2

Bases, Penalties, and Likelihoods

P-splines combine two simple ideas: regression on (many) B-splines and a difference penalty on their coefficients. The B-splines are local functions, each of them covering only a small part of the x-axis. They can give a very flexible fit to data. To keep a fitted curve from getting too flexible, the penalty comes in. It lets adjacent coefficients "hold hands," encouraging a smooth fit. Once the number of B-splines has been set, a single parameter, λ, tunes smoothness.

P-splines have many interesting and useful properties. Interpolation and extrapolation of the fitted curve are automatic. Standard errors and derivatives of the fitted curve are easy to compute.

With a heavy penalty a polynomial curve fit is obtained, creating a bridge between semi-parametric and classic parametric models. But in essence, P-splines are parametric models. The coefficients have a very clear interpretation and can be presented graphically as the skeleton of the fitted curve.

P-splines are grounded in linear regression. Extensions to generalized linear regression are straightforward through penalized likelihood. Counts and binomial data can be handled in an elegant way.

The penalty makes the number of B-splines irrelevant, as long as it is large enough. With many of them, say 10 to 50, the number of coefficients in the model is moderately large. Yet, as will be shown, the effective dimension of the model will be (much) smaller than this number, depending on the amount of smoothing.

2.1 Linear and Polynomial Regression

Consider a scatterplot of data pairs (x_i, y_i), $i = 1:m$. Figure 2.1 displays $m = 111$ daily readings of wind speed (mph) (x-axis) and the maximum of daily ozone (ppm) (y-axis) in New York City (data set `airquality` in `R`).

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

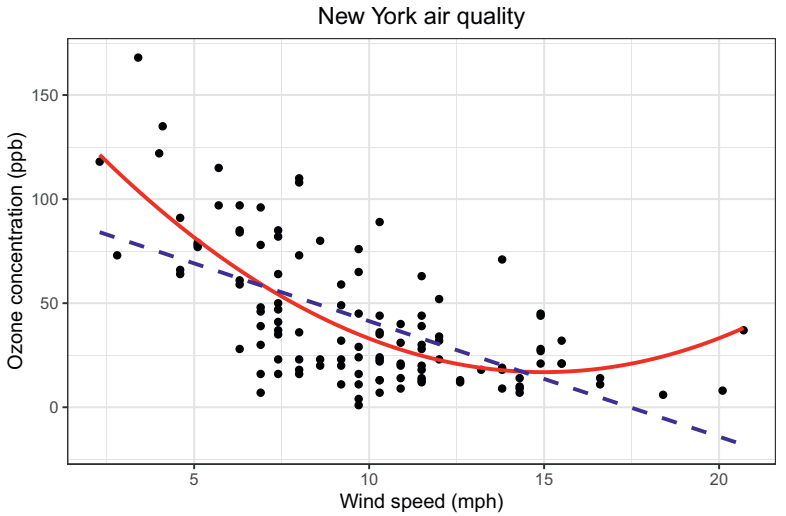

Figure 2.1 Air pollution in New York: scatterplot of daily maximum ozone concentrations and wind speed. Least squares linear (blue broken line) and quadratic (solid red curve) fits. R code in `f-air-wind.R`

The straight blue broken line shows the linear least squares fit, while the solid red line shows the quadratic fit.

The formula for a quadratic curve is $\mu_i = \alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2$. The vector $\alpha = [\alpha_0\, \alpha_1\, \alpha_2]'$ that gives the "best" fit to the data is found by minimizing the least squares objective

$$S = \sum_{i=1}^{m} (y_i - \mu_i)^2 = \sum_{i=1}^{m} (y_i - \alpha_0 - \alpha_1 x_i - \alpha_2 x_i^2)^2.$$

It is easier to work in matrix notation. For the above quadratic curve, we express the m by 3 regressor matrix B, the 3 by 1 unknown parameter vector α, and the m by 1 mean vector μ as

$$B = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 \end{bmatrix} \quad \alpha = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} \quad \mu = B\alpha, \tag{2.1}$$

respectively. Now the least squares solution minimizes the objective

$$S = ||y - B\alpha||^2, \tag{2.2}$$

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

defined as the squared-norm $(y - B\alpha)'(y - B\alpha)$. This leads to the normal equations $B'B\alpha = B'y$ or $\hat{\alpha} = (B'B)^{-1}B'y$.

A more general setting introduces a vector of weights, w. They can reflect known precisions of the data, or w can contain zeros and ones, where a zero indicates a missing y. Using such weights, missingness can be introduced deliberately, e.g., to (temporarily) exclude selected observations. It is more convenient than excluding rows of B and y. With $W = \text{diag}(w)$, we get

$$S = (y - B\alpha)'W(y - B\alpha), \tag{2.3}$$

and the normal equations $B'WB\hat{\alpha} = B'Wy$, with solution $\hat{\alpha} = (B'WB)^{-1}B'Wy$.

The regression scheme can be extended to higher powers of x by adding columns in B with third, fourth, or higher powers. In theory, the computation of $\hat{\alpha}$ does not change, but in practice one has to center and scale x to avoid numerical instabilities. Modern regression software overcomes this issue by using specialized algorithms, like the QR decomposition (Wood, 2017). We will not get into the details of the QR decomposition here. After showing that high-degree polynomial curve fits have serious and fundamental problems, we will discard them as a general smoothing tool.

More generally, the nth degree polynomial model is

$$\mu_i = \alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2 + \alpha_3 x_i^3 + \cdots + \alpha_n x_i^n,$$

resulting in $n + 1$ columns in the matrix B, augmenting (2.1) to powers of n. This again gives $\mu = B\alpha$ in matrix notation. We call B a basis matrix and the powers of x the basis functions.

Figure 2.2 shows data from a simulated motorcycle crash, with a complicated trend: it is a time series of the acceleration of a helmet (Härdle, 1992). These data have become a workbench data set and a rite of passage for many smoothing techniques. A polynomial of low degree has no chance to fit these data well, so we try degree 9 (an arbitrary choice). Two fits are provided: one where all data were used (solid blue curve) and another where all data less than 5 ms were dropped (broken red curve). This small change has rather large consequences. The two curves differ strongly at the left (near 5 ms), which is expected, as we have changed the data there. But we also find large differences at the very right end (near 50 ms), which is unsettling.

