5

Smoothing functional data with a roughness penalty

5.1 Introduction

We saw in Chapter 4 that basis expansions can provide good approximations to functional data provided that the basis functions have the same essential characteristics as the process generating the data. Thus, a Fourier basis is useful if the functions we observe are periodic and do not exhibit fluctuations in any particular interval that are much more rapid than those elsewhere. However, fitting basis expansions by least squares implies clumsy discontinuous control over the degree of smoothing, and we wonder if it is not possible to get better results with other methods.

Kernel smoothing and local polynomial fitting techniques, on the other hand, are based on appealing, efficient and easily understood algorithms that are fairly simple modifications of classic statistical techniques. They offer continuous control of the smoothness of the approximation, but they are seldom optimal solutions to an explicit statistical problem, such as minimizing a measure of total squared error, and their rather heuristic character makes extending them to other smoothing situations difficult.

In this chapter we introduce a more powerful option for approximating discrete data by a function. The *roughness penalty* or *regularization* approach retains the advantages of the basis function and local expansion smoothing techniques developed in Chapter 4, but circumvents some of their limitations. More importantly, it adapts gracefully to the more general functional data analysis problems that we consider in subsequent chapters. Finally, it often produces better results, and especially in the estimation of

derivatives. This roughness penalty approach will be our smoothing method of choice throughout this book.

Like the least squares methods of Chapter 4, roughness penalty methods are based on optimizing a fitting criterion that defines what a smooth of the data is trying to achieve. But here the precise meaning of "smooth" is expressed explicitly at the level of the criterion being optimized, rather than implicitly in terms of the number of basis functions being used. Moreover, roughness penalty approaches can be applied to a much wider range of smoothing problems than simply estimating a curve x from observations of $x(t_j)$ for certain points t_j. Green and Silverman (1994) discuss a variety of statistical problems that can be approached using roughness penalties, including those where the data's dependence on the underlying curve is akin to the dependence on parameters in generalized linear models. Here we extend still further the scope of roughness penalty methods by discussing various functional data analysis contexts where roughness penalties are an elegant way to introduce smoothing into the analysis.

Figure 5.1 shows what we are trying to achieve. The refinery data from the top panel of Figure 1.4 show measurements that seem flat up to time 67, followed by a sharp upward turn and then an smooth approach toward a new level. In Chapter 17 we will want to model the change or derivative of this trend. A good estimate should show near zero derivative to time 64, an abrupt increase to a maximum value, and then an approximately exponential decay thereafter. Three estimates of this derivative computed by penalizing the roughness of the derivative are shown in the Figure. The best of these seems to be the heavy line, which combines a near zero value on the left with the abrupt upward turn, high peak value, and fairly smooth decay that we want. The smoother of the other two curves fails at both the upward turn and at the peak, and the other is too wild below time 50.

5.2 Spline smoothing

Let us consider how regularization works in the simplest functional case when the goal is to estimate a non-periodic function x on the basis of discrete and noisy observations in a vector $\mathbf{y}$. We continue with the data-smoothing problem described in Chapter 4. However, we will reserve the term "spline smoothing" for using roughness penalties in the way described in this section. By contrast, the smoothing literature often refers to the least squares fitting of B-spline expansions that we described in Chapter 4 as "regression spline smoothing."

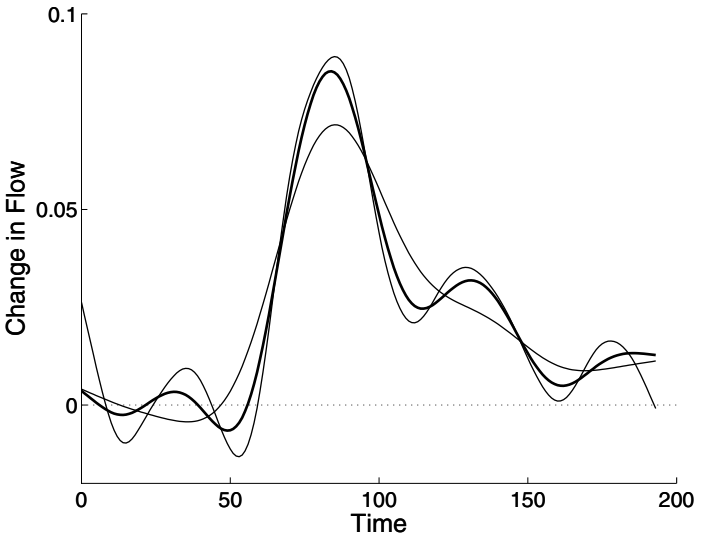

Figure 5.1. Three estimates of the rate of change or first derivative of the data shown in the top panel of Figure 1.4. Each curve has its roughness penalized.

5.2.1 Two competing objectives in function estimation

The spline smoothing method estimates a curve x from observations $y_j = x(t_j) + \epsilon_j$ by making explicit two conflicting goals in curve estimation. On the one hand, we wish to ensure that the estimated curve gives a good fit to the data, for example in terms of the residual sum of squares $\sum[y_j - x(t_j)]^2$. On the other hand, we do not wish the fit to be too good if this results in a curve x that is excessively "wiggly" or locally variable.

These competing aims correspond to the elements of the basic principle of statistics, discussed in Section 4.5,

$$\text{Mean squared error} = \text{Bias}^2 + \text{Sampling variance},$$

where bias, sampling variance and mean squared error were defined in Section 4.5.1. A completely unbiased estimate of the function value $x(t_j)$ can be produced by a curve fitting y_j exactly, since this observed value is itself an unbiased estimate of $x(t_j)$ according to our error model. But any such curve must have high variance, manifested in the rapid local variation of the curve.

In spline smoothing, as in other smoothing methods, the mean squared error, usually abbreviated MSE, is one way of capturing what we usually mean by the quality of estimate. We noted in Section 4.5.1 that other loss functions may be preferable in certain situations, but that the notion of a trade-off between bias and sampling variance applies more widely to these situations as well, although not with this exact decomposition.

MSE can often be dramatically reduced by sacrificing some bias in order to reduce sampling variance, and this is a key reason for imposing smoothness on the estimated curve. By requiring that the estimate vary only gently from one value to another, we are effectively "borrowing information" from neighboring data values, thereby expressing our faith in the regularity of the underlying function x that we are trying to estimate. This pooling of information is what makes our estimated curve more stable, at the cost of some increase in bias. The roughness penalty makes explicit what we sacrifice in bias to achieve an improvement MSE or some other loss function.

5.2.2 Quantifying roughness

Here is popular way to quantify the notion "roughness" of a function. The square of the second derivative $[D^2x(t)]^2$ of a function at t is often called its *curvature* at t, since a straight line, which we all agree has no curvature, also has a zero second derivative. Consequently, a natural measure of a function's roughness is the integrated squared second derivative

$$\texttt{PEN}_2(x) = \int [D^2x(s)]^2\, ds\ . \tag{5.1}$$

Highly variable functions can be expected to yield high values of $\texttt{PEN}_2(x)$ because their second derivatives are large over at least some of the range of interest. For example, consider the two acceleration curves displayed in Figure 4.3, the estimated and actual growth acceleration according to the Jolicoeur model. The values of $\texttt{PEN}_2(x)$ for these curves are 0.22 and 1.42, respectively, indicating the the estimated acceleration curve is substantially rougher than the true curve.

Of course, since these curves are themselves second derivatives, these values are actually the values of

$$\texttt{PEN}_4(x) = \int [D^4x(s)]^2\, ds,$$

where x is a height function. This suggests that we may need to generalize the roughness penalty (5.1) by allowing a derivative $D^m x$ of arbitrary order so as to work with the penalty

$$\texttt{PEN}_m(x) = \int [D^m x(s)]^2\, ds\ . \tag{5.2}$$

5.2.3 The penalized sum of squared errors fitting criterion

We now need to modify the last squares fitting criterion (4.5), defined in Chapter 4, so as to allow the roughness penalty $\texttt{PEN}_2(x)$ to play a role in defining $x(s)$. Let $x(\mathbf{t})$ be the vector resulting from function x being

evaluated at the vector $\mathbf{t}$ of argument values. We define a compromise that explicitly trades off smoothness against data fit by defining the *penalized* residual sum of squares as

$$\texttt{PENSSE}_\lambda(x|\mathbf{y}) = [\mathbf{y} - x(\mathbf{t})]'\mathbf{W}[\mathbf{y} - x(\mathbf{t})]^2 + \lambda \times \texttt{PEN}_2(x) \ , \tag{5.3}$$

Our estimate of the function is obtained by finding the function x that minimizes $\texttt{PENSSE}_\lambda(x)$ over the space of functions x for which $\texttt{PEN}_2(x)$ is defined.

The parameter λ is a *smoothing parameter* that measures the rate of exchange between fit to the data, as measured by the residual sum of squares in the first term, and variability of the function x, as quantified by $\texttt{PEN}_2(x)$ in the second term. As λ becomes larger and larger, functions which are not linear must incur a more and more substantial roughness penalty through the term $\texttt{PEN}_2(x)$, and consequently the composite criterion $\texttt{PENSSE}_\lambda(x)$ must place more and more emphasis on the smoothness of x and less and less on fitting the data. For this reason, as $\lambda \to \infty$ the fitted curve x must approach the standard linear regression to the observed data, where $\texttt{PEN}_2(x) = 0$.

