4

Smoothing functional data by least squares

4.1 Introduction

In this chapter and the next we turn to a discussion of specific smoothing methods. Our goal is to give enough information to those new to the topic of smoothing to launch a functional data analysis. Here we focus on the more familiar technique of fitting models to data by minimizing the sum of squared errors, or *least squares estimation*. This approach ties in functional data analysis with the machinery of multiple regression analysis. A number of tools taken from this classical field are reviewed here, and especially those that arise because least squares fitting defines a model whose estimate is a linear transformation of the data.

The treatment is far from comprehensive, however, and primarily because we will tend to favor the more powerful methods using roughness penalties to be taken up in the next chapter. Rather, notions such as degrees of freedom, sampling variance, and confidence intervals are introduced here as a first exposure to topics that will be developed in greater detail in Chapter 5.

4.2 Fitting data using a basis system by least squares

Recall that our goal is to fit the discrete observations $y_j, j = 1, \ldots, n$ using the model $y_j = x(t_j) + \epsilon_j$, and that we are using a basis function expansion

for $x(t)$ of the form

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

The vector $\mathbf{c}$ of length K contains the coefficients c_k. Let us define the n by K matrix $\boldsymbol{\Phi}$ as containing the values $\phi_k(t_j)$.

4.2.1 Ordinary or unweighted least squares fits

A simple linear smoother is obtained if we determine the coefficients of the expansion c_k by minimizing the least squares criterion

$$\texttt{SMSSE}(\mathbf{y}|\mathbf{c}) = \sum_{j=1}^{n} [y_j - \sum_{k}^{K} c_k \phi_k(t_j)]^2. \tag{4.1}$$

The criterion is expressed more cleanly in matrix terms as

$$\texttt{SMSSE}(\mathbf{y}|\mathbf{c}) = (\mathbf{y} - \boldsymbol{\Phi}\mathbf{c})'(\mathbf{y} - \boldsymbol{\Phi}\mathbf{c}) \ . \tag{4.2}$$

The right side is also often written in functional notation as $\|\mathbf{y} - \boldsymbol{\Phi}\mathbf{c}\|^2$.

Taking the derivative of criterion $\texttt{SMSSE}(\mathbf{y}|\mathbf{c})$ with respect to $\mathbf{c}$ yields the equation

$$2\boldsymbol{\Phi}\boldsymbol{\Phi}'\mathbf{c} - 2\boldsymbol{\Phi}'\mathbf{y} = 0$$

and solving this for $\mathbf{c}$ provides the estimate $\hat{\mathbf{c}}$ that minimizes the least squares solution,

$$\hat{\mathbf{c}} = (\boldsymbol{\Phi}'\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}'\mathbf{y} \ . \tag{4.3}$$

The vector $\hat{\mathbf{y}}$ of fitted values is

$$\hat{\mathbf{y}} = \boldsymbol{\Phi}\hat{\hat{\mathbf{c}}} = \boldsymbol{\Phi}(\boldsymbol{\Phi}'\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}'\mathbf{y} \ . \tag{4.4}$$

Simple least squares approximation is appropriate in situations where we assume that the residuals ϵ_j about the true curve are independently and identically distributed with mean zero and constant variance σ^2. That is, we prefer this approach when we assume the *standard model for error* discussed in Section 3.2.4.

As an example, Figure 4.1 shows the daily temperatures in Montreal averaged over 34 years, 1960–1994, for 101 days in the summer and 101 days in the winter. There is some higher frequency variation that seems to require fitting in addition to the smooth quasi-sinusoidal long-term trend. For example, there is a notable warming period from about January 16 to January 31 that is present in the majority of Canadian weather stations. The smooth fit shown in the figure was obtained with 109 Fourier basis functions, which would permit $108/2 = 54$ cycles per year, or roughly one per week. The curve seems to track nicely these shorter-term variations in temperature.

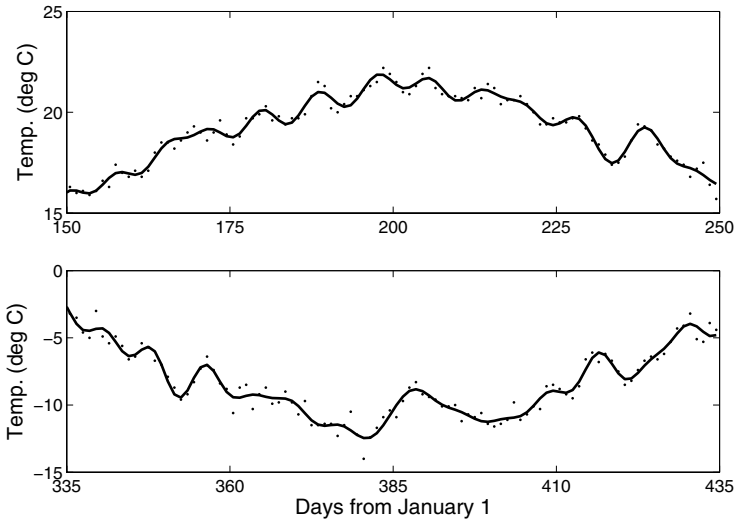

Figure 4.1. The upper panel shows the average daily temperatures for 101 days over the summer in Montreal, and the lower panel covers 101 winter days, with the day values extended into the following year. The solid curves are unweighted least squares smooths of the data using 109 Fourier basis functions.

4.2.2 Weighted least squares fits

As we noted in Section 3.2.4, the standard model for error will often not be realistic. To deal with nonstationary and/or autocorrelated errors, we may need to bring in a differential weighting of residuals by extending the least squares criterion to the form

$$\texttt{SMSSE}(\mathbf{y}|\mathbf{c}) = (\mathbf{y} - \mathbf{\Phi c})'\mathbf{W}(\mathbf{y} - \mathbf{\Phi c}) \tag{4.5}$$

where $\mathbf{W}$ is a symmetric positive definite matrix that allows for unequal weighting of squares and products of residuals.

Where do we get $\mathbf{W}$? If the variance-covariance matrix $\mathbf{\Sigma}_e$ for the residuals ϵ_j is known, then

$$\mathbf{W} = \mathbf{\Sigma}_e^{-1} \ .$$

In applications where an estimate of the complete $\mathbf{\Sigma}_e$ is not feasible, the covariances among errors are often assumed to be zero, and then $\mathbf{W}$ is diagonal with, preferably, reciprocals of the error variance associated with the y_j's in the diagonal. We will consider various ways of estimating $\mathbf{\Sigma}_e$ in Section 4.6.2. But in the meantime, we will not lose anything if we always include the weight matrix $\mathbf{W}$ in results derived from least squares estimation; we can always set it to $\mathbf{I}$ if the standard model is assumed.