Polynomial basis functions are global: they have a nonzero value for almost every x. The net effect is that any change in one of the coefficients in α results in a change in the curve over the entire domain of x. Worse, the higher the degree of the polynomial, the stronger this effect becomes.

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

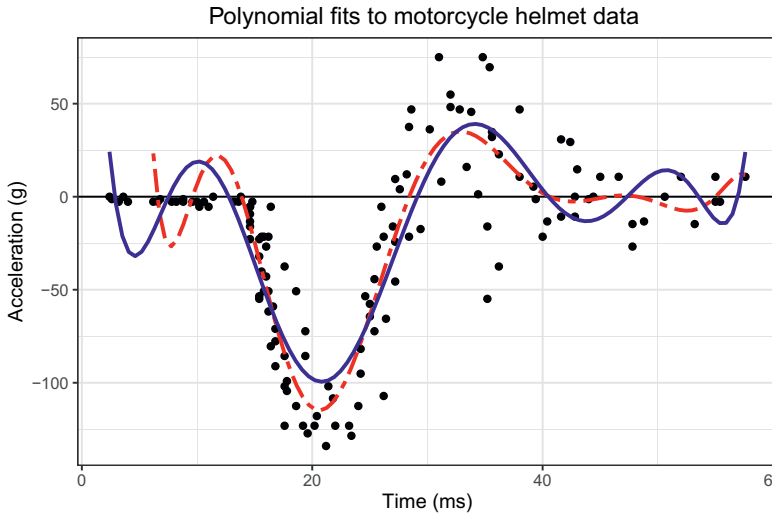

Figure 2.2 The acceleration of a motorcycle helmet in a simulated crash. Two polynomial (degree 9) fits are displayed. Blue line: based on all data; red broken line: after discarding the observations at less than 5 ms. `R` code in `f-motpol1.R`

2.2 B-splines

We first visualize B-splines before using them. The left panel of Figure 2.3 shows seven B-splines, shifted vertically to separate them. The right panel provides a more standard presentation. For either panel, the middle curve shows one complete B-spline, which strongly resembles a normal density. The other curves are shifted copies of this middle curve, but truncated at the left or right boundary.

These are so-called cubic, or degree 3, B-splines. Each B-spline consists of four polynomial segments, each of degree 3, that begin and end at specific values of x called knots. In Figure 2.3, the knots are located at the integer numbers 0 to 4. At the inner knots (1 to 3) two polynomial segments of the same B-spline meet; their values and those of the first and second derivative are equal on both sides of each knot. Together these (degree $+1$) polynomial segments form one B-spline basis function (resembling a normal density).

The knots divide the domain of x into four sections of equal length. The number of B-splines is seven because they have degree 3. In both panels a vertical broken line visualizes the evaluation of the B-splines for one value of x.

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

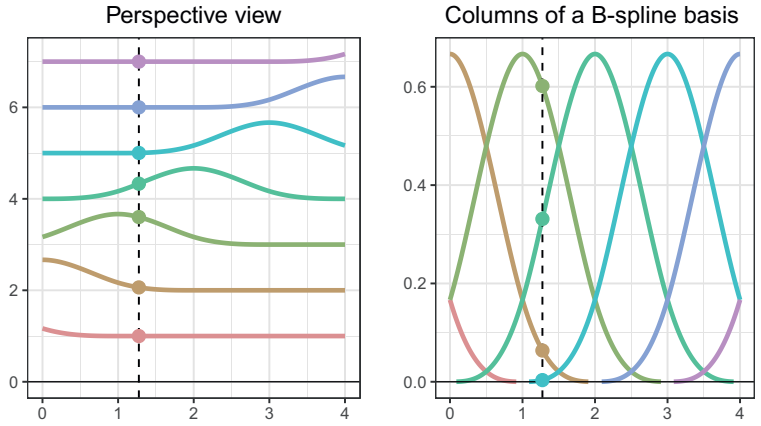

Figure 2.3 B-splines in perspective. In the left panel, the splines are offset vertically, in the right panel they are plotted on top of each other. R code in `f-persp.R`

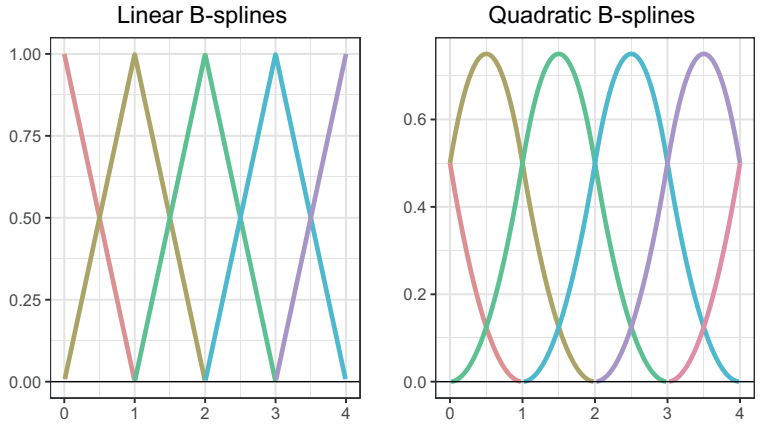

Figure 2.4 Linear (left) and quadratic (right) B-spline bases illustrated. R code in `f-B-lin-quad.R`

Only four of the evaluations have a nonzero value; which four is determined by the value of x. It is easy to check this by imagining a vertical line anywhere in the two panels. The number four is determined only by the degree of the B-splines and does not depend on their number. Said differently, even in a large basis with many B-splines, only four of them are nonzero for any x. Figure 2.4 shows linear and quadratic B-splines for the same choice of knots.