On the other hand, for small λ the curve tends to become more and more variable since there is less and less penalty placed on its roughness, and as $\lambda \to 0$ the curve x approaches an *interpolant* to the data, satisfying $x(t_j) = y_j$ for all j. However, even in this limiting case the interpolating curve is not arbitrarily variable; instead, it is the smoothest twice-differentiable curve that exactly fits the data.

5.2.4 The structure of a smoothing spline

Suppose for the moment that we make no assumptions about function x except that it has a second derivative.[1] We also assume here that sampling points $t_j, j = 1, \ldots, n$ are distinct. What kind of function minimizes this penalized error sum of squares?

A remarkable theorem, found in de Boor (2002) and other more advanced texts on smoothing, states that the curve x that minimizes $\texttt{PENSSE}_\lambda(x|y)$ is a cubic spline with knots at the data points t_j. Note that we have not here assumed anything about how x is constructed; the spline structure of x is a consequence of this theorem, in which an objective function is optimized with respect to an entire function. Solutions to problems that involve optimizing with respect to functions rather than with respect to parameters are called *variational problems.*

Placing knots at data points eliminates one of the issues in the use of regression splines: where to place the knots. Smoothing splines adapt nat-

[1] More technically, a slightly weaker assumption is required: that the integral of the squared second derivative is finite.

urally to unequal spacing of sampling points, and thus automatically take advantage of regions where data density is high, and at the same time are especially smooth over regions where there are few observations.

The most common computational technique for spline smoothing is to use an order four B-spline basis function expansion with knots at the sampling points, and to minimize criterion (5.3) with respect to the coefficients of the expansion. In this case, the fitting function is piece-wise cubic, and the method is often referred to as *cubic spline smoothing*.

Recalling the relation between number of knots, the order of the spline and the number of basis functions that was described in Chapter 3, using order four B-splines in this way implies that we have $n+2$ basis functions, which are obviously enough to fit n data points exactly if $\lambda = 0$.

5.2.5 How spline smooths are computed

Reminding ourselves of expressions and relations drawn from Chapter 4 will help us to see how the use of a roughness penalty changes the smoothing process from a projection to something that generalizes the idea of a projection.

Recall that, without a roughness penalty, the coefficient vector $\mathbf{c}$ in the expansion

$$x(t) = \sum_{k}^{K} c_k \phi_k(t) = \mathbf{c}'\boldsymbol{\phi}(t) = \boldsymbol{\phi}'(t)\mathbf{c},$$

where $\mathbf{c}$ is the K-vector of coefficients and $\boldsymbol{\phi}$ is the K-vector of basis functions, has the solution

$$\hat{\mathbf{c}} = (\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}'\mathbf{W}'\mathbf{y} \tag{5.4}$$

where n by K matrix $\boldsymbol{\Phi}$ contains the values of the K basis functions at the n sampling points, $\mathbf{W}$ is a weight matrix to allow for possible covariance structure among residuals, and where $\mathbf{y}$ is the vector of discrete data to be smoothed. The corresponding expression for the vector of fits to the data is

$$\hat{\mathbf{y}} = \boldsymbol{\Phi}(\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}'\mathbf{W}\mathbf{y} = \mathbf{S}_\phi\mathbf{y}, \tag{5.5}$$

where $\mathbf{S}_\phi$ is the projection operator

$$\mathbf{S}_\phi = \boldsymbol{\Phi}(\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}'\mathbf{W} \tag{5.6}$$

corresponding to the basis system $\boldsymbol{\phi}$.

We can re-express the roughness penalty $\texttt{PEN}_m(x)$ in matrix terms as follows.

$$\texttt{PEN}_m(x) \quad = \quad \int [D^m x(s)]^2 \, ds$$

$$
\begin{aligned}
&= \int [D^m \mathbf{c}'\boldsymbol{\phi}(s)]^2 \, ds \\
&= \int \mathbf{c}' D^m \boldsymbol{\phi}(s) D^m \boldsymbol{\phi}'(s) \mathbf{c} \, ds \\
&= \mathbf{c}' [\int D^m \boldsymbol{\phi}(s) D^m \boldsymbol{\phi}'(s) \, ds] \mathbf{c} \\
&= \mathbf{c}' \mathbf{R} c \ , \qquad (5.7)
\end{aligned}
$$

where

$$\mathbf{R} = \int D^m \boldsymbol{\phi}(s) D^m \boldsymbol{\phi}'(s) \, ds \ . \qquad (5.8)$$

Note that we will often encounter matrices like $\mathbf{R}$ that contain integrals of *outer products* of vectors of functions, and it will keep the notation cleaner if we can suppress the argument s and ds and use the notation

$$\mathbf{R} = \int D^m \boldsymbol{\phi} D^m \boldsymbol{\phi}' \ .$$

By adding the error sum of squares $\mathtt{SSE}(y|c)$ and $\mathtt{PEN}_m(x)$ multiplied by a smoothing parameter λ, we obtain

$$\mathtt{PENSSE}_m(\mathbf{y}|\mathbf{c}) = (\mathbf{y} - \boldsymbol{\Phi}\mathbf{c})' \mathbf{W} (\mathbf{y} - \boldsymbol{\Phi}\mathbf{c}) + \lambda \mathbf{c}' \mathbf{R} \mathbf{c} \ . \qquad (5.9)$$

Taking the derivative with respect to parameter vector $\mathbf{c}$, we obtain

$$-2\boldsymbol{\Phi}'\mathbf{W}\mathbf{y} + \boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi}\mathbf{c} + \lambda \mathbf{R}\mathbf{c} = 0,$$

from which we obtain the expression for the estimated coefficient vector

$$\hat{\mathbf{c}} = (\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi} + \lambda \mathbf{R})^{-1} \boldsymbol{\Phi}'\mathbf{W}\mathbf{y} \ . \qquad (5.10)$$

5.2.6 Spline smoothing as a linear operation

The expression for the data-fitting vector $\hat{\mathbf{y}}$ is

$$\hat{\mathbf{y}} = \boldsymbol{\Phi}(\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi} + \lambda \mathbf{R})^{-1} \boldsymbol{\Phi}'\mathbf{W}\mathbf{y} = \mathbf{S}_{\phi,\lambda}\mathbf{y} \ , \qquad (5.11)$$

where the order n symmetric "hat" matrix is

$$\mathbf{S}_{\phi,\lambda} = \boldsymbol{\Phi}(\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi} + \lambda \mathbf{R})^{-1} \boldsymbol{\Phi}'\mathbf{W} \ . \qquad (5.12)$$

Comparing this expression with (5.6) shows us that the only change is the addition of $\lambda\mathbf{R}$ to the cross-product matrix $\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi}$ prior to its inversion, and that the two operators become identical when $\lambda = 0$. The more general operator (5.12) can be called a *sub-projection* operator because, unlike the projection operator, the sub-projection does not satisfy the *idempotency* relation, since

$$\mathbf{S}_{\phi,\lambda}\mathbf{S}_{\phi,\lambda} \neq \mathbf{S}_{\phi,\lambda}.$$

In plain language, this says that the spline smooth of a spline smooth is even smoother.

Expressions of the form (5.10) occur often in statistics where linear models are used. For example, in multilevel or random coefficient models, a similar expression arises when information about within-level regression coefficients is borrowed across levels. In Bayesian versions of multiple regression, the variance-covariance matrix $\mathbf{\Sigma}_0$ of the prior density for the regression coefficient matrix shows up where we have $\mathbf{R}$. Indeed, we can think of the roughness penalty as analogous to the logarithm of a prior density, just as the error sum of squares term is, except for a scale factor, the logarithm of a likelihood. An early example of regularization, *ridge regression*, also used this operator.

Computing the matrix $\mathbf{R}$ will generally require approximating the integral in (5.8) by a numerical quadrature scheme, although exact expressions are possible where both B-spline and Fourier bases are involved. In fact, it is seldom necessary to have very high accuracy in the approximation. An illustration of this point is that replacing $\mathbf{R}$ by a matrix of mth order difference operators applied to the coefficients themselves appears to work very well as a smoothing technique for equally spaced sampling points (Eilers and Marx, 1996).

It is also useful to plot the *linear filter* defined by the smoothing process for estimating acceleration. Let $\mathbf{\Phi}^{(2)}$ contain the values of the second derivatives of the basis functions evaluated at the sampling points, that is, $D^2\phi_k(t_j)$, and let $\hat{\mathbf{y}}^{(2)}$ be the vector of acceleration estimates at the sampling points. Then

$$\hat{\mathbf{y}}^{(2)} = \mathbf{\Phi}^{(2)}(\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi} + \lambda\mathbf{R})^{-1}\mathbf{\Phi}'\mathbf{W}\mathbf{y} = \mathbf{S}^{(2)}_{\phi,\lambda}\mathbf{y} \ ,$$

where $\mathbf{S}^{(2)}_{\phi,\lambda}$ is the matrix mapping the data vector into the vector of acceleration estimates. Rows in this matrix corresponding to acceleration estimates at ages one, ten and eighteen years are displayed in Figure 5.2. Notice that the acceleration estimate at

- age one requires some weighting of data all the way up to eight years,
- age ten, in the middle of the pubertal growth spurt, uses data from ages seven to thirteen, and
- age eighteen uses data back to age thirteen.