The weighted least squares estimate $\hat{\mathbf{c}}$ of the coefficient vector $\mathbf{c}$ is

$$\hat{\mathbf{c}} = (\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi})^{-1}\mathbf{\Phi}'\mathbf{W}\mathbf{y} \ . \tag{4.6}$$

Whether the approximation is by simple least squares or by weighted least squares, we can express what is to be minimized in the more universal functional notation $\texttt{SMSSE}(\mathbf{y}|\mathbf{c}) = \|\mathbf{y} - \mathbf{\Phi}\mathbf{c}\|^2$.

4.3 A performance assessment of least squares smoothing

It may be helpful to see what happens when we apply least squares smoothing to a situation where we know what the right answer is, and can therefore check the quality of various aspects of the fit to the data, as well as the accuracy of data-driven bandwidth selection methods.

We turn now to the growth data, where a central issue was obtaining a good estimate of the acceleration or second derivative of the height function. For example, can we trust the acceleration curves displayed in Figure 1.1?

The parametric growth curve proposed by Jolicoeur (1992) has the following form:

$$h(t) = a\frac{\sum_{\ell=1}^{3}[b_\ell(t+e)]^{c_\ell}}{1+\sum_{\ell=1}^{3}[b_\ell(t+e)]^{c_\ell}}. \tag{4.7}$$

Jolicoeur's model is now known to be a bit too smooth, and especially in the period before the pubertal growth spurt, but it does offer a reasonable account of most growth records for the comparatively modest investment of estimating eight parameters, namely a, e and $(b_\ell, c_\ell), \ell = 1, 2, 3$. The model has been fit to the Fels growth data (Roche, 1992) by R. D. Bock (2000), and from these fits it has been possible to summarize the variation of parameter values for both genders reasonably well using a multivariate normal distribution. The average parameter values are $a = 164.7, e = 1.474, \mathbf{b} = (0.3071, 0.1106, 0.0816)', \mathbf{c} = (3.683, 16.665, 1.474)'$. By sampling from this distribution, we can simulate the smooth part of as many records as we choose.

The standard error of measurement has also been estimated for the Fels data as a function of age by one of the authors, and Figure 4.2 summarizes this relation. We see height measurements are noisier during infancy, where the standard error is about eight millimeters, but by age six or so, the error settles down to about five millimeters. Simulated noisy data were generated from the smooth curves by adding independent random errors having a mean of zero and standard deviation defined by this curve to the smooth values at the sampling points. The reciprocal of the square of this function was used to define the entries of the weight matrix $\mathbf{W}$, which in this case was diagonal. The sampling ages were those of the Berkeley data, namely

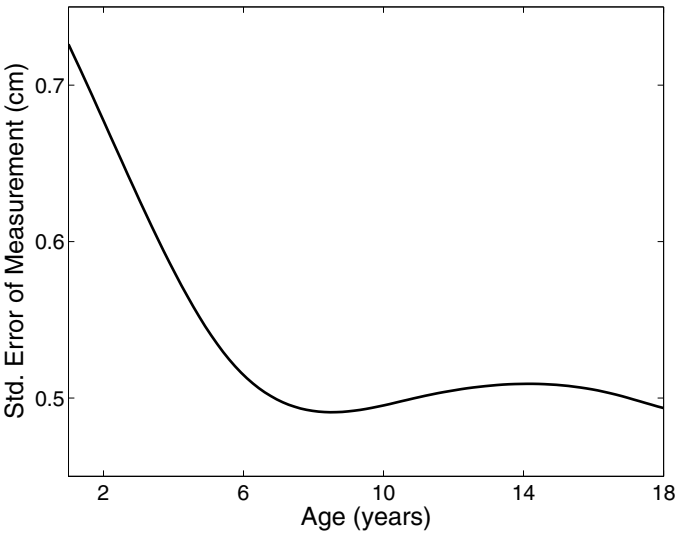

Figure 4.2. The estimated relation between the standard error of height measurements and age for females based on the Fels growth data.

quarterly between one and two years, annually between two to eight years, and twice a year after that to eighteen years of age.

We estimated the growth acceleration function by fitting a single set of data for a female. For the analysis, a set of 12 B-spline basis functions were used of order six and with equally spaced knots. We chose order six splines so that the acceleration estimate would be a cubic spline and hence smooth. A weighted least squares analysis was used with $\mathbf{W}$ being diagonal and with diagonal entries being the reciprocals of the squares of the standard errors shown in Figure 4.2.

Figure 4.3 shows how well we did. The maximum and minimum for the pubertal growth spurt are a little underestimated, and there are some peaks and valleys during childhood that aren't in the true curve. However, the estimate is much less successful at the lower and upper boundaries, and this example is a warning that we will have to look for ways to get better performance in these regions. On the whole, though, the important features in the true acceleration curve are reasonably reflected in the estimate.

4.4 Least squares fits as linear transformations of the data

The smoothing methods described in this chapter all have the property of being *linear*. Linearity simplifies computational issues considerably, and

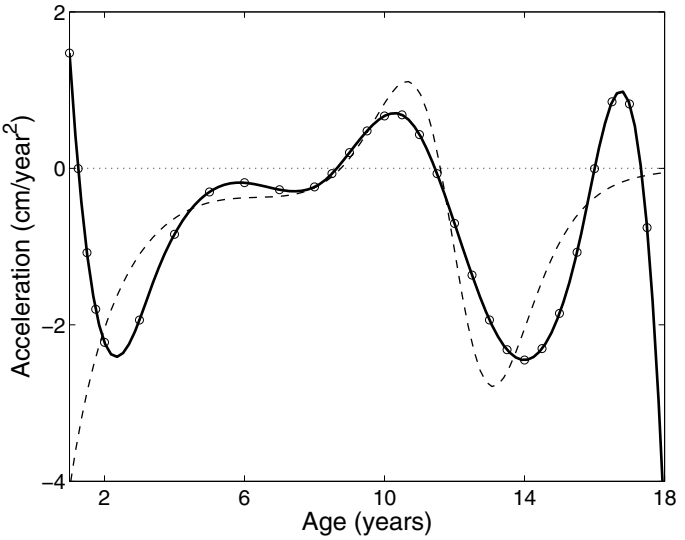

Figure 4.3. The solid curve is the estimated growth acceleration for a single set of simulated data, and the dashed curve is the errorless curve. The circles indicate the ages at which simulated observations were generated.

is convenient in a number of other ways. Most smoothing in practice gets done by linear procedures. Consequently, before we turn to other smoothing methods, we need to consider what linearity in a smoothing procedure means.