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

With x of length m and n B-splines, we form the basis matrix

$$B = [b_{ij}] = [B_j(x_i)] = \begin{bmatrix} B_1(x_1) & B_2(x_1) & \dots & B_n(x_1) \\ B_1(x_2) & B_2(x_2) & \dots & B_n(x_2) \\ \vdots & \vdots & \vdots & \vdots \\ B_1(x_m) & B_2(x_m) & \dots & B_n(x_m) \end{bmatrix},$$

where $i = 1 : m$ and $j = 1 : n$. Note that the elements of x do not have to be evenly spaced. They can have any value on the chosen domain, their order is immaterial, and repeated values are allowed.

Like with other basis functions, $\mu = B\alpha$ gives us values of a curve, at the positions determined by x. Given a vector y of data to be fitted, linear regression gives us, in principle, an estimate of the coefficients: $\hat{\alpha} = (B'WB)^{-1}B'Wy$. This is only true when $B'WB$ can be inverted, which is only the case when every B-spline has enough support, meaning that there are no columns in B with only zeros. For the moment we assume that this is the case. In Section 2.3, we will introduce penalties to solve the support problem.

Given the properties of the chosen B-splines (domain, number of segments, and degree of the splines), a basis matrix $B^\star$ can be computed for any desired new $x^\star$. Multiplication with the coefficients then gives a curve $\hat{f}(x^\star) = \hat{f} = B^\star\hat{\alpha}$. Generally the observed x does not form a nice grid, but one can choose a detailed $x^\star$ for plotting the fit.

The coefficients $\hat{\alpha}$ form the skeleton of a B-spline fit. A plot of them already gives a good impression of what a detailed curve f would look like, especially when the number of splines is large; we can get a glimpse of this by looking ahead to the different panels found in Figure 2.8. Although one generally speaks of a non-parametric model, in fact the influence of each B-spline coefficient on a curve fit can be seen very clearly. In a parametric model with a power functions basis, this is much more difficult.

To characterize a B-spline basis, we use the number of segments on the chosen domain. The number of B-splines is this number plus the degree of the B-splines. It is also the number of columns of B (or $B^\star$). The number of segments is independent of the degree of the B-splines and lends itself well to the human tendency to prefer numbers like 10, 20, or 50.

Returning to the motorcycle data, Figure 2.5 shows two B-spline fits, with slightly different choices of the domain. The left parts of the curves are quite different from each other, whereas the right parts are essentially identical. This illustrates the local behavior of B-splines, in contrast to polynomial fits (Figure 2.2).

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

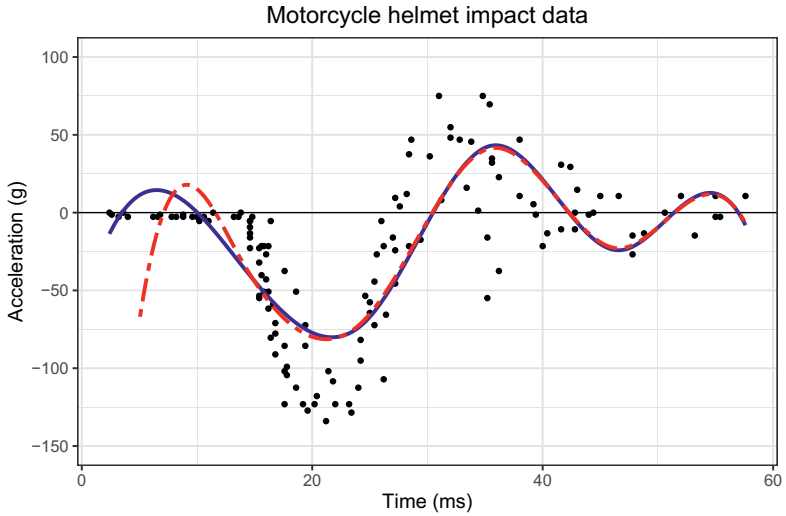

Figure 2.5 The motorcycle data, fit with cubic B-splines (five segments). Blue curve: based on all data; red broken curve: after discarding the observations at less than 5 ms. R code in `f-mot-bsp.R`

B-splines are zero over the largest part of the domain of x. As we have seen, this limited support makes them respond only locally to data, which is an advantage. On the other hand, it makes them vulnerable to sparse or missing data. Figure 2.6 presents two fits to the motorcycle data, one using 10 and the other using 20 segments. In the latter case, the rightmost B-splines are poorly supported, leading to $B'B$ being almost singular and the presence of so-called variance inflation. These consequences are illustrated by the downward swing near the end of the fitted curve.

Figure 2.7 shows B-spline fits to simulated data. The curve is smooth with a small basis and more wiggly with a larger one. One may get the impression that more splines in B automatically lead to a more detailed and wiggly curve $B\alpha$. While this is often true, it is not necessary. In fact, the smoothness of $\mu = B\alpha$ depends on α, not only on B. Consider Figure 2.8, which shows a variety of curves μ using the *same* B-spline basis, with different α. The values of the coefficients are plotted in the graphs to show that the smoother curves correspond to a less erratic α. As noted above, a plot of α alone already gives a strong impression of what the curve will be.

Figure 2.8 also displays the main idea of P-splines: use a rich B-spline basis with relatively many columns, say 20 or even 50, and control the smoothness of the coefficients α. To achieve such control, we need a measure for the

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

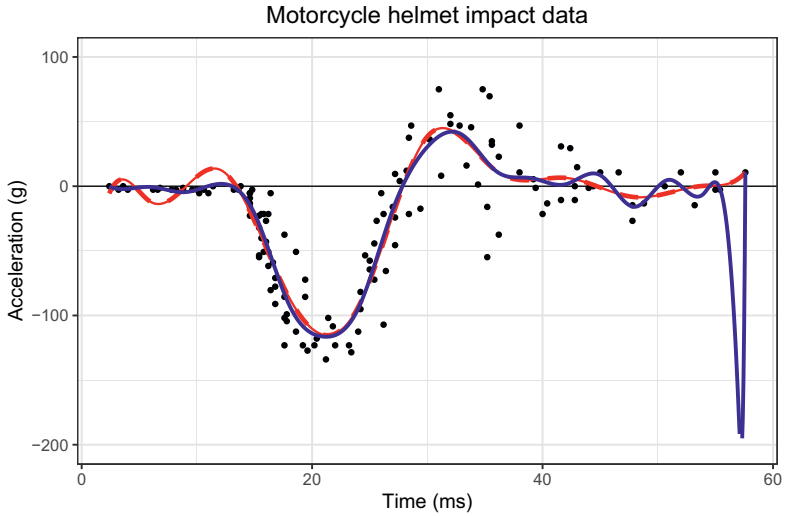

Figure 2.6 The motorcycle data fit with cubic B-splines using 10 (red broken curve) and 20 (blue solid curve) segments. R code in `f-mot-bsize.R`