If the widths of these age ranges seems surprising, recall, firstly, that acceleration is intrinsically much harder to estimate than height; and, secondly, that the sparse sampling of function values forces us to borrow information from widely dispersed sampling points. Put another way, acceleration at age ten is a composite of a peak, a valley, and a peak, and uses about twelve data points, which works out to four per feature, and this in turn is close to the minimum possible of three per feature.

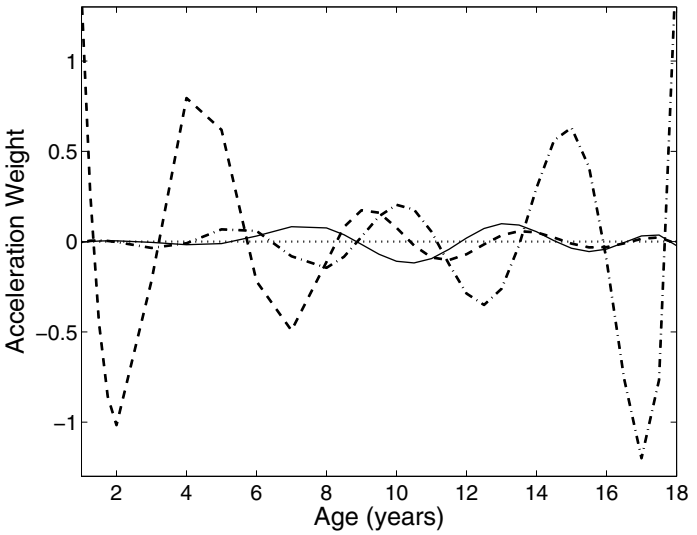

Figure 5.2. The solid curve indicates the weights placed on observations in the growth data for estimating acceleration at age ten. The dashed line corresponds to weights for height acceleration at age one, and the dashed-dotted line for age eighteen.

Another useful application of $\mathbf{S}_{\phi,\lambda}$ is in computing a *degrees of freedom* value for a spline smooth,

$$df(\lambda) = \operatorname{trace} \mathbf{S}_{\phi,\lambda} \ . \tag{5.13}$$

Hastie and Tibshirani (1990) discuss this and other ways of assessing the degrees of freedom of a smoothing procedure and, more generally, any estimation procedure that maps the data vector linearly to a parameter vector. Zhang (2003) offers a more in-depth update of this issue.

5.2.7 Spline smoothing as an augmented least squares problem

Expression (5.9) can be interpreted as arising from an *augmented least squares problem.* First, since $\mathbf{R}$ is a positive semidefinite matrix because of its cross-product structure, we can express it as

$$\mathbf{R} = \mathbf{L}'\mathbf{L}$$

by applying, among other possibilities, the Choleski decomposition. Now let

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} \ ,$$

where the zero vector is of the same length as $\mathbf{c}$. We can match this augmented response vector by the augmented design matrix

$$\tilde{\mathbf{\Phi}} = \begin{bmatrix} \mathbf{\Phi} \\ \sqrt{\lambda}\mathbf{L} \end{bmatrix} .$$

Finally, we augment the weight matrix $\mathbf{W}$ with the identity matrix $\mathbf{I}$ on the diagonal and zeros elsewhere to get the augmented weight matrix $\tilde{\mathbf{W}}$.

Now we can express coefficient vector $\mathbf{c}$ using the roughness penalty as the solution to the weighted least squares problem

$$\mathtt{SSE}(\tilde{\mathbf{y}}|\mathbf{c}) = (\tilde{\mathbf{y}} - \tilde{\mathbf{\Phi}}\mathbf{c})'\tilde{\mathbf{W}}(\tilde{\mathbf{y}} - \tilde{\mathbf{\Phi}}\mathbf{c}) \ . \tag{5.14}$$

This version of the roughness penalty problem makes clear that a roughness penalized least squares is a regular least squares where the data $\mathbf{y}$ are augmented by a vector of zeros, and the zeros are fit using the augmented portion of the design matrix $\sqrt{\lambda}\mathbf{L}$. Moreover, using the QR decomposition to minimize (5.14) rather than using (5.10) directly is preferable from the standpoint of rounding error in computing $\hat{\mathbf{c}}$.

5.2.8 Estimating derivatives by spline smoothing

Many functional data analyses call for the estimation of derivatives, either because these are of direct interest, or because they play a role in some other part of the analysis. The penalty (5.1) may not be suitable, since it controls curvature in x itself, and therefore only slope in the derivative Dx. It does not require the second derivative D^2x even to be continuous, let alone smooth in any sense.

If the derivative of order m is the highest required, one should actually penalize the derivatives of order $m + 2$ in order to control the curvature of the highest order derivative. For example, the estimate of acceleration is better if we use

$$\mathtt{PEN}_4(x) = \int [D^4x(s)]^2 ds = \|D^4x\|^2 \tag{5.15}$$

in (5.3) since this controls the curvature in D^2x.

We can use relation (5.13), for example, to compare the acceleration estimates by least squares and roughness penalized smoothing from the single simulated observation in Figure 4.3. By solving for the value of λ that produces a value of *df* of 12, we obtain $\lambda = 0.06$. Smoothing the data with this observation produces the acceleration estimate shown as a heavy line in Figure 5.3. This estimate does much better at the boundaries than the least squares estimate, which is also shown. Over the interior of the interval, however, the two estimates are rather similar, although the spline smooth does a slightly better job on the pubertal growth spurt.

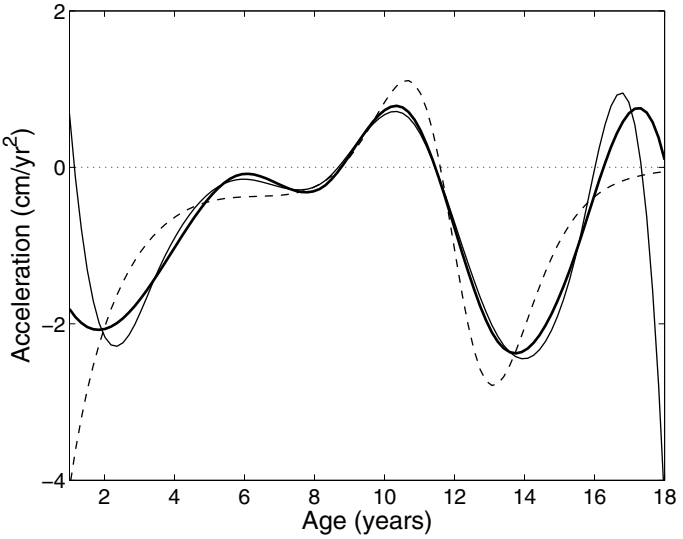

Figure 5.3. The heavy solid curve is the estimated growth acceleration for a single set of simulated data shown in Figure 4.3 computed by roughness penalized spline smoothing. The lighter solid line is for least squares smoothing, and the dashed curve is the errorless true curve.

5.3 Some extensions

The spline smoothing procedure given above can be extended in many ways, and many of these extensions are of great importance in applications. We summarize fairly briefly a number of these in this section.

5.3.1 Roughness penalties with fewer basis functions

In applications such as the study of handwriting and other forms of movement, we may use motion capture equipment that produce observations hundreds or thousands of times per second. Even the nondurable goods manufacturing index involves nearly a thousand sampling points. In these situations, placing a knot at every sampling point may imply a prohibitive amount of computation. Moreover, rounding errors may accumulate in the calculations to the point where the results are seriously inaccurate. See Section 5.4.3 below for more comments on this problem.

In these situations involving very large numbers of sampling points, it may entirely reasonable to use a lower-dimensional B-spline basis defined by some appropriate more limited knot sequence τ, provided that there remains sufficient flexibility to capture the features of interest. For example, handwriting data in Ramsay (2000) involved sampling pen position 400 times per second. The strokes making up the characters being produced

each lasted around 120 milliseconds, and thus included about 50 argument values, but was found that only about ten equally-spaced knots per stroke was more than sufficient to capture the shape of any stroke as well as three of its derivatives.

As a further example, 34 years of daily weather measurements represents about 12,500 observations, and it is a heavy chore to use that many basis functions. Instead, a system of 500 spline basis functions was considered sufficient in Ramsay and Silverman (2002) to capture all the variation of interest, and a roughness penalty was then used with this system to impose further smoothness on the result.

None of the mathematics outlined above changes when we use fewer knots than sampling points, and yet roughness penalization can remain an effective way to ensure a smooth fit and stable derivative estimates.