4.4.1 How linear smoothers work

A *linear smoother* estimates the function value $\hat{y}_j = \hat{x}(t_j)$ by a linear combination of the discrete observations

$$\hat{x}(t_j) = \sum_{\ell=1}^{n} S_j(t_\ell) y_\ell \ , \tag{4.8}$$

where $S_j(t_\ell)$ weights the ℓth discrete data value in order to generate the fit to y_j.

In matrix terms,

$$\hat{x}(\mathbf{t}) = \mathbf{S}\mathbf{y} \ , \tag{4.9}$$

where $\hat{x}(\mathbf{t})$ is a column vector containing the values of the estimate of function x at each sampling point t_j.

In the unweighted least squares case, for example, we see in (4.4) that

$$\mathbf{S} = \mathbf{\Phi}(\mathbf{\Phi}'\mathbf{\Phi})^{-1}\mathbf{\Phi}'. \tag{4.10}$$

In regression analysis, this matrix is often called the "hat matrix" because it converts the dependent variable vector $\mathbf{y}$ into its fit $\hat{\mathbf{y}}$.

In the context of least squares estimation, the smoothing matrix has the property of being a *projection matrix*. This means that it creates an image of data vector y on the space spanned by the columns of matrix $\mathbf{\Phi}$ such that the residual vector $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$ is orthogonal to the fit vector $\hat{y}$,

$$(\mathbf{y} - \hat{\mathbf{y}})'\hat{\mathbf{y}} = 0 \ .$$

This in turn implies that the smoothing matrix has the property $\mathbf{SS} = \mathbf{S}$, a relation called *idempotency*. In the next chapter on roughness-penalized least squares smoothing, we shall see that property does not hold.

The corresponding smoothing matrix for weighted least squares smoothing is

$$\mathbf{S} = \mathbf{\Phi}(\mathbf{\Phi}'\mathbf{W}\mathbf{\Phi})^{-1}\mathbf{\Phi}'\mathbf{W} \ . \tag{4.11}$$

Matrix $\mathbf{S}$ is still an orthogonal projection matrix, except that now the residual and fit vectors are orthogonal in the sense that

$$(\mathbf{y} - \hat{\mathbf{y}})'\mathbf{W}\hat{\mathbf{y}} = 0 \ .$$

In this case $\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$ is often said to be a *projection in the metric* $\mathbf{W}$.

Figure 4.4 shows the weights associated with estimating the growth acceleration curve in Figure 4.3 for ages six, twelve, and eighteen. For ages away from the boundaries, the weights have a positive peak centered on the age being estimated, and two negative side-lobes. For age twelve in the middle of the pubertal growth spurt for females, the observations receiving substantial weight, of either sign, range from ages seven to seventeen. This is in marked contrast to second difference estimates

$$D^2x(t_j) \approx \left(\frac{y_{j+1} - y_j}{t_{j+1} - t_j} - \frac{y_j - y_{j-1}}{t_j - t_{j+1}}\right)/(t_{j+1} - t_{j-1}),$$

which would only use three adjacent ages.

At the upper boundary, we see why there is likely to be considerable instability in the estimate. The final observation receives much more weight than any other value, and only observations back to age fifteen are used at all. The boundary estimate pools much less information than do interior estimates, and is especially sensitive to the boundary observations.

Many widely used smoothers are linear. The linearity of a smoother is a desirable feature for various reasons: The linearity property

$$\mathbf{S}(a\mathbf{y} + b\mathbf{z}) = a\mathbf{S}\mathbf{y} + b\mathbf{S}\mathbf{z}$$

is important for working out various properties of the smooth representation, and the simplicity of the smoother implies relatively fast computation. On the other hand, some nonlinear smoothers may be more adaptive to different behavior in different parts of the range of observation, and may be robust to outlying observations. Smoothing by the thresholded

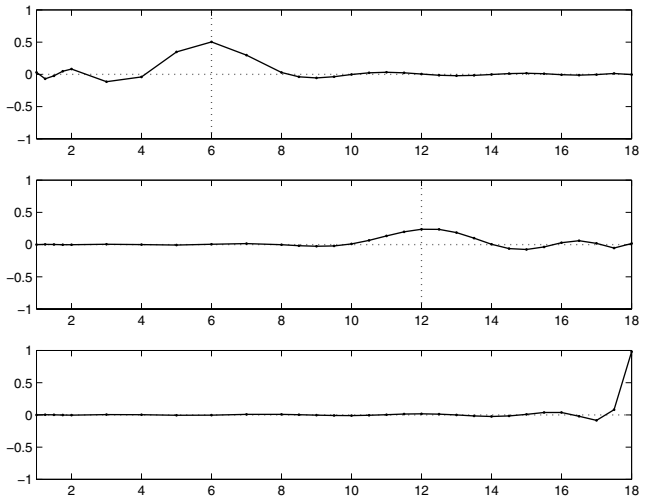

Figure 4.4. The top panel indicates how observations are weighted in order to estimate growth acceleration at age six in figure 4.3. The central panel shows the weights for age twelve, and the bottom for age eighteen. The dots indicate the ages at which simulated observations were generated.

wavelet transform, discussed in Section 3.6.1, is an important example of a nonlinear smoothing method.

Speed of computation can be critical; a smoother that is useful for a few hundred data points can be completely impractical for thousands. Smoothers that require a number of operations that is proportional to n to compute n smoothed values $\hat{x}(s_j)$, abbreviated $O(n)$ operations, are virtually essential for large n. If $\mathbf{S}$ is band-structured, meaning that only a small number K of values on either side of its diagonal in any row are nonzero, then $O(n)$ computation is assured.

4.4.2 The degrees of freedom of a linear smooth

We are familiar with the idea that the model for observed data offers an image of the data that has fewer *degrees of freedom* than are present in the original data. In most textbook situations, the concept of the degrees of freedom of a fit means simply the number of parameters estimated from the data that are required to define the model.