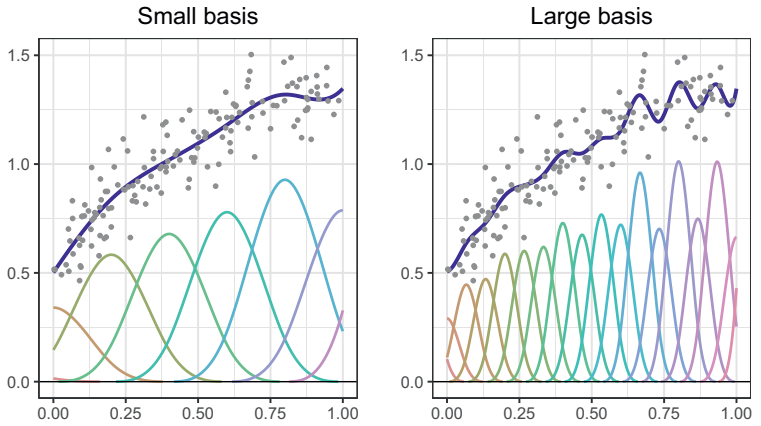

Figure 2.7 Two cubic B-spline fits to the same simulated data, with a small basis (left) and with a larger one (right). R code in `f-bsize.R`

roughness of α, so that we can penalize for it in a properly chosen objective function. Excellent candidates for measuring roughness are differences in adjacent elements of α, and we will consider a variety of differencing orders, e.g., first, second, or even higher.

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

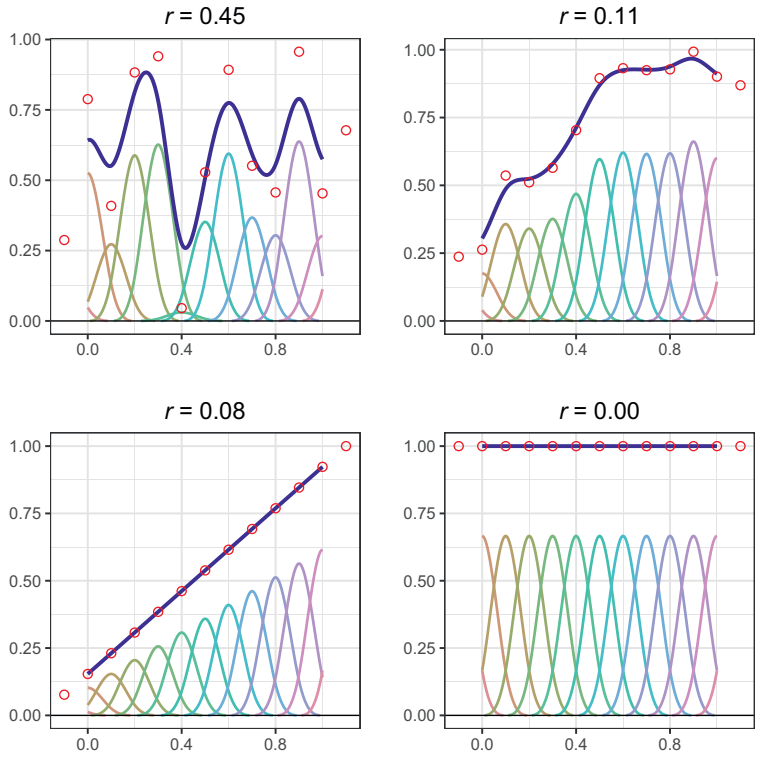

Figure 2.8 An illustration of how roughness of the mean μ can vary dramatically while using exactly the same basis B. The roughness of the curve only depends on the roughness of the coefficients α, measured by r. The red circles show the values of the individual coefficients associated with their corresponding B-splines. R code in `f-brough2.R`

2.3 Penalized Least Squares

In this section, we introduce penalties that are based on differences of neighboring elements of α. First-order differences are defined as $\Delta\alpha_j = \alpha_j - \alpha_{j-1}$. Here Δ is an operator, not a number. Second-order differences are obtained by applying the operator twice: $\Delta^2\alpha_j = \Delta(\alpha_j - \alpha_{j-1}) = \Delta\alpha_j - \Delta\alpha_{j-1}$, and higher orders follow by induction. With $\Delta^d\alpha$, we indicate dth ordered differences of all elements of α. Note that $\Delta^d\alpha_j$ does not exist for $j < 1 + d$. If α has n elements, $\Delta^d\alpha$ has $n - d$ elements.

Both for theoretical and numerical work, it is convenient to have matrix operations for differences, so that $\Delta\alpha$ can be written as $D_1\alpha$ and $\Delta^2\alpha$ as $D_2\alpha$,

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

and so on for higher orders. The required matrices show patterns as in the low dimensional examples below:

$$D_1 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}. \quad (2.4)$$

In R, these matrices are obtained simply by `D=diff(diag(n),diff=d)`, for $d < n$.

To measure the roughness of α, we use the sum of the squares of differences of order d:

$$R = R(\alpha) = \sum_{j=1+d}^{n} (\Delta^d \alpha_j)^2 = \alpha' D_d' D_d \alpha = ||D_d \alpha||^2. \quad (2.5)$$

A derived measure is

$$r = \sqrt{R/(n-d)}.$$

Figure 2.8 gives an impression of r for four different choices of the α vector, each with decreasing roughness. Note that in the limit all elements of α are equal and $r = 0$. In contexts where it is not explicitly needed, we will drop the subscript in D_d, and simply use the notation D. Typically we use a second-order penalty ($d = 2$) in this book.