5.3.2 More general measures of data fit

There are aspects of the roughness penalty method that are really useful in our treatment of functional data analysis. For example, instead of quantifying fit to the data by the residual sum of squares, we can penalize *any* criterion of fit by a roughness penalty. For instance, we might have a model for the observed y_j for which the log likelihood of x can be written down. Subtracting $\lambda \times \texttt{PEN}_2(x)$ from the log likelihood and then finding the maximum allows smoothing to be introduced in a wide range of statistical problems, not merely those in which error is appropriately measured by a residual sum of squares. These extensions of the roughness penalty method are a major theme of Green and Silverman (1994).

In the functional data analysis context, we adopt this philosophy in considering functional versions of several multivariate techniques. The function estimated by these methods is expressed as the solution of a maximization (or minimization) problem based on the given data. For example, principal components are chosen to have maximum possible variance subject to certain constraints. By penalizing this variance using a roughness penalty term appropriately, the original aim of the analysis can be traded off against the need to control the roughness of the estimate. There are different ways of incorporating the roughness penalty according to the context, but the overall idea remains the same: Penalize whatever is the appropriate measure of goodness-of-fit to the data for the problem under consideration.

5.3.3 More general roughness penalties

The second extension of the roughness penalty method uses measures of roughness other than $\|D^2x\|^2$. We have already seen one reason for this in Section 5.2.8, where the estimation of derivatives of x was considered. However, even if the function itself is of primary interest, there are two related reasons for considering more general roughness penalties. On the

one hand, we may wish that the class of functions with zero roughness were wider than, or otherwise different from, those that are of the form $a + bt$. On the other hand, we may have in mind that, locally at least, curves x should ideally satisfy a particular differential equation, and we may wish to penalize departure from this.

For example, if we are analyzing periodic data, it would be more natural to use the *harmonic acceleration* operator

$$Lx = D^3x + \omega^2 Dx \tag{5.16}$$

since zero roughness implies that x is of the form

$$x(t) = c_1 + c_2 \sin \omega t + c_3 \cos \omega t,$$

where ω is the period.

We can achieve both of these goals by replacing the second derivative operator D^2 with a more general linear differential operator L, defined as

$$Lx = w_0 x + w_1 Dx + \ldots + w_{m-1} D^{m-1} x + D^m x,$$

where the weights w_j may be either constants or functions $w_j(t)$. Then we can define

$$\texttt{PEN}_L(x) = \int [(Lx)^2](t)\, dt = \|Lx\|^2, \tag{5.17}$$

the integral of the square of Lx.

As an alternative to pre-specifying the differential operator, we can use observed functional data to *estimate* the operator L. These ideas are developed further in Chapters 19 and 21.

5.3.4 Computing the roughness penalty matrix

The roughness penalty matrix $\mathbf{R}$ defined in (5.8) is composed of the integrals of products of a derivative D^m of basis functions. For B-spline bases, Fourier bases, and most of the basis systems that we are likely to work with in practice, these integrals can be computed analytically. In the B-spline case, however, the details (see de Boor, 2002) are intricate, and few users of FDA will want to write programming code for this problem. There are functions in the MATLAB®, R and S-PLUS languages that can do this work for you.

When more general roughness penalties are involved of the kind defined in (5.17) above, it will be necessarily to resort to numerical approximation of the integrals in (5.8) for matrix $\mathbf{R}$. There are two main strategies in this case.

The safer approach is to use a numerical method that iteratively improves its estimate of an integral until a test for its accuracy is satisfied. A classic approach is to use a simple method such as the trapezoidal rule and to double the number of points at which the integrand is evaluated

until an estimate of the integral is judged to have converged. We have had good experience with Romberg integration, also called Richardson's extrapolation, and have used variants of the algorithm described in Press et al. (1999). However, there are more modern methods that may well perform even better.

However, these adaptive methods can be too slow for applications where $\mathbf{R}$ must be evaluated many times during the course of a calculation. In this case, a lower accuracy non-iterative approach that is still considered to be sufficiently accurate may be preferable. For example, the integrals in (5.8) can be converted to matrix products using a fine mesh of values of t and a numerical quadrature method such as Simpson's Rule (Stoer and Bulirsch, 2002). As a rough guideline, we have found that about 21 evaluation points per interval when working with B-spline basis functions gives a level of accuracy that has sufficed for our purposes.

If multiple knots at the same location are used in order to allow for discontinuity in a derivative or function value, be careful not to evaluate the discontinuous quantity at the function value. Aside from the fact that the value is not defined mathematically, available software for evaluating spline basis functions can fail to warn you that you did something wrong, and cheerfully return a function value of large and unpredictable size, which will play havoc with your integral approximation. The better procedure is to carry out the integration piecewise over each interval, and integrate only up to a t-value separated from the knot location by a small constant.

5.4 Choosing the smoothing parameter

When we fit data using a roughness penalty instead of least squares, we switch from defining the smooth in terms of degrees of freedom K to defining the smooth in terms of the smoothing parameter λ. Nevertheless, strategies for selecting λ are rather similar to those that we used in Chapter 4 in that we use a "discounted" measure of fit that compensates for the degrees of freedom in the data used up by the fit.

5.4.1 Some limits imposed by computational issues

Although from a mathematical perspective we can contemplate any positive values of λ, the realities of floating point computation actually impose some severe limits. These limits are due to the need to solve a system of linear equations with the coefficient matrix

$$\mathbf{M}(\lambda) = \mathbf{\Phi}'\mathbf{W}\mathbf{\Phi} + \lambda\mathbf{R},$$

where $\mathbf{R}$ is defined in (5.8). The two matrices $\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}$ and $\mathbf{R}$ can have elements of radically different sizes. In particular, the size of

$$\|\mathbf{R}\| = \sqrt{\sum_k \sum_\ell r_{k\ell}^2}$$

increases by roughly an order of magnitude for each increase in the order m of derivative that is used to define the roughness penalty.

Now $\mathbf{R}$ itself has rank $K-m$, and so cannot itself be useful as a coefficient matrix. This implies that we cannot have $\lambda\mathbf{R}$ so large as to overwhelm $\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}$; otherwise, attempting to invert $\mathbf{M}(\lambda)$ will either produce an error message or, worse, a result that is so full of rounding error as to lead to seriously wrong results further on down the line. A rough rule of thumb is that the size of $\lambda\mathbf{R}$ should not be more than 10^{10} times the size of $\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}$.

Consider the handwriting data, for example. There are 1401 sampling points evenly spaced between 0 and 2.3 seconds. We will need to estimate the third derivative of the X and Y coordinates of pen position in Chapter 19, and consequently will need to penalize the size of the derivative of order $m = 5$. The minimal order of B-spline that will serve to define an integrable fifth derivative is 7. If we choose to use smoothing splines with a knot at each sampling value, this implies 1406 basis functions defining matrix $\mathbf{\Phi}$. The size $\|\mathbf{R}\|$ of $\mathbf{R}$ in this context is about 2×10^{31}! By contrast, $\|\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}\| \approx 20$. Hence, by our rule of thumb, we will be in trouble if $\lambda > 10^{-20}$ or so.

This illustrates the importance of some preliminary explorations along these lines before plunging into functional data analysis, and especially when high orders of derivatives are involved. In any case, the cure is simple; as we indicated in Section 3.7, these problems arise because the unit of measurement, 2.3 seconds, for t is far larger than the length of the interval over which a spline basis function is nonzero. Measuring time in milliseconds removes the problem.

On the lower limit side, we clearly cannot always use $\lambda = 0$; in this example, there are more basis functions than data points and consequently $\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}$ would not be invertible. Again, a rule of thumb can be proposed: Choose λ at least large enough to ensure that the size of $\lambda\mathbf{R}$ is at least with ten orders of magnitude of the size of $\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}$.

Now we turn to two strategies for choosing smoothing parameter somewhere between these broad limits.

These difficulties are actually a result of the way the penalized least squares criterion is defined in almost all the statistical literature. The application of the method of *dimensional analysis* used routinely in the physical sciences can be helpful here. The basic idea is that two quantities that are added should have the same units of measurement.

Now the error sum of squares $\|\mathbf{y} - \hat{\mathbf{y}}\|^2$ has the unit of measurement of x squared. In the handwriting data, this would be squared meters, for

example. We should probably also divide this criterion by n to allow for the number of sampling points.

Smoothing parameter λ can be made dimensionless by using its logarithm, which is consistent with the idea that it will be a positive quantity. Thus, we should multiply a roughness penalty such as $\texttt{PEN}_m(x)$ by 10^ν where $\nu = \log_{10}\lambda$. In fact, we basically do this already since we tend to vary λ by multiplying it by a fixed factor.

Finally, the units of $D^m x$ are those of x itself divided by the time unit, that we can indicate by τ, taken to the power m. This suggests that $\texttt{PEN}_m(x)$ should be multiplied by T^{2m}, where T is the length of time in τ units over which the integration takes places. This will cancel out the role of the time unit in the integrand. We might divide the integral, on the other hand, by T itself to allow for the summation over time that the integral represents. Putting this all together, it would be better to redefine the penalized least squares criterion as

$$\texttt{PENSSE}(x) = \frac{1}{n}\|\mathbf{y} - \hat{\mathbf{y}}\|^2 + 10^\nu T^{2m-1}\texttt{PEN}_m(x). \tag{5.18}$$

For the handwriting data, for example, if we use the time unit $\tau = 1$ second, so that the interval of integration is 2.3 seconds, along with $m = 5$ to control the curvature of the third derivative, then $T^9 \approx 1800$, but if we opt for milliseconds as the time unit, then $T^9 \approx 1.8 \times 10^{12}$. Now, of course, the fifth derivative takes on huge values in the time scale of seconds, but comparatively mild values on the time scale of milliseconds so that, finally, we will wind up using the same value of ν in either time scale.