The notion of degrees of freedom applies without modification to data smoothing using least squares, where the number of parameters is the length K of the coefficient vector $\mathbf{c}$. The number of *degrees of freedom for error* is therefore $n - K$.

When we begin to use roughness penalty methods in Chapter 5, however, things will not be so simple, and we will need a more general way of computing the effective degrees of freedom of a smooth fit to the data, and consequently the corresponding degrees of freedom for error. We do this by using the "hat" matrix $\mathbf{S}$ by defining the degrees of freedom of the smooth fit to be

$$df = \text{trace}\,\mathbf{S} \tag{4.12}$$

where the trace of a square matrix means the sum of its diagonal elements. This more general definition yields exactly K for least squares fits, and therefore does not represent anything new. But this definition will prove invaluable in our later chapters.

There are also situations in which it may be more appropriate to use the alternative definition

$$df = \text{trace}\,(\mathbf{S}\mathbf{S}') \tag{4.13}$$

but most of the time (4.12) is employed. In any case, the two definitions give the same answer for least squares estimation.

4.5 Choosing the number K of basis functions

How do we choose the order of the expansion K? The larger K, the better the fit to the data, but of course we then risk also fitting noise or variation that we wish to ignore. On the other hand, if we make K too small, we may miss some important aspects of the smooth function x that we are trying to estimate.

4.5.1 The bias/variance trade-off

This trade-off can be expressed in another way. For large values of K, n the *bias* in estimating $x(t)$, that is

$$\texttt{Bias}[\hat{x}(t)] = x(t) - \text{E}[\hat{x}(t)], \tag{4.14}$$

is small. In fact, if the notion of additive errors having expectation zero expressed in (3.1) holds, then we know that the bias will be zero for $K = n$.

But of course, that is only half of the story. One of the main reasons that we do smoothing is to reduce the influence of noise or ignorable variation on the estimate $\hat{x}$. Consequently we are also interested in the *variance of estimate*

$$\texttt{Var}[\hat{x}(t)] = \text{E}[\{\hat{x}(t) - \text{E}[\hat{x}(t)]\}^2]\ . \tag{4.15}$$

If $K = n$, this is almost certainly going to be unacceptably high. Reducing variance leads us to look for smaller values of K, but of course not so small

as to make the bias unacceptable. The worse the signal-to-noise ratio in the data, the more reducing sampling variance will outweigh controlling bias.

One way of expressing what we really want to achieve is *mean-squared error*

$$\texttt{MSE}[\hat{x}(t)] = \mathrm{E}[\{\hat{x}(t) - x(t)\}^2] \ , \tag{4.16}$$

also called the $\mathcal{L}^2$ *loss function*. In most applications we can't actually minimize this since we have no way of knowing what $x(t)$ is without using the data. However, one of the most important equations in statistics links mean squared error to bias and sampling variance by the simple additive decomposition

$$\texttt{MSE}[\hat{x}(t)] = \texttt{Bias}^2[\hat{x}(t)] + \texttt{Var}[\hat{x}(t)] \ . \tag{4.17}$$

What this relation tells us is that it would be worthwhile to tolerate a little bias if the result is a big reduction in sampling variance. In fact, this is almost always the case, and is the fundamental reason for smoothing data in order to estimate functions. We will return to this matter in Chapter 5.

Figure 4.5 shows some total squared error measures as a function of various numbers of basis functions. The measures were computed by summing mean squared error, sampling variance and squared bias across the ages ranging from three to sixteen. This range was used to avoid ages near the boundaries, where the curve estimates tend to have much greater error levels. The results are based on smoothing 10,000 random samples constructed in the same manner as that in Figure 4.3.

Notice that the measures for sampling variance and squared bias sum to those for mean squared error, as in (4.17). Sampling variance increases rapidly when we use too many basis functions, but squared bias tends to decay more gently to zero at the same time. We see there that the best results for totaled mean squared error are obtained with ten and twelve basis functions, and we broke the tie by opting for the result with the least bias.

It may see surprising that increasing K does not always decrease bias. If so, recall that, when the order of a spline is fixed and knots are equally spaced, K B-splines do not span a space that lies within that defined by $K+1$ B-splines. Complicated effects due to knot spacing relative to sampling points can result in a lower-dimensional B-spline system actually producing better results than a higher-dimensional system.

Although the decomposition mean squared error (4.17) is helpful for expressing the bias/variance tradeoff in a neat way, the principle applies more widely. In fact, there are many situations where it is preferable to use other loss functions. For example, minimizing $\mathrm{E}[|\hat{x}(t) - x(t)|]$, called the $\mathcal{L}^1$ norm, is more effective if the data contain outliers. For this and nearly any fitting criterion or loss function for smoothing, we can assume that when bias goes down, sampling variance goes up, and some bias must be tolerated to achieve a stable estimate of the smooth trend in the data.

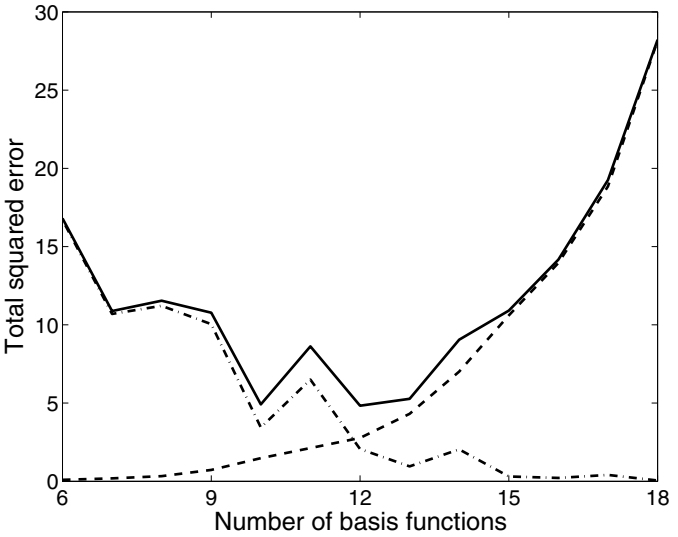

Figure 4.5. The heavy solid line indicates mean squared error totaled across the ages of observation between three and sixteen. The dashed line shows the totaled sampling variance, and the dotted-dashed line shows the totaled squared bias.