Now that we have a measure of roughness associated with the B-spline coefficient vector, we can use it as a penalty to minimize the following objective function:

$$Q = (y - B\alpha)' W (y - B\alpha) + \lambda ||D\alpha||^2. \quad (2.6)$$

The second term is the penalty measuring the roughness of α. Its influence is determined by λ, a positive number, which is sometimes referred to as a tuning parameter. Standard B-spline smoothing results when $\lambda = 0$. Initially we will assume that λ is chosen by the user. In Chapter 3, we will present procedures for finding a reasonable data-driven or model-based choices for λ.

Equation (2.6) now technically presents the essence of P-splines: regression on a rich B-spline basis, combined with a discrete roughness penalty on the coefficients. As mentioned, we recommend using many splines to avoid any discussion about the size of the basis. The default choice can be 50, unless prior knowledge of flexibility indicates more. There is no way in which this simple recipe can go wrong; even 1,000 B-splines will work well with only 10 observations. In most of our examples, we will use 20 to 50 segments, unless indicated otherwise. Often 50 is (far) more than needed, but we suggest this number to emphasize that it is impossible to have too many B-splines.

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

Setting the derivative of Q, with respect to α, equal to zero produces the penalized least squares equations:

$$(B'WB + \lambda D'D)\alpha = B'Wy, \quad \text{or} \quad \hat{\alpha} = (B'WB + \lambda D'D)^{-1}B'Wy.$$

The solution depends on the value of λ, giving easy control over smoothness. Note that $B'WB$, $D'D$, and $B'Wy$ only have to be computed once. The entire smoothing problem is now driven by the value of λ. Remarkably, $B'WB+\lambda D'D$ is of full rank, even though this is certainly not the case for $D'D$, and potentially not for $B'WB$. This is the reason that it can do no harm to have many basis functions in B, allowing very flexible fits to our data, even for the case of $n \gg m$. Eilers et al. (2015) provide an example, smoothing 10 data points with 40 segments on the basis. The appendix of that paper also contains a proof.

Figure 2.9 displays the penalty in action for a simple scatter plot. As λ increases, the objective Q places more and more weight on the roughness

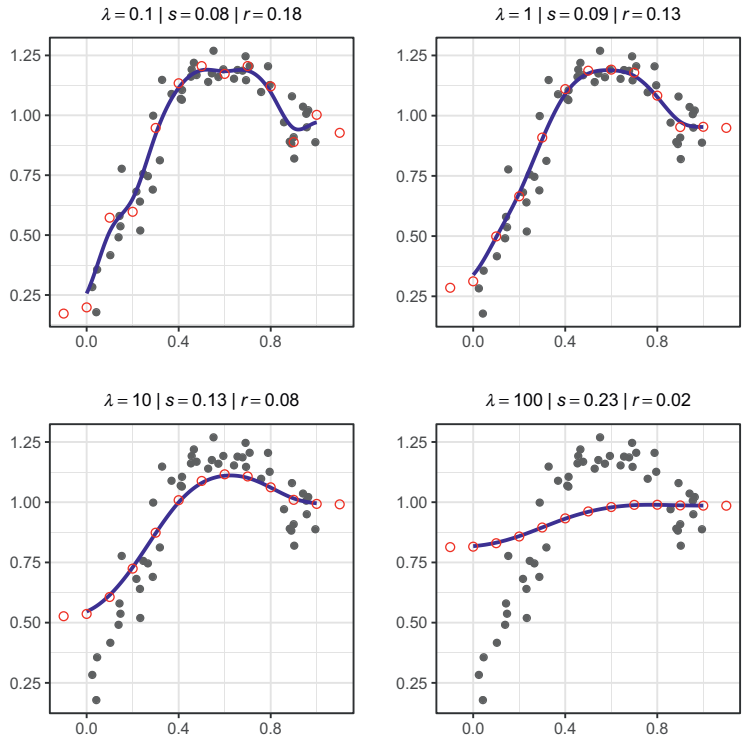

Figure 2.9 The first-order penalty in action for various values of λ. Also shown are the standard deviations of the residuals (s) and the roughness measure (r). Cubic B-splines, 20 segments. R code in `f-dlpen.R`

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

measure R, and the minimization of Q produces fitted curves that become increasingly more smooth. This feature is also observed through the estimated coefficients $\hat{\alpha}$: they fluctuate less as the penalty increases. This is expressed numerically by r. On the other hand s, the standard deviation of the residuals, increases.

The smooth fit tends toward a horizontal line as λ gets larger, and it is easy to see why this is the case. If the differences between neighboring coefficients are small, they will essentially have identical values.

The tendency toward a horizontal line can make the curve fit inflexible, generating a strong bias. Second-order differences are more attractive and generally give a smoother curve without increasing the sum of squares of the residuals; see Figure 2.10. They are nothing more than difference of differences, and the simple line of R code is provided just below (2.4) with $d = 2$.