5.4.2 The cross-validation or CV method

The basic idea behind *cross-validation* is to set part of the data to one side, calling it a *validation sample*, and fit the model to the balance of the data, called the *training sample*. In that way, we see how well the model fits data that were not used to estimate the model, thus avoiding the somewhat incestuous procedure of using the data to both fit the model and assess fit.

A versatile technique for choosing a smoothing parameter involves taking this notion to the extreme situation where we leave only one observation out as the validation sample, fitting the data to the rest, and then estimating the fitted value for the left out data value. If this procedure is repeated for each observation in turn, and the resulting error sum of squares summed over all values, the result is the *cross-validated* error sum of squares. We compute this criterion over a range of values of λ, and choose that value that yields its minimum.

Cross-validation can be used in a wide range of situations, and in effect rests only on the assumption that observations are relatively independent of one another. However, the method has two problems. First, it is usually computationally intensive, and not the sort of thing that would be feasible

for sample sizes in the thousands. However, there are specific situations in which some computational tricks can be used to reduce the computational burden. The second problem is that minimizing CV can lead to under-smoothing the data because the method tends too often to favor fitting noisy or high-frequency types of variation that we would prefer to ignore.

5.4.3 The generalized cross-validation or GCV method

A measure that is popular in the spline smoothing literature is the *generalized cross-validation* measure GCV developed by Craven and Wahba (1979). It was originally developed as a simpler version of the cross-validation procedure that avoided the need to re-smooth n times. But it also has been found to be rather more reliable than cross-validation in the sense of having less of a tendency to under-smooth. The criterion is usually expressed as

$$\texttt{GCV}(\lambda) = \frac{n^{-1}\,\texttt{SSE}}{[n^{-1}\text{trace}\,(\mathbf{I} - \mathbf{S}_{\phi,\lambda})]^2}\ ,$$

where df is the equivalent degrees of freedom measure (5.13) and $\mathbf{S}_\lambda$ is the smoothing operator defined in (5.12). But it can be more revealing to use the equivalent expression

$$\texttt{GCV}(\lambda) = \big(\frac{n}{n - df(\lambda)}\big)\big(\frac{\texttt{SSE}}{n - df(\lambda)}\big)\ . \tag{5.19}$$

Notice that this is a twice-discounted mean squared error measure. The right factor is the unbiased estimate of error variance σ^2 familiar in regression analysis, and thus represents some discounting by subtracting $df(\lambda)$ from n. The left factor further discounts this estimate by multiplying by $n/(n - df(\lambda))$.

As a practical matter, C. Gu (2002) reports that the remaining tendency for GCV to yield under-smoothing can be further reduced by multiplying df by factors such as 1.2 or 1.4 in (5.19). This is a third level of discounting, in effect. Apparently the additional discounting does not seriously increase the odds of over-smoothing the data.

The minimization of GCV with respect to λ will inevitably involve trying a large number of values of λ, whether grid-search or a numerical optimization algorithm is used. The computation of $\texttt{GCV}(\lambda)$ can be greatly speeded up by performing a preliminary generalized eigenanalysis. Criterion GCV can be expressed in terms of the n by N data matrix $\mathbf{Y}$, the $n \times K$ matrix $\mathbf{\Phi}$ of basis function values and the order K penalty matrix $\mathbf{R}$ as follows:

$$\texttt{GCV}(\lambda) = \frac{n\,\text{trace}\{\mathbf{Y}'[\mathbf{I} - \mathbf{S}_{\phi,\lambda}]^{-2}\mathbf{Y}\}}{\{\text{trace}[\mathbf{I} - \mathbf{S}_{\phi,\lambda}]\}^2}\ ,$$

where the "hat" matrix $\mathbf{S}_{\phi,\lambda}$ has the expression

$$\mathbf{S}_{\phi,\lambda} = \mathbf{\Phi}\mathbf{M}(\lambda)^{-1}\mathbf{\Phi}'\mathbf{W}$$

and where, in turn

$$\mathbf{M}(\lambda) = \mathbf{\Phi}'\mathbf{W}\mathbf{\Phi} + \lambda\mathbf{R}.$$

Note that we have dropped the weight matrix $\mathbf{W}$ from these expressions to keep the notation a little simpler.

We actually don't need to invert $\mathbf{M}(\lambda)$ each time we change λ, but we do need to solve a linear system of equations for which it is the coefficient matrix, and this is that we want to avoid. This can be achieved if we first solve the generalized eigenvalue problem

$$\mathbf{R}\mathbf{V} = \mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}\mathbf{V}\mathbf{D},$$

where $\mathbf{D}$ is the matrix of eigenvalues of $\mathbf{R}$ in the metric defined by $\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}$ and $\mathbf{V}$, the columns of which are the corresponding eigenvectors of $\mathbf{R}$, satisfy the orthogonality condition

$$\mathbf{V}'\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}\mathbf{V} = \mathbf{I}.$$

Note that the generalized eigenvalue problem has a solution only if $\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}$ is nonsingular. This will not be the case if knots are placed at every data point. However, a trick recommended by de Boor (2002) is to drop enough knots next to the boundary to make the number of basis functions equal to the number of sampling points. For example, if we are working with cubic smoothing splines of order four and we have 101 sampling points, then this implies 103 basis functions. But if we drop the knots associated with sampling points 2 and 100, the number of basis functions drops to 101, and $\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}$ will be nonsingular, at least if sampling points are reasonably well-spaced. It is, needless to say, always a good idea to check $\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}$ for singularity.

We now express, for any new value of λ, the required inverse very efficiently as

$$\mathbf{M}(\lambda)^{-1} = \mathbf{V}(\mathbf{I} + \lambda\mathbf{D})^{-1}\mathbf{V}',$$

since the matrix now being inverted is diagonal. Moreover, taking the derivative of $\texttt{GCV}(\lambda)$ involves calculating the matrix

$$\mathbf{M}(\lambda)^{-1}\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi}\mathbf{M}(\lambda)^{-1} = \mathbf{V}(\mathbf{I} + \lambda\mathbf{D})^{-2}\mathbf{V}'$$

so that providing a derivative value to a numerical optimization algorithm is also computationally efficient and likely to decrease the number of evaluations of $\texttt{GCV}(\lambda)$ substantially.

Gu (2002) offers a detailed and up to date discussion of theoretical and computational issues associated with the $\texttt{CV}(\lambda)$, $\texttt{GCV}(\lambda)$ and other methods for choosing λ.

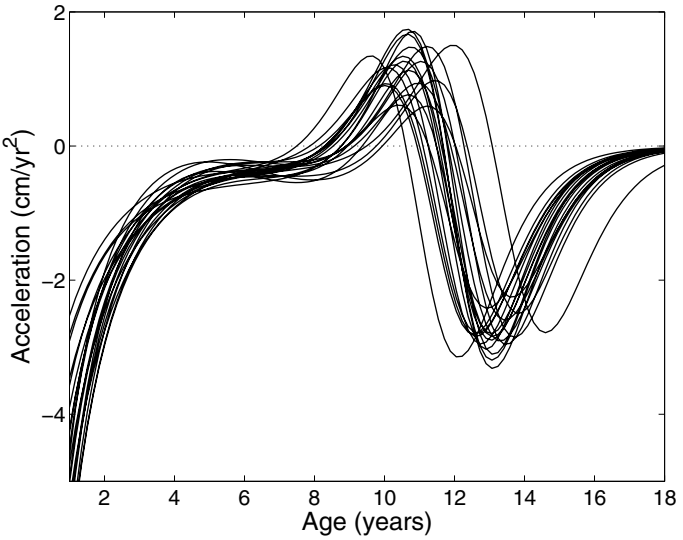

Figure 5.4. A sample of twenty height acceleration curves for females generated using the Jolicoeur model.

5.4.4 Spline smoothing the simulated growth data

We illustrate here some of the points made in this chapter by the analysis of 1000 simulated records for females using the Jolicoeur model described in Section 4.3. A random sample of twenty acceleration curves from this model are shown in Figure 5.4.

Figure 5.5 shows the variation of the generalized cross-validation statistic GCV over a range of $\log_{10}(\lambda)$ values in its top panel. We see that the minimum GCV is attained at $\lambda = -0.1$. At this smoothing level, the degrees of freedom measure has the value of 11.4, which is not far from the number twelve of basis functions that we used in least squares smoothing.

In the lower panel, we see the square root of the mean squared error ($RMSE$) of the acceleration curve values at ages eight, before puberty; twelve, mid-puberty for the average girl; and sixteen, post-puberty for most girls. These curves do not bottom out at the same value as the GCV curve, but they come close to doing so. It is not surprising that the curve for age twelve favors a lower value of λ; the curvature of the acceleration function is much sharper for the average girl at mid-puberty. The more stable curves typical for most girls at ages eight and sixteen favor higher values of λ. Nevertheless, the GCV-favored value gives nearly optimal values for RMSE.