4.5.2 Algorithms for choosing K

The vast literature on multiple regression contains many ideas for deciding how many basis functions to use. For example, *stepwise variable selection* would proceed in a step-up fashion by adding basis functions one after another, testing at each step whether the added function significantly improves fit, and also checking that the functions already added continue to play a significant role. Conversely, *variable-pruning* methods are often used for high-dimensional models, and work by starting with a generous choice of K and dropping a basis function on each step that seems to not account for a substantial amount of variation.

These methods all have their limitations, and are often abused by users who do not appreciate these problems. The fact that there is no one gold standard method for the variable selection problem should warn us at this point that we face a difficult task in attempting to fix model dimensionality. The discrete character of the K-choice problem is partly to blame, and the methods described in Chapter 5 providing a continuum of smoothing levels will prove helpful.

4.6 Computing sampling variances and confidence limits

4.6.1 Sampling variance estimates

The estimation of the coefficient vector $\mathbf{c}$ of the basis function expansion $x = \mathbf{c}'\boldsymbol{\phi}$ by minimizing least squares defines a linear mapping (4.6) from the raw data vector $\mathbf{y}$ to the estimate. With this mapping in hand, it is a relatively simple matter to compute the sampling variance of the coefficient vector, and of anything that is linearly related to it.

We begin with the fact that if a random variable y is normally distributed with a variance-covariance matrix $\boldsymbol{\Sigma}_y$, then the random variable $\mathbf{Ay}$ defined by any matrix $\mathbf{A}$ has the variance-covariance matrix

$$\mathtt{Var}[\mathbf{Ay}] = \mathbf{A}\boldsymbol{\Sigma}_y\mathbf{A}' \ . \tag{4.18}$$

Now in this and other linear modelling situations that we will encounter, the model for the data vector $\mathbf{y}$, in this case $x(\mathbf{t})$, is regarded as a fixed effect having zero variance. Consequently, the variance-covariance matrix of y using the model $\mathbf{y} = x(\mathbf{t}) + \boldsymbol{\epsilon}$ is the variance-covariance matrix $\boldsymbol{\Sigma}_e$ of the residual vector $\boldsymbol{\epsilon}$. We must in some way use the information in the actual residuals to replace the population quantity $\boldsymbol{\Sigma}_e$ by a reasonable sample estimate $\hat{\boldsymbol{\Sigma}}_e$.

For example, to compute the sampling variances and covariances of the coefficients themselves in $\mathbf{c}$, we use that fact that in this instance

$$\mathbf{A} = (\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}'\mathbf{W} \ .$$

to obtain

$$\mathtt{Var}[\mathbf{c}] = (\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Sigma}_e\mathbf{W}\boldsymbol{\Phi}(\boldsymbol{\Phi}'\mathbf{W}\boldsymbol{\Phi})^{-1} \ . \tag{4.19}$$

When the standard model is assumed, $\boldsymbol{\Sigma}_e = \sigma^2\mathbf{I}$, and if unweighted least squares is used, then we obtain the simpler result that appears in textbooks on regression analysis

$$\mathtt{Var}[\mathbf{c}] = \sigma^2(\boldsymbol{\Phi}'\boldsymbol{\Phi})^{-1} \ . \tag{4.20}$$

However, in our functional data analysis context there will seldom be much interest in interpreting the coefficient vector $\mathbf{c}$ itself. Rather, we will want to know the sampling variance of some quantity computed from these coefficients. For example, we might want to know the sampling variance of the the fit to the data defined by $x(t) = \boldsymbol{\phi}(t)'\mathbf{c}$. Since we now have in hand the sampling variance of $\mathbf{c}$ through (4.19) or (4.20), we can simply apply result (4.18) again to get

$$\mathtt{Var}[\hat{x}(t)] = \boldsymbol{\phi}(t)'\mathtt{Var}[\mathbf{c}]\boldsymbol{\phi}(t) \tag{4.21}$$

and the variances of all the fitted values corresponding to the sampling values t_j are in the diagonal of the matrix

$$\mathtt{Var}[\hat{\mathbf{y}}] = \mathbf{\Phi}\mathtt{Var}[\mathbf{c}]\mathbf{\Phi}'$$

which, in the standard model/unweighted least squares case, and using (4.10), reduces to

$$\mathtt{Var}[\hat{\mathbf{y}}] = \sigma^2\mathbf{\Phi}(\mathbf{\Phi}'\mathbf{\Phi})^{-1}\mathbf{\Phi}' = \sigma^2\mathbf{S} \ .$$

4.6.2 Estimating $\mathbf{\Sigma}_e$

Clearly our estimates of sampling variances are only as good as our estimates of the variances and covariances among the residuals ϵ_j.

When we are smoothing a single curve, the total amount of information involved is insufficient for much more than estimating either a single constant variance σ^2 assuming the standard model for error, or at most a variance function with values $\sigma^2(t)$, that has fairly mild variation over t. It is important to use methods which produce relatively unbiased estimate of variance in order to avoid underestimating sampling variance. For example, if the standard model for error is accepted,

$$s^2 = \frac{1}{n-K}\sum_j^n (y_j - \hat{y}_j)^2 \tag{4.22}$$

is much preferred as an estimate of σ^2 than the maximum likelihood estimate that involves dividing by n. In fact, we shall see in the next chapter that this estimate is related to a popular more general method for choosing smoothing level called *generalized cross-validation.*

One reasonable strategy for choosing K is to add basis functions until s^2 fails to decrease substantially. Figure 4.6 shows how s decreases to a value of about 0.56 degrees Celsius by the time we use 109 Fourier basis functions for smoothing the Montreal temperature data shown in Figure 4.1. There are places where s^2 is even lower, but we worried that the minimum at 240 basis functions corresponded to over-fitting the data.

A common strategy for estimating at least a limited number of covariances in $\mathbf{\Sigma}_e$ given a small N, or even $N = 1$, is to assume an autoregressive (AR) structure for the residuals. This is often realistic, since adjacent residuals are frequently correlated because they are mutually influenced by unobserved variables. For example, the weather on one day is naturally likely to be related to the weather on the previous day because of the influence of large slow-moving low or high pressure zones. An intermediate level text on regression analysis such as Draper and Smith (1998) can be consulted for details on how to estimate AR structures among residuals.