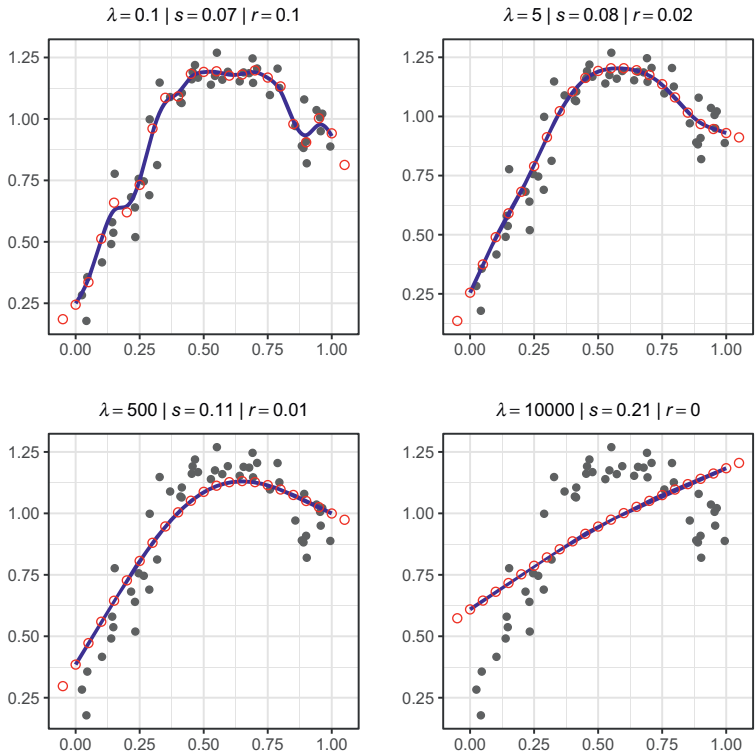

Figure 2.10 The second-order penalty in action for various values of λ. Also shown are the standard deviations of the residuals (s) and the roughness measure (r). Cubic B-splines, 20 segments. R code in `f-d2pen.R`

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

In holds that with differences of order d, the limit of heavy smoothing will be a polynomial of degrees $d-1$. It is the best-fitting (least squares) polynomial of that degree. In Section 2.8, we discuss the limiting behavior of P-splines in more detail.

2.4 Interpolation and Extrapolation

As explained in Section 2.2, to get fitted values, we only need the B-spline basis matrix B and the estimated $\hat{\alpha}$. We can also interpolate to any desired resolution by evaluating the same B-splines, but now on a finer grid of x and multiplying it by $\hat{\alpha}$.

P-splines offer a second type of interpolation, one of the coefficients of B-splines that does not have support. As if by magic, they are filled in automatically. Eilers and Marx (2010) present a detailed analysis that we summarize here. The penalized least squares equations are the key. We have that

$$(B'WB + \lambda D'D)\alpha = B'Wy. \tag{2.7}$$

Assume that several neighboring elements of $B'Wy$ are zero because of zero weights in W. The corresponding rows and columns of $B'WB$ will then also be zero, and we will have that

$$\lambda \breve{D}'\breve{D}\breve{\alpha} = 0, \tag{2.8}$$

where $\breve{\alpha}$ and $\breve{D}$ contain only the affected rows of α and columns of D. Equation 2.8 is a homogeneous linear difference equation of order $2d$. The solution is an exact polynomial in j, the index of the coefficients, of degree $2d-1$. With a first-order penalty, interpolation is linear, and with a second-order penalty it is cubic. This holds for the coefficients. When "fleshing out" the curve with B-splines, some rounding of the shape will occur at the boundaries of interpolated regions.

Extrapolation is just as easy as interpolation. Pseudo-observations with zero weights are added at one or both ends of the data, extending the domain of x. The corresponding y-values can be any arbitrary number, but zero is an obvious choice. P-spline fitting automatically gives extrapolated B-spline coefficients from which an extrapolated curve fit can be calculated.

We do not get linear, but constant, extrapolation when $d=1$. The explanation is found in the upper left and lower right corners of $D'D$. Consider

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

extrapolation at the left boundary with $d = 1$. Then $\breve{D}'\breve{D}$ would be the upper left block of

$$D'D = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots \\ -1 & 2 & -1 & 0 & \dots \\ 0 & -1 & 2 & -1 & \dots \\ 0 & 0 & -1 & 2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

The first row of $\breve{D}'\breve{D}\breve{\alpha} = 0$ forces $\breve{\alpha}_1 = \breve{\alpha}_2$, leaving $\breve{\alpha} \equiv c$ as the only possibility, where c is computed automatically to connect to the observed data smoothly.

With second-order differences, we have that

$$D_2'D_2 = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & 0 & \dots \\ -2 & 5 & -4 & 1 & 0 & 0 & \dots \\ 1 & -4 & 6 & -4 & 1 & 0 & \dots \\ 0 & 1 & -4 & 6 & -4 & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \end{bmatrix}. \tag{2.9}$$

In the first two rows, the values of the elements are such that they "kill" quadratic and cubic terms, like the linear term was annihilated for $d = 1$.

We used zero weights for existing data as example for explaining inter- and extrapolation. The situation is a little different when there are simply no observations for large parts of the domain of x. As a result, some (or many) of the B-splines in the basis matrix have no support and the corresponding columns of B will contain only zeros. They do not contribute to $B'WB$ and $B'Wy$, and the corresponding rows of the latter and rows of columns of the former contain only zeros. We thus get the same results as for the analysis with zero weights.

Figure 2.11 demonstrates interpolation and extrapolation of data having a large gap in the central region of x. The basis matrix has three unsupported B-splines in the center and five at each end.

Automatic interpolation is extremely convenient. We do not have to worry about lack of support when choosing the number of B-splines in a basis. The function `bbase` in our package `JOPS` is happy to compute a basis with (many) unsupported B-splines. Many high-profile packages, like `mgcv` and `gamlss`, use algorithms that simply refuse to do such.

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

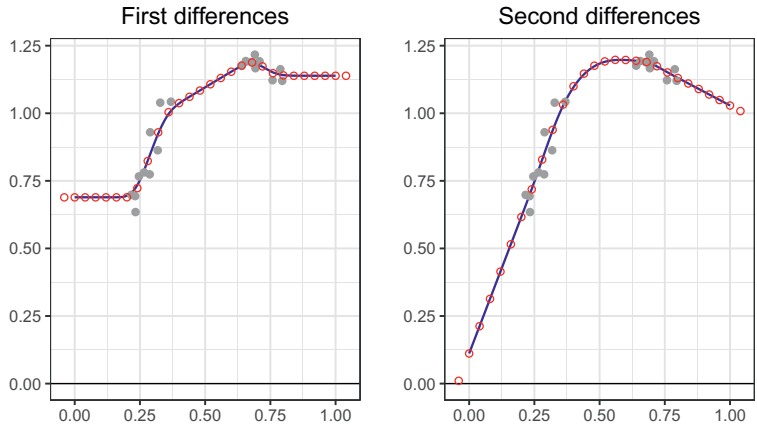

Figure 2.11 P-spline extrapolation and interpolation of a large gap in the x data, with penalty order $d = 1$ (left) and $d = 2$ (right). The gray dots show the data, and the blue circles show the values of the B-spline coefficients. The fitted curve is shown in red. R code in `f-extrapol1.R`