Figure 5.6 indicates the variation in RMSE, bias, and sampling standard error over age for the smoothing level minimizing GCV. We see that the curve estimates are of limited value for ages less than three years or more than sixteen years. But they aren't bad at all in between these extremes,

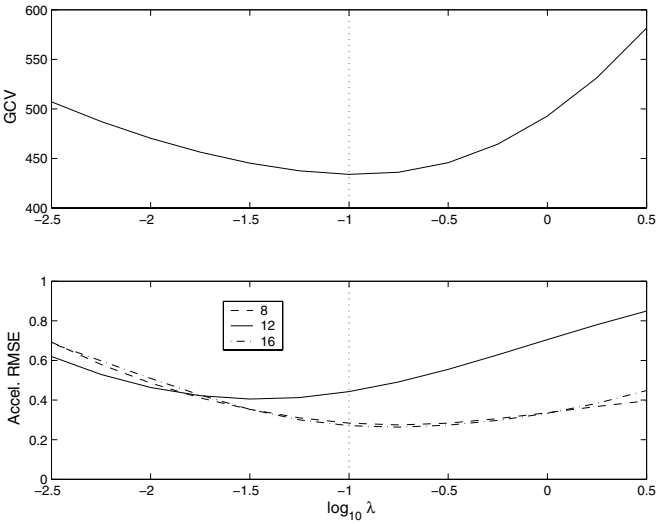

Figure 5.5. The top panel displays the relation between the GCV statistic and smoothing level for 1000 simulated female records. The bottom panel displays the root-mean-squared error in acceleration estimates at the selected ages of 8, 12 and 16 years of age.

and the bias in particular is small. Of course, we could do better if we had sampled height more often. It is also not surprising that sampling error is higher during the pubertal growth spurt when curvature is high.

Perhaps the main conclusion to be drawn here is that the spline smoothing method does a good job in this context, and especially given that there are only 31 observations in each record. Choosing λ using the GCV criterion gets us close to the best answer, on the average.

5.5 Confidence intervals for function values and functional probes

We now want to see how to compute confidence limits on some useful quantities that depend on an estimated function x that has, in turn, been computed by the smoothing with a roughness penalty a vector of discrete data $\mathbf{y}$.

For example, how precisely is the function value at t, $x(t)$, determined by our sample of data $\mathbf{y}$? Or, what sampling standard deviation can we expect if we re-sample the data over and over again, estimating $x(t)$ anew with each sample? Can we construct a pair of *confidence limits* such that the probability that the true value of $x(t)$ lies within these limits is a specified

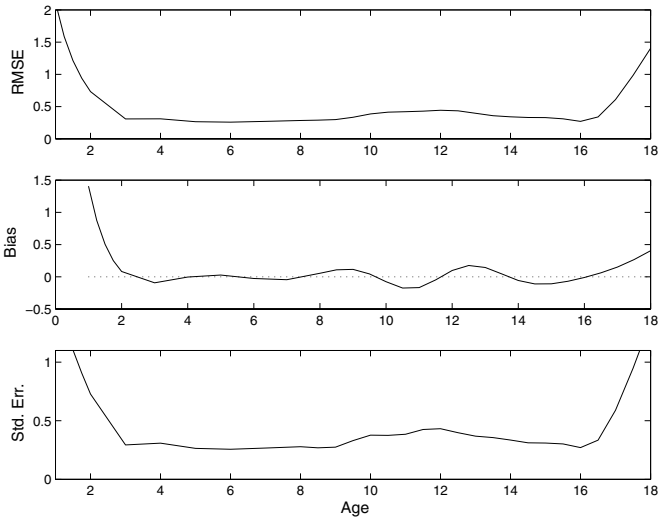

Figure 5.6. The root-mean-squared error for the GCV-optimal smoothing level as a function of age is shown in the top panel, and the corresponding values of bias and sampling standard error are shown in the middle and bottom panels, respectively.

value, such as 0.95? Displaying functions or their derivatives with point-wise confidence limits is a useful way of conveying how much information there is in the data used to estimate these functions. See Figure 5.7 below for an example.

However, do be aware of the distinction between these point-wise limits, which tell us only the precision at a fixed location, and global confidence limits, which would tell us a region of confidence for the entire function. Constructing an upper and a lower curve such that the probability that the *entire* true curve lies between these functional limits can be achieved by computationally intensive methods such as bootstrapping (Efron and Tibshirani, 1993).

5.5.1 Linear functional probes

More generally, we may wish to examine quantities of the form

$$\rho_\xi(x) = \int \xi(t)x(t)\,dt\ . \tag{5.20}$$

We use the term *functional probe* for the quantity $\rho_\xi(x)$ and *linear probe function* for the weighting function ξ that defines it. The probe function, in turn, is chosen so as to highlight some interesting feature, such as a

peak, valley, or difference between function values over two non-overlapping regions.

A probe is a generalization of the notion of a *contrast* in analysis of variance, used there to probe a set of treatment effects for specific types of variation. However, there is no need for the values of ξ to integrate to zero. If we have multiple probes, it may be helpful for pairs of probe functions to be orthogonal, but this is not essential.

For example, to highlight the behavior of x over an interval, an appropriate probe function ξ might be the box function, which takes the value 1 within the interval and 0 elsewhere. Or, to highlight the difference between x in two intervals A and B of equal length, one could use a probe function taking the value 1 on A, -1 on B, and 0 elsewhere. In cases like these, we will want to compute the sampling standard deviation of the scalar ρ_ξ in order to decide whether it differs significantly from some reference value like zero.

Functional probes ρ_ξ of this nature include the simpler situation of $x(t)$ as a special case, since $x(t)$ can be obtained by choosing ξ to be nonnegative and concentrating its nonzero values arbitrarily near t while preserving unit area under the its curve. We can denote such a probe by

$$\rho_t(x) = x(t) \ , \tag{5.21}$$

and it is called the *evaluation map* because it maps function x into its value $x(t)$ at t. Probes of this nature are taken up in detail in Section 20.3.

Probe ρ_ξ is a linear function of the estimated smoothing function x in the sense if that we multiply two such functions, x_1 and x_2 by the constants a and b, respectively, then

$$\rho_\xi(ax_1 + bx_2) = a\rho_\xi(x_1) + b\rho_\xi(x_2).$$

This linearity implies that there is a linear transformation of the coefficient vector $\mathbf{c}$ that defines x that yields the value $\rho_\xi(x)$. At the same time, we already worked out in this chapter the linear transformation that takes or *maps* the data vector $\mathbf{y}$ to the coefficient vector $\mathbf{c}$.

5.5.2 Two linear mappings defining a probe value

In order to study the sampling behavior of ρ_ξ, we need to compute these two linear mappings plus their composite. They are given names and described as follows:

1. Mapping `y2cMap`, which converts the raw data vector $\mathbf{y}$ to the coefficient vector $\mathbf{c}$ for the basis function expansion of x. If $\mathbf{y}$ and $\mathbf{c}$ are lengths n and K, respectively, the mapping is a K by n matrix $\mathbf{S}$ such that

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

2. Mapping `c2rMap`, which converts the coefficient vector $\mathbf{c}$ to the scalar quantity $\rho_\xi(x)$. This mapping is a 1 by K row vector $\mathbf{L}$ such that

$$\rho_\xi(x) = \mathbf{Lc} \ .$$

3. The composite mapping called `y2rMap` defined by

$$\texttt{y2rMap} = \rho_\xi(x) = \texttt{c2rMap} \circ \texttt{y2cMap},$$

which takes data vector $\mathbf{y}$ directly to the probe value and is the 1 by n row vector $\mathbf{LS}$ that yields

$$\rho_\xi(x) = \mathbf{LSy} \ .$$

As an illustration, consider a conventional linear regression model with design matrix $\mathbf{Z}$

$$\mathbf{y} = \mathbf{Zc} + \mathbf{e},$$

where the regression coefficient vector $\mathbf{c}$ is estimated by ordinary least squares. Then, since $\mathbf{c} = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'\mathbf{y}$, the matrix corresponding to `y2cMap` is $\mathbf{S} = (\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'$. Now suppose that for some reason we want to estimate the difference between the first and second regression coefficients, possibly because we conjecture that they may be equal in the population. Then the probe function ξ is equivalent to the probe vector $\mathbf{L} = (1, -1, 0, \ldots)$, and this is the row vector corresponding to mapping `c2rMap`. Finally, the composite mapping `y2rMap` taking $\mathbf{y}$ directly into the value of this difference is simply the row vector $\mathbf{L}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'$.