When a substantial number N of replicated curves are available, as in the growth curve data and Canadian weather data, we can attempt more sophisticated and detailed estimates of $\mathbf{\Sigma}_e$. For example, we may opt for

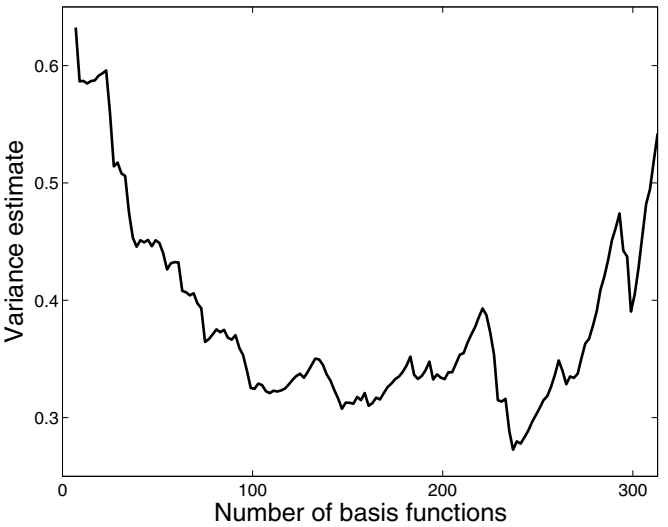

Figure 4.6. The relation between the number of Fourier basis functions and the unbiased estimate of the residual variance (4.22) in fitting the Montreal temperature data.

estimating the entire variance-covariance matrix from the N by n matrix $\mathbf{E}$ of residuals by

$$\hat{\mathbf{\Sigma}}_e = (N-1)^{-1}\mathbf{E}'\mathbf{E}.$$

However, even then, an estimate of a completely unrestricted $\mathbf{\Sigma}_e$ requires the estimation of $n(n-1)/2$ variances and covariances from N replications, and it is unlikely that data with the complexity of the daily weather records would ever have N sufficiently large to do this accurately.

4.6.3 Confidence limits

Confidence limits are typically computed by adding and subtracting a multiple of the standard errors, that is, the square root of the sampling variances, to the actual fit. For example, 95% limits correspond to about two standard errors up and down from a smooth fit. These standard errors are estimated using (4.21). Confidence limits on fits computed in this way are called *point-wise* because they reflect confidence regions for *fixed* values of t rather than regions for the curve as a whole.

Figure 4.7 shows the temperatures during the 16 days over which the January thaw takes place in Montreal, along with the smooth to the data and 95% point-wise confidence limits on the fit. The standard error of the estimated fit was 0.26 degrees Celsius.

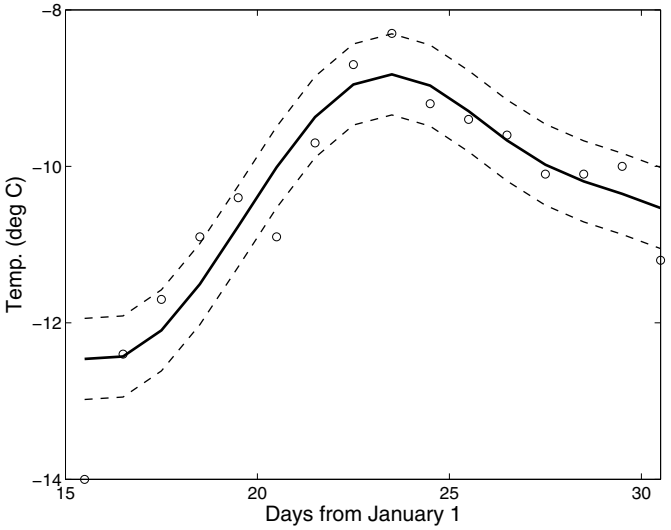

Figure 4.7. The temperatures over the mid-winter thaw for the Montreal temperature data. The solid line is the smooth curve estimated in Figure 4.1 and the lower and upper dashed lines are estimated 95% point-wise confidence limits for this fit.

We will have much to say in the next chapter and elsewhere about the hazards of placing too much faith in sampling variances and confidence limits estimated in these ways. But we should at least note two important ways in which confidence limits computed in this way may be problematic. First, it is implicitly assumed that K is a fixed constant, but the reality is that K for smoothing problems is more like a parameter estimated from the data, and consequently the size of these confidence limits does not reflect the uncertainty in our knowledge of K. Secondly, the smooth curve to which we add and subtract multiples of the standard error to get point-wise limits is itself subject to bias, and especially in regions of high curvature. We can bet, for example, that the solid curve in Figure 4.7 is too low on January 23rd, the center of the January thaw. Thus, the confidence limits calculated in this way are themselves biassed, and the region covered by them may not be quite as advertised.

4.7 Fitting data by localized least squares

For a smoothing method to make any sense at all, the value of the function estimate at a point t must be influenced mostly by the observations near t. This feature is an implicit property of the estimators we have considered

so far. In this section, we consider estimators where the local dependence is made more explicit by means of local weight functions.

Keeping within the domain of linear smoothing means that our estimate of the value of function x at argument t_j is of the form

$$x(t_j) = \sum_{\ell}^{n} w_\ell y_\ell \ .$$

It seems intuitively reasonable that the weights w_ℓ will only be relatively large for sampling values t_ℓ fairly close to the target value t_j. And, indeed, this tends to hold for the basis function smoothers (4.10) and (4.11).

We now look at smoothing methods that make this *localized weighting principle* explicit. The localizing weights w_j are simply constructed by a location and scale change of a *kernel* function with values $\texttt{Kern}(u)$. This kernel function is designed to have most of its mass concentrated close to 0, and to either decay rapidly or disappear entirely for $|u| \geq 1$. Three commonly used kernels are

Uniform:	$\texttt{Kern}(u) = 0.5$ for $\lvert u\rvert \leq 1$,	0 otherwise
Quadratic:	$\texttt{Kern}(u) = 0.75(1-u^2)$ for $\lvert u\rvert \leq 1$,	0 otherwise
Gaussian:	$\texttt{Kern}(u) = (2\pi)^{-1/2}\exp(-u^2/2)$.	

If we then define weight values to be

$$w_\ell(t) = \texttt{Kern}\left(\frac{t_\ell - t_j}{h}\right) \ , \tag{4.23}$$

then substantially large values $w_\ell(t)$ as a function of ℓ are now concentrated for t_ℓ in the vicinity of t_j. The degree of concentration is controlled by the size of h. The concentration parameter h is usually called the *bandwidth* parameter, and small values imply that only observations close to t receive any weight, while large h means that a wide-sweeping average uses values that are a considerable distance from t.

