion to choose the smoothing parameter. The polynomial smooth gave values of $\texttt{GCV}, \texttt{DF}_1$ and λ of 487.9, 9.0 and 4.4, respectively, and the L-spline smooth produced corresponding values of 348.2, 11.2 and 0.63.

How well would these two smoothing techniques approximate the curves generating the data? To answer this question, we generated 1,000 new simulated curves using the same generation process, and applied these two smoothers using the training sample values of λ. Since we knew the values of the true curves, we could estimate the root-mean-squared error criterion

$$\texttt{RMSE}(t) = \sqrt{\mathrm{E}\{\hat{x}(t) - x(t)\}^2}$$

by averaging the squared error across the 1,000 curves for a given specific age t, and then taking the square root. This yielded the two RMSE curves

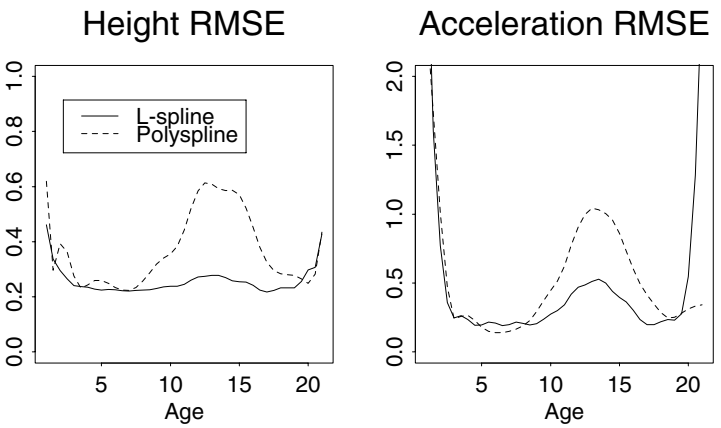

Figure 21.7. The left panel displays root-mean-squared error (RMSE) as a function of age for the simulated growth data. The solid line is for smoothing using the estimated differential operator L, and the dashed line is for polynomial smoothing using $L = D^3$. The right panel shows these results for the estimated height acceleration.

displayed in Figure 21.7. We see that the estimate of both the growth curve itself and its acceleration by the L-spline procedure is much better for all but the final adult period, where the L-spline estimate of the acceleration curve becomes rather noisy and unstable. The improvement in the estimation of both curves is especially impressive prior to and during the pubertal growth spurt: The mean square error for the polynomial smooth is about four times that of the L-spline smooth. That is, using the L-spline is roughly equivalent to using the polynomial smooth with quadruple the sample size. same generation process, and applied these two smoothers using the training sample values of λ. Since we knew the values of the true curves, we could estimate the root-mean-squared error criterion

$$\texttt{RMSE}(t) = \sqrt{\mathrm{E}\{\hat{x}(t) - x(t)\}^2}$$

by averaging the squared error across the 1,000 curves for a given specific age t, and then taking the square root. This yielded the two `RMSE` curves displayed in Figure 21.7. We see that the estimate of both the growth curve itself and its acceleration by the L-spline procedure is much better for all but the final adult period, where the L-spline estimate of the acceleration curve becomes rather noisy and unstable. The improvement in the estimation of both curves is especially impressive prior to and during the pubertal growth spurt: The mean square error for the polynomial smooth is about four times that of the L-spline smooth. That is, using the L-spline is roughly equivalent to using the polynomial smooth with quadruple the sample size.