Now the random behavior of the estimator of whatever we choose to estimate is ultimately tied to the random behavior of the data vector $\mathbf{y}$. Let us indicate the order n variance-covariance matrix of $\mathbf{y}$ as $\texttt{Var}(y) = \mathbf{\Sigma}_e$, as we did in Sections 4.6.1 and 4.6.2. Recall that we are operating in this chapter with the model

$$\mathbf{y} = x(\mathbf{t}) + \boldsymbol{\epsilon} \ ,$$

where $x(\mathbf{t})$ here means the n-vector of values of x at the n argument values t_j. In this model $x(\mathbf{t})$ is regarded as fixed, and as a consequence $\mathbf{\Sigma}_e = \texttt{Var}(\boldsymbol{\epsilon})$.

5.5.3 Computing confidence limits for function values

Now let's express these mappings in the context of estimating confidence limits specifically for a function value $x(t)$. Let n by K matrix $\mathbf{\Phi}$ contain the values $\phi_k(t_j)$, and let the matrices $\mathbf{R}$ and $\mathbf{W}$ be defined as before. Then the matrix corresponding to `y2cMap` is

$$\mathbf{S} = (\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi} + \lambda\mathbf{R})^{-1}\mathbf{\Phi}'\mathbf{W}$$

for smoothing parameter λ.

Suppose that we are interested in the sampling behavior of the function value $\rho_t(x) = x(t)$. We know that

$$x(t) = \boldsymbol{\phi}'(t)\mathbf{c} = \boldsymbol{\phi}'(t)\mathbf{S}\mathbf{y},$$

and from this we can see that the 1 by K matrix $\mathbf{L}$ corresponding to `c2rMap` is simply $\boldsymbol{\phi}'$, the row vector resulting from evaluating each of the basis functions at t. And of course the composite mapping `y2rMap` corresponds to the matrix $\mathbf{LS}$. Consequently, using the expression for the variance of a linear transform of a random vector, we have that

$$\mathtt{Var}[\hat{x}(t)] = \mathbf{LS}\boldsymbol{\Sigma}_e\mathbf{S}'\mathbf{L}'. \tag{5.22}$$

The matrix $\mathbf{LS}$ used in (5.22) is also of interest in itself. Each row of this matrix indicates the profile of weights on the data used to define what is being estimated for that row. For example, if row j corresponds to the function evaluation $\rho_{t_j}(x)$ at time t_j, then a plot of the values in this row shows the entries in $\mathbf{y}$ that are used to define this estimate. A row of this matrix is often called a *linear filter* for estimating the quantity in question by engineers. See Figure 5.7 below for an example.

5.5.4 Confidence limits for growth acceleration

With this information in hand, we can gain an impression of how well the acceleration function can be estimated using the results in Section 5.5. If we use spline smoothing using order six B-splines as the basis for smoothing, a smoothing parameter $\lambda = 0.1$, and weight matrix $\mathbf{W}$ a diagonal matrix containing the values of the variances of estimate as derived from Figure 4.2, then Figure 5.7 shows the acceleration curve for the Jolicoeur model based on using the mean coefficients along with point-wise 95% confidence limits. The confidence limits balloon out at the extremes because of the difficulty of estimating derivatives in these regions.

5.6 A bi-resolution analysis with smoothing splines

We now turn to a more general approach, of which spline smoothing turns out to be a special case. So far we have used basis functions in two essentially different ways. In section 4.2 of Chapter 4, we forced the function x to lie in a relatively low dimensional space, defined in terms of a suitable basis. On the other hand, in Section 4.7, we did not assume that the whole function was in the span of a particular basis, but rather we considered a local basis expansion at any given point. In this section, we allow the function to have a higher-dimensional basis expansion, but use a roughness penalty in fitting the function to the observed data.

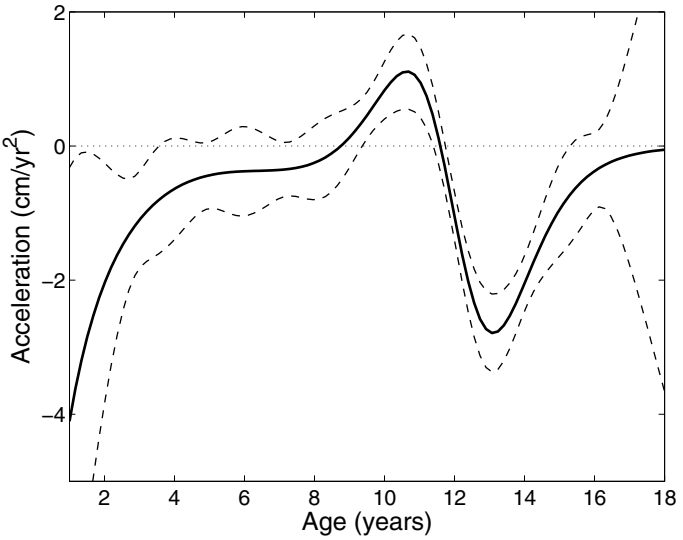

Figure 5.7. The solid curve is an acceleration curve derived from the Jolicoeur model. The dashed lines are 95% point-wise confidence limits on the curve based on a smoothing spline estimate from data having the standard error of measurement plotted in Figure 4.2.

5.6.1 Complementary bases

To develop our approach, suppose that we have two sets of basis functions, $\phi_j, j = 1, \ldots, J$ and $\psi_k, k = 1, \ldots, K$, that complement one another. Let functions ϕ_j be small in number and chosen to give reasonable account of the large-scale features of the data. The complementary basis functions ψ_k will generally be much larger in number, and are designed to catch local and other features not representable by the ϕ_j. Assume that any function x of interest can be expressed in terms of the two bases as

$$x(s) = \sum_{j=1}^{J} d_j \phi_j(s) + \sum_{k=1}^{K} c_k \psi_k(s). \tag{5.23}$$

For example, for the Canadian temperature data, the first three Fourier series functions with $\omega = \pi/6$ would be a natural choice for the ϕ_j, setting $J = 3$ and letting the $\boldsymbol{\phi}$ basis be the functions

$$1, \ \sin(\omega t), \ \cos(\omega t).$$

The appropriate choice for the ψ_k in this case would be the remaining K functions in an order $(J + K)$ Fourier series expansion. In the monthly temperature data case, they could be the remaining nine Fourier series terms needed to represent the data exactly. Usually, as in the Fourier case

above, the bases $[\phi_j]$ and $[\psi_k]$ are mutually linearly independent, and the expansion is unique, but this is not entirely essential to our method.

5.6.2 Specifying the roughness penalty

Let us now develop a roughness penalty for x so that linear combinations of the ϕ_j are in effect completely smooth, in that they contribute nothing to the roughness penalty. Then the roughness penalty must depend only on the coefficients of the ψ_k. One way of motivating this choice is by thinking of x as the sum of two parts, an "ultrasmooth" function $x_S = \sum_j d_j\phi_j$ and a function $x_R = \sum_k c_k\psi_k$. Therefore we seek a measure $\texttt{PEN}(x_R)$ of how rough, or in any other way important, we would consider the function x_R expressed solely in terms of the ψ_k. One possibility is simply to take the usual L_2 norm of x_R, defining

$$\texttt{PEN}_0(x_R) = \int x_R(s)^2\,ds = \int(\mathbf{c}'\boldsymbol{\psi})^2 = \int[\sum_{k=1}^{K} c_k\psi_k(s)]^2\,ds.$$

Another possibility is to take a certain order of derivative of the expansion prior to squaring and integrating, just as we did for the function x itself in Section 5.2. For example, we might use

$$\texttt{PEN}_2(x_R) = \int(D^2x_R)^2 = \int[\sum_{k=1}^{K} c_kD^2\psi_k(s)]^2\,ds$$

to assess the importance of x_R in terms of its total curvature, as measured by its squared second derivative, or $\texttt{PEN}_4(x_R) = \int(\mathbf{c}'D^4\boldsymbol{\psi})^2$ to assess the curvature of its second derivative. More generally, we can use any linear differential operator L, defining

$$\texttt{PEN}_L(x_R) = \int(Lx_R)^2 = \int[\sum_{k=1}^{K} c_kL\psi_k(s)]^2\,ds.$$

Of course, setting L as the identity operator or the second derivative operator yields $\texttt{PEN}_0$ and $\texttt{PEN}_2$ as special cases.

We can express these penalties in matrix terms as

$$\texttt{PEN}_L(x_R) = \mathbf{c}'\mathbf{R}\mathbf{c},$$

where the order K symmetric matrix $\mathbf{R}$ contains elements

$$\mathbf{R}_{kl} = \int L\psi_k(s)L\psi_l(s)ds.$$

If computing the integrals proves difficult, a simple numerical integration scheme, such as the trapezoidal rule applied to a fine mesh of argument values, usually suffices, and then we can also estimate derivatives numerically. Alternatively, we can specify $\mathbf{R}$ directly as any suitable symmetric

non-negative definite matrix, without explicit reference to the roughness of the function x_R.