4.7.1 Kernel smoothing

The simplest and classic case of an estimator that makes use of local weights is the *kernel estimator*. The estimate at a given point is a linear combination of local observations,

$$\hat{x}(t) = \sum_{j}^{n} S_j(t) y_j \tag{4.24}$$

for some suitably defined weight functions S_j. Probably the most popular kernel estimator the Nadaraya-Watson estimator (Nadaraya, 1964; Watson,

1964) is constructed by using the weights

$$S_j(t) = \frac{\texttt{Kern}[(t_j - t)/h]}{\sum_r \texttt{Kern}[(t_r - t)/h]}. \tag{4.25}$$

Although the weight values $w_j(t)$ for the Nadaraya-Watson method are normalized to have a unit sum, this is not essential. The weights developed by Gasser and Müller (1979, 1984) are constructed as follows:

$$S_j(t) = \frac{1}{h} \int_{\bar{t}_{j-1}}^{\bar{t}_j} \texttt{Kern}(\frac{u - t}{h})\ du, \tag{4.26}$$

where $\bar{t}_j = (t_{j+1} + t_j)/2, 1 < j < n, \bar{t}_0 = t_1$ and $\bar{t}_n = t_n$. These weights are faster to compute, deal more sensibly with unequally spaced arguments, and have good asymptotic properties.

The need for fast computation favors the compact support uniform and quadratic kernels, and the latter is the most efficient when only function values are required and the true underlying function x is twice-differentiable. The Gasser-Müller weights using the quadratic kernel are

$$S_j(t) = \frac{1}{4}[\{3r_{j-1}(t) - r_{j-1}^3(t)\} - \{3r_j(t) - r_j^3(t)\}]$$

for $|t_j - t| \leq h$ and 0 otherwise, and where

$$r_j(t) = \frac{t - \bar{t}_j}{h}. \tag{4.27}$$

We need to take special steps if t is within h units of either t_1 or t_n. These measures can consist of simply extending the data beyond this range in some reasonable way, making h progressively smaller as these limits are approached, or sophisticated modifications of the basic kernel function `Kern`. The problem that all kernel smoothing algorithms have of what to do near the limits of the data is one of their major weaknesses, and especially when h is large relative to the sampling rate.

Estimating the derivative just by taking the derivative of the kernel smooth is not usually a good idea, and in any case kernels such as the uniform and quadratic are not differentiable. However, kernels specifically designed to estimate a derivative of fixed order can be constructed by altering the nature of kernel function `Kern`. For example, a kernel $\texttt{Kern}(u)$ suitable for estimating the first derivative must be zero near $u = 0$, positive above zero, and negative below, so that it is a sort of smeared-out version of the first central difference. The Gasser-Müller weights for the estimation of the first derivative are

$$S_j(t) = \frac{15}{16h}[\{r_{j-1}^4(t) - 2r_{j-1}^2(t)\} - \{r_j^4(t) - 2r_j^2(t)\}] \tag{4.28}$$

and for the second derivative are

$$S_j(t) = \frac{105}{16h^2}[\{2r_{j-1}^3(t) - r_{j-1}^5(t) - r_{j-1}(t)\} - \{2r_j^3(t) - r_j^5(t) - r_j(t)\}] \tag{4.29}$$

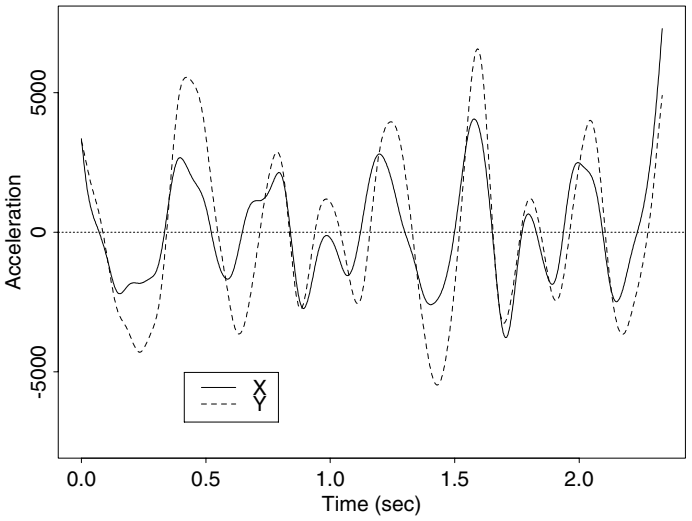

Figure 4.8. The second derivative or acceleration of the coordinate functions for the handwriting data. Kernel smoothing was used with a bandwidth $h = 0.075$.

for $|t_j - t| \leq h$ and 0 otherwise. It is usual to need a somewhat larger value of bandwidth h to estimate derivatives than is required for estimating the function.

Figure 4.8 shows the estimated second derivative or acceleration for the two handwriting coordinate functions. After inspection of the results produced by a range of bandwidths, we settled on $h = 0.075$. This implies that any smoothed acceleration value is based on about 150 milliseconds of data and about 90 values of y_j.

4.7.2 Localized basis function estimators

The ideas of kernel estimators and basis function estimators can, in a sense, be combined to yield *localized basis function estimators*, which encompass a large class of function and derivative estimators. The basic idea is to extend the least squares criterion (4.1) to give a local measure of error as follows:

$$\texttt{SMSSE}_t(\mathbf{y}|\mathbf{c}) = \sum_{j=1}^{n} w_j(t)[y_j - \sum_{k=1}^{K} c_k \phi_k(t_j)]^2, \tag{4.30}$$

where the weight functions w_j are constructed from the kernel function using (4.23).

In matrix terms,

$$\texttt{SMSSE}_t(\mathbf{y}|\mathbf{c}) = (\mathbf{y} - \mathbf{\Phi c})' \mathbf{W}(t)(\mathbf{y} - \mathbf{\Phi c}), \tag{4.31}$$

where $\mathbf{W}(t)$ is a diagonal matrix containing the weight values $w_j(t)$ in its diagonal. Don't be confused by the formal similarity of this expression with (4.5); the matrix $\mathbf{W}(t)$ plays a very different role here.