2.5 Derivatives

In some applications, we are not only interested in a fitted curve, but also in its derivative(s). A typical example is the growth speed of children. We use the fact that

$$\frac{d}{dt}\sum_j B_j(t;p)\alpha_j = \sum_j B_j(t;p-1)\Delta\alpha_j/h, \tag{2.10}$$

where p the degree of the B-splines and h the distance between the knots, which is equal to the length of the domain of x divided by the number of segments. When $\hat{\mu} = B\hat{\alpha}$, we get

$$\frac{d}{dt}\hat{\mu} = \tilde{B}(D\hat{\alpha})/h, \tag{2.11}$$

where $\tilde{B}$ contains the B-spline basis of degree $p-1$, and D forms first-order differences. Derivatives of order k can be computed in a similar way:

$$\frac{d^k}{dt^k}\sum_j B_j(t;p)\alpha_j = \sum_j B_j(t;p-k)(\Delta^k\alpha_j)/h^k. \tag{2.12}$$

A condition is that $p > k$, otherwise the B-splines of degree $p-k$ will be zero everywhere.

Figure 2.12 shows an example for a sample of 1,000 boys from the data set `boys7482` in the R package `AGD`. Note that the growth speed does not go to zero

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

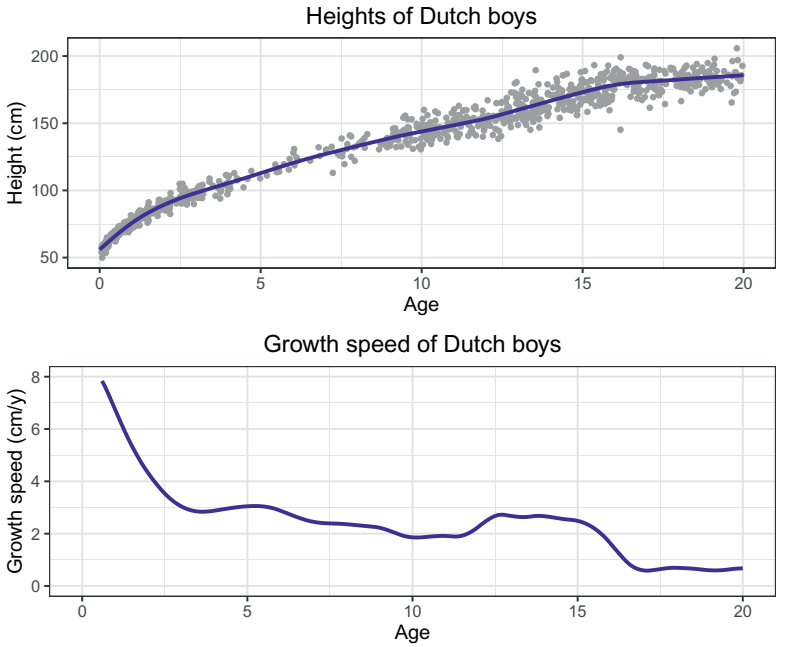

Figure 2.12 Height against age of 1,000 Dutch boys. Top panel: observations and fitted trend (50 cubic segments, $\lambda = 100$). Bottom panel: first derivative of the trend. R code in `f-slope-height.R`

at age 20, as one would expect for human height. In Section 8.7 we show how an extra penalty can force the slope to be zero from a specified age.

2.6 The Effective Dimension

For standard (unpenalized) regression models, we have the estimated mean $\hat{\mu} = B\hat{\alpha} = Hy$, where the least squares solution for α is $\hat{\alpha} = (B'WB)^{-1}B'Wy$. The "hat" matrix is defined as

$$H = B(B'WB)^{-1}B'W, \tag{2.13}$$

such that $\hat{\mu} = Hy$. For a general (full rank) regressor matrix B of dimension $m \times n$, we have the trace(H) = trace$((B'WB)^{-1}B'WB)$ = trace$(I) = n$ because the identity matrix is of dimension n. This result holds due to the invariance of the trace operator under cyclical permutation, i.e., trace(AB) = trace(BA) for general matrices A and B of proper dimension. Thus, for standard multiple regression, the trace(H) yields the exact dimension of the fit. This

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

result is developed further and extended by Hastie and Tibshirani (1990) in a manner to compute the effective dimension of a smooth fit $\hat{\mu} = Hy$, where H is a general smoother matrix and the *effective dimension*, $\text{ED} = \text{trace}(H)$.

A more principled proposal comes from Ye (1998). This work states that the effective model dimension is

$$\text{ED} = \sum_i \partial \hat{y}_i / \partial y_i . \tag{2.14}$$

In the linear case, this gives again $\text{ED} = \sum_i h_{ii} = \text{trace}(H)$. At first, this definition of ED may not look impressive, but in fact it is powerful. The partial derivatives can be decomposed into separate and quantifiable contributions of individual components within larger models. Examples are the mixed models presented in Chapter 3 and the additive models found in Chapter 4.

This definition of the effective dimension also applies to P-splines, now with $H = B(B'WB + \lambda D'D)^{-1}B'W$. Using cyclic permutation, we find

$$\text{ED} = \text{trace}(H) = \text{trace}(G) = \text{trace}(B'WB(B'WB + \lambda D'D)^{-1}). \tag{2.15}$$

Because G is an n by n matrix and generally much smaller than H, the latter definition is computationally more attractive.

There is also a monotone relationship between ED and λ. When λ approaches zero, ED approaches n (assuming support for all B-splines). The limit for increasing λ is $\text{ED} = d$. We see this relationship more completely in Figure 2.13. Note that y does not occur in the definition of ED; it is purely a

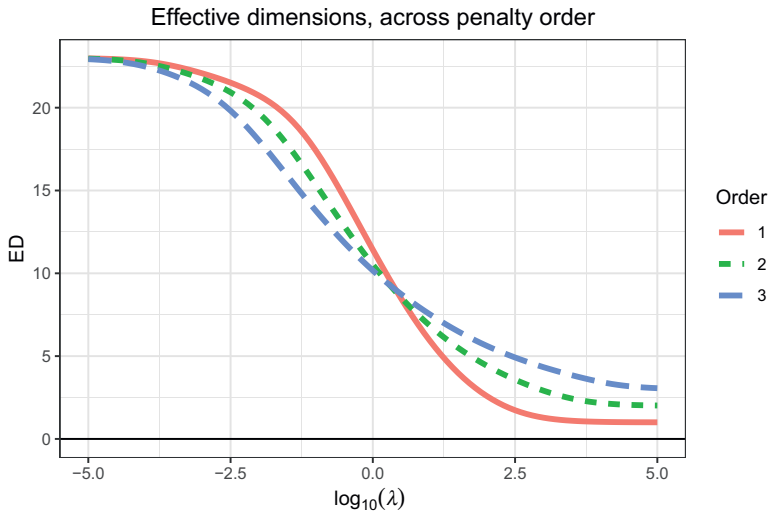

Figure 2.13 ED versus $\log_{10}(\lambda)$, by penalty order using 20 segments on the basis. R code in `f-Peffdim.R`

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

property of the design of the model (basis plus penalty for given λ). We will find in Section 2.8, that for large λ, the fitted curve approaches a polynomial of degree $d - 1$.

2.7 Standard Errors

The covariance matrix of $\hat{\alpha} = (B'WB + \lambda D'D)^{-1}B'Wy$, or that of the estimated coefficients, is

$$\hat{C} = \text{cov}(\hat{\alpha}) = \hat{\sigma}^2(B'WB + \lambda D'D)^{-1}, \tag{2.16}$$

where $\hat{\sigma}^2 = ||y - B\hat{\alpha}||^2/(m - \text{ED})$. This is sometimes called the Bayesian covariance matrix. Its derivation is outlined in Appendix F, where it is also compared to an alternative, the sandwich estimate.

The diagonal of $B\hat{C}B'$ gives the variance of the fitted values, $\hat{\mu} = B\hat{\alpha}$. From its square root, we can construct error bands.

The underlying assumption in these equations is that we fill in the true λ, which is of course unknown. One hopes that "optimal smoothing" (see Chapter 3) will give a good estimate. Figure 2.14 displays an "optimal" P-spline fit (as determined by leave-one-out cross-validation) with twice standard error bands for the motorcycle data.

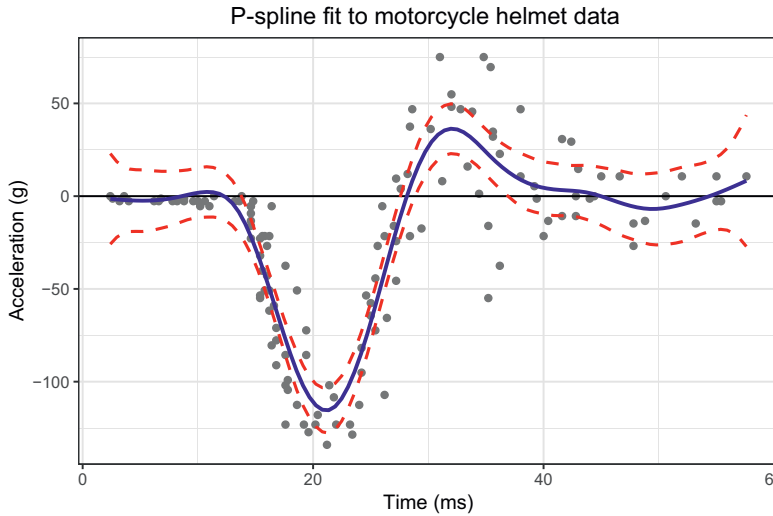

Figure 2.14 P-spline fit to the motorcycle helmet data, with twice standard error bands (20 cubic segments, second-order penalty). R code in `f-se.R`

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

2.8 Heavy Smoothing and Polynomial Limits

In the penalized least squares objective

$$Q = S + \lambda R = (y - B\alpha)'W(y - B\alpha) + \lambda ||D\alpha||^2,$$

the second term becomes dominant for large λ, which essentially forces $D\alpha \approx 0$.

If $\alpha_j = \sum_{k=0}^{d-1} \gamma_j j^k$, i.e., when the coefficients follow a polynomial of degree $d - 1$, then $||D\alpha||^2 = 0$ exactly. This is easy to prove when $\alpha_j = a + bj$. It follows

$$\Delta\alpha_{j+1} = \alpha_{j+1} - \alpha_j = a + b(j + 1) - (a + bj) = b, \tag{2.17}$$

for any j, and $\Delta^2\alpha_{j+1} = \Delta b = 0$. Generalizations to higher-order differences are straightforward.

As λ increases, $\hat{\alpha}$ approaches a polynomial, and so will $B\hat{\alpha}$. The limiting polynomial is not arbitrary: it minimizes the sum of squares in the first term of the objective, hence the fit is the least squares polynomial. This bridge between P-splines and polynomials can be useful: sometimes strong smoothing is indicated by the data, and the P-splines can be replaced by a simple parametric model.

2.9 P-splines as a Parametric Model

A polynomial fit to data is commonly called a parametric model. A handful of coefficients fully determines the curve. Often it is suggested that a parametric model is desirable because one knows "what the parameters stand for." Yet, beyond a cubic polynomial this is dubious: it is almost impossible to relate the value of one coefficient of a higher power of x directly to the shape of the curve.

On the other hand, smoothing methods often are termed non-parametric or semi-parametric models, implying that there are no (or very few) parameters with a clear interpretation. This is certainly true for kernel smoothers and local likelihood methods. Truncated polynomials have clearly defined parameters, but they cannot be precisely interpreted (Eilers and Marx, 2010).

P-splines are different. As illustrated for cubic P-splines in Figure 2.15, the values of the coefficients are close to the value of the fitted curve directly above the peak of the corresponding B-spline. The interpretation of