Now we consider a general function x of the form (5.23), and simply define the roughness of x as

$$\texttt{PEN}(x) = \mathbf{c}'\mathbf{R}\mathbf{c}.$$

To express the penalized sum of squares, we need to express the residual sum of squares in terms of the coefficient vectors d and $\mathbf{c}$. Working just as in (4.1),

$$\sum_j [y_j - x(t_j)]^2 = \|\mathbf{y} - \mathbf{\Phi}\mathbf{d} - \mathbf{\Psi}\mathbf{c}\|^2,$$

where the $n \times K$ matrix $\mathbf{\Psi}$ has elements $\Psi_{ik} = \psi_k(t_j)$. We can now define the composite smoothing criterion

$$\texttt{PENSSE}_\lambda(x|y) = \|\mathbf{y} - \mathbf{\Phi}\mathbf{d} - \mathbf{\Psi}\mathbf{c}\|^2 + \lambda\mathbf{c}'\mathbf{R}\mathbf{c}. \tag{5.24}$$

We can minimize this quadratic form in $\mathbf{d}$ and $\mathbf{c}$ to find the fitted curve x in terms of its expansion (5.23) as follows. The solution for $\mathbf{d}$ for any fixed value of $\mathbf{c}$ is given by

$$\mathbf{d} = (\mathbf{\Phi}'\mathbf{\Phi})^{-1}\mathbf{\Phi}'(\mathbf{y} - \mathbf{\Psi}\mathbf{c}) \tag{5.25}$$

and, consequently,

$$\mathbf{\Phi}\mathbf{d} = \mathbf{S}_\phi(\mathbf{y} - \mathbf{\Psi}\mathbf{c}),$$

where the projection matrix $\mathbf{S}_\phi$ is

$$\mathbf{S}_\phi = \mathbf{\Phi}(\mathbf{\Phi}'\mathbf{\Phi})^{-1}\mathbf{\Phi}'.$$

In words, the ϕ basis component of the fit is the conventional basis expansion of the residual vector $\mathbf{y} - \mathbf{\Psi}\mathbf{c}$. Substitute this solution for $\mathbf{d}$ into $\texttt{PENSSE}_\lambda$ and define the complementary projection matrix $\mathbf{Q}_\phi$ by

$$\mathbf{Q}_\phi = \mathbf{I} - \mathbf{S}_\phi.$$

Recalling that because $\mathbf{Q}_\phi$ is a projection matrix, $\mathbf{Q}_\phi\mathbf{Q}_\phi = \mathbf{Q}_\phi$, we arrive at the equations

$$\begin{aligned} \hat{\mathbf{c}} &= (\mathbf{\Psi}'\mathbf{Q}_\phi\mathbf{\Psi} + \lambda\mathbf{R})^{-1}\mathbf{\Psi}'\mathbf{Q}_\phi\mathbf{y} \\ \hat{\mathbf{d}} &= (\mathbf{\Phi}'\mathbf{\Phi})^{-1}\mathbf{\Phi}'[\mathbf{I} - \mathbf{\Psi}(\mathbf{\Psi}'\mathbf{Q}_\phi\mathbf{\Psi} + \lambda\mathbf{R})^{-1}\mathbf{\Psi}']\mathbf{y}\ . \end{aligned} \tag{5.26}$$

5.6.3 Some properties of the estimates

The first term of (5.24) is identical in structure to the error sum of squares criterion $Q(\mathbf{c})$ defined in (4.1), except that both sets of basis functions are used in the expansion. The second term, however, modifies the basis function expansion problem by penalizing the roughness or size in some sense of the ψ-component of the expansion.

The size of the penalty on the ψ-component is controlled by the smoothing parameter λ. In the limit as $\lambda \to 0$, no penalty whatsoever is applied, and the estimates obtained by minimizing the criterion $\texttt{PENSSE}_\lambda$ revert to those obtained by an ordinary basis expansion in the combined basis of ϕ_j and ψ_k. At the other extreme, when $\lambda \to \infty$, the penalty is so severe that ψ-contribution to the roughness penalty is forced to zero; if $\mathbf{R}$ is strictly positive-definite, we obtain the basis function estimate corresponding to the basis $[\phi_j]$ alone. If $\mathbf{R}$ is not strictly positive-definite, then a contribution x_R from the $[\psi_k]$ basis is allowed, provided that it satisfies $Lx_R(s) = 0$ for all s.

It is instructive to study the minimizing values of the coefficient vectors $\mathbf{d}$ and $\mathbf{c}$. The smoothing matrix $\mathbf{S}$ then becomes

$$\mathbf{S}_\lambda = \mathbf{S}_\phi \mathbf{Q}_{\psi,\lambda} + \mathbf{S}_{\psi,\lambda}\mathbf{Q}_\phi,$$

where the smoothing operator $\mathbf{S}_{\psi,\lambda}$ is

$$\mathbf{S}_{\psi,\lambda} = \boldsymbol{\Psi}(\boldsymbol{\Psi}'\mathbf{Q}_\phi\boldsymbol{\Psi} + \lambda\mathbf{R})^{-1}\boldsymbol{\Psi}'. \tag{5.27}$$

and $\mathbf{Q}_{\psi,\lambda} = \mathbf{I} - \mathbf{S}_{\psi,\lambda}$. From this we can see that $\mathbf{S}_{\psi,\lambda}$ is a kind of "sub-projection"' matrix in the metric of the projection $\mathbf{Q}_\phi$ in that it has the structure of a true projection except for a perturbation of $\boldsymbol{\Psi}'\mathbf{Q}_\phi\boldsymbol{\Psi}$ by $\lambda\mathbf{R}$.

Moreover, the fit vector $\hat{\mathbf{y}}$ is now partitioned into two orthogonal parts, $\hat{\mathbf{y}} = \hat{\mathbf{y}}_0 + \hat{\mathbf{y}}_1$, where

$$\begin{aligned} \hat{\mathbf{y}}_0 &= \mathbf{S}_\phi \mathbf{Q}_{\psi,\lambda}\mathbf{y} \\ \hat{\mathbf{y}}_1 &= \mathbf{S}_{\psi,\lambda}\mathbf{Q}_\phi\mathbf{y}. \end{aligned} \tag{5.28}$$

The first "ultra-smooth" term comes from first smoothing $\mathbf{y}$ using rough basis $\boldsymbol{\psi}$, and then projecting the residual from that smooth onto the space spanned by smooth basis $\boldsymbol{\phi}$. The second "rough" term comes from first projecting $\mathbf{y}$ on to the orthogonal complement of the ϕ-space, and then applying the ψ-smoother to the result.

This elaborates the way in which the regularized basis approach provides a continuous range of choices between low-dimensional basis expansion in terms of the functions ϕ_j and a high-dimensional expansion also making use of the functions ψ_k.

5.6.4 Relationship to the roughness penalty approach

We conclude with some remarks about the connections between the regularized basis method and the method discussed in Section 5.3.3 above. Firstly, to minimize the residual sum of squares penalized by $\|Lx\|^2$, we need not specify any functions at all in the ϕ_j part of the basis, but merely ensure that $[\psi_k]$ is a suitable basis for the functions of interest. In the original spline smoothing context, with $L = D^2$, we can take the $[\psi_k]$ to be a B-spline basis with knots at the data points, and, by using suitable

methods of numerical linear algebra, we can obtain a stable $O(n)$ algorithm for spline smoothing; this is the approach of the S-PLUS function `smooth.spline`.

Secondly, if we wish to prescribe a particular ultrasmooth class $\mathcal{F}_0$, the regularized basis approach allows us to choose basis functions ϕ_j to span $\mathcal{F}_0$, and then allow $\mathbf{R}$ to be any appropriate strictly positive-definite matrix. In this way, the choice of the ultrasmooth class is decoupled from the way that roughness is measured.

5.7 Further reading and notes

We drew on the treatment of roughness penalties in Green and Silverman (1994) in preparing this chapter, but possibly the best current reference for fairly advanced readers is Gu (2002). Wahba (1990) reviews the many remarkable contributions of the author and her students to spline smoothing technology, but requires a background in functional analysis to read.

Although we have expressed roughness penalties in terms of integrated squared derivatives, many authors have used the simpler approach of summing squared first or second difference values instead. This only works if the sampling points t_j are equally spaced, but in this context, summing squared differences works well, and is discussed in Eilers and Marx (1996), and also by O'Sullivan (1986) and O'Sullivan, Yandell and Raynor (1986).

Two efforts stand out as path-breaking attempts to use derivative information in data analysis. The first of these is a series of papers on human growth data beginning with Largo et al. (1978) that focussed on the shape of the acceleration function. By careful and innovative use of smoothing techniques, spline and kernel, they were able to isolate a hitherto ignored phenomenon, the so-called mid-spurt, or hump in the acceleration curve that precedes the pubertal growth spurt and occurs at around seven to eight years in almost all children of either gender. These studies confirmed a principle that we have seen in many of our own functional data analyses: Exogenous influences and other interesting events are often much more apparent in some order of derivative than in the original curves.

On a somewhat more technical note, the thesis by Besse (1979) and his subsequent papers (Besse and Ramsay, 1986; Besse, 1980 & 1988) moved the French data analytic school into a new realm involving data that take values in spaces of functions possessing a certain number of derivatives. Besse's discussion of principal components analysis in the context of observations in Sobelev space was inspired by Dauxois and Pousse (1976), Dauxois, Pousse and Romain (1982) and the functional analytic approaches to spline smoothing by Atteia (1965). Ramsay and Dalzell (1991), who coined the term functional data analysis, extended this line of work to linear models.