Choosing the coefficients $\mathbf{c}(t)$ to minimize $\mathtt{SMSSE}_t$ yields

$$\hat{\mathbf{c}}(t) = [\mathbf{\Phi}'\mathbf{W}(t)\mathbf{\Phi}]^{-1}\mathbf{\Phi}'\mathbf{W}(t)\mathbf{y},$$

and substituting back into the expansion $\hat{x}(t) = \sum_{k=1}^{K} \hat{c}_k\phi_k(t)$ gives a linear smoothing estimator of the form (4.8) with smoothing weight values $S_j(t)$ being the elements of the vector

$$S(t) = \mathbf{W}(t)\mathbf{\Phi}[\mathbf{\Phi}'\mathbf{W}(t)\mathbf{\Phi}]^{-1}\boldsymbol{\phi}(t), \tag{4.32}$$

where $\boldsymbol{\phi}(t)$ is the vector with elements $\phi_k(t)$.

The weight values $w_j(t)$ in (4.30) are designed to have substantially nonzero values only for observations located close to the evaluation argument t at which the function is to be estimated. This implies that only the elements in $S(t)$ in (4.32) associated with data arguments values t_j close to evaluation argument t are substantially different from zero, and consequently that $\hat{x}(t)$ is essentially a linear combination of only the observations y_j in the neighborhood of t.

Since the basis has only to approximate a limited segment of the data surrounding t, the basis can do a better job of approximating the local features of the data and, at the same time, we can expect to do well with only a small number K of basis functions. The computational overhead for a single t depends on the number of data argument values t_j for which $w_j(t)$ is nonzero, as well as on K. Both of these are typically small. However, the price we pay for this flexibility is that the expansion must essentially be carried out anew for each evaluation point t.

4.7.3 Local polynomial smoothing

It is interesting to note that the Nadaraya-Watson kernel estimate can be obtained as a special case of the localized basis expansion method by setting $K = 1$ and $\phi_i(t) = 1$. A popular class of methods is obtained by extending from a single basis function to a low order polynomial basis. Thus we choose the estimated curve value $\hat{x}(t)$ to minimize the localized least squares criterion

$$\mathtt{SMSSE}_t(\mathbf{y}|\mathbf{c}) = \sum_{j=1}^{n} \mathtt{Kern}_h(t_j, t)[y_j - \sum_{\ell=0}^{L} c_\ell(t - t_j)^\ell]^2 \ . \tag{4.33}$$

Setting $L = 0$, we recover the Nadaraya-Watson estimate. For values of $L \geq 1$, the function value and L of its derivatives can be estimated by the corresponding derivatives of the locally fitted polynomial at t. In general, the value of L should be at least one and preferably two higher than the highest order derivative required.

Local polynomial smoothing has a strong appeal; see, for example, the detailed discussion provided by Fan and Gijbels (1996). Its performance is superior in the region of the boundaries, and it adapts well to unequally spaced argument values. Local linear expansions give good results when we require only an estimate of the function value. They can easily be adapted in various ways to suit special requirements, such as robustness, monotonicity and adaptive bandwidth selection.

4.7.4 Choosing the bandwidth h

In all the localized basis expansion methods we have considered, the primary determinant of the degree of smoothing is the bandwidth h, rather than the number of basis functions used. The bandwidth controls the balance between two considerations: bias and variance in the estimate. Small values of h imply that the expected value of the estimate $\hat{x}(t)$ must be close to the true value $x(t)$, but the price we pay is in terms of the high variability of the estimate, since it is be based on comparatively few observations. On the other hand, variability can always be decreased by increasing h, although this is inevitably at the expense of higher bias, since the values used cover a region in which the function's shape varies substantially. Mean squared error at t, which is the sum of squared bias and variance, provides a composite measure of performance.

There is a variety of data-driven automatic techniques for choosing an appropriate value of h, usually motivated by the need to minimize mean squared error across the estimated function. Unfortunately, none of these can always be trusted, and the problem of designing a reliable data-driven bandwidth selection algorithm continues to be a subject of active research and considerable controversy. Our own view is that trying out a variety of values of h and inspecting the consequences graphically remains a suitable means of resolving the bandwidth selection problem for most practical problems.

4.7.5 Summary of localized basis methods

Explicitly localized smoothing methods such as kernel smoothing and local polynomial smoothing are easy to understand and have excellent computational characteristics. The role of the bandwidth parameter h is obvious, and as a consequence it is even possible to allow h to adapt to curvature variation. On the negative side, however, is the instability of these methods near the boundaries of the interval, although local polynomial smoothing performs substantially better than kernel smoothing in this regard. As with unweighted basis function expansions, it is well worthwhile to consider matching the choice of basis functions to known characteristics of the data, especially in regions where the data are sparse, or where they are asymmetrically placed around the point t of interest, for example near

the boundaries. The next chapter on the roughness penalty approach looks at the main competitor to kernel and local polynomial methods: spline smoothing.

4.8 Further reading and notes

This chapter and the next are so tightly related that you may prefer to read on, and then consider these notes along with those found there.

Much of the material in this chapter is an application of multiple regression, and references such as Draper and Smith (1998) are useful supplements, and especially on other ways of estimating residual covariance structures.

For more complete treatments of data smoothing, we refer the reader to sources such as Eubank (1999), Green and Silverman (1994), Härdle (1990) and Simonoff (1996). Fan and Gijbels (1996) and Wand and Jones (1995) focus more on kernel smoothing and local polynomial methods. Hastie and Tibshirani (1990) use smoothing methods in the context of estimating the generalized additive or GAM model, but their account of smoothing is especially accessible. Data smoothing also plays a large role in data mining and machine learning, and Hastie, Tibshirani and Friedman (2001) is a recent reference on these topics.

We use spline expansions by fixing the knot locations in advance of the analysis, and optimizing fit with respect to the coefficients multiplying the spline basis functions defined by this fixed knot sequence. The main argument for regarding knots as fixed is computational convenience, but there is also a large literature on using the data to estimate knot locations. Such splines are often called *free-knot splines*. The least squares fitting criterion is highly nonlinear in knot locations, and the computational challenges are severe. Nevertheless, in certain applications where strong curvature is localized in regions not known in advance, this is the more natural approach. For recent contributions to free-knot spline model estimation, see Lindstrom (2002), Lindstrom and Kotz (2004) and Mao and Zhao (2003).

We hope that we have not left the reader with the impression that least squares estimation is the only way to do smoothing. One of the most important developments in statistics in recent years has been the development of *quantile regression* methods by R. Koenker and S. Portnoy, where the model estimates a quantile of the conditional distribution of the dependent variable. Least squares methods, by contrast, attempt to estimate the mean of this distribution. Quantile regression minimizes the sum of absolute values of residuals rather than their sum of squares. Koenker and Portnoy (1994) applied quantile regression to the spline smoothing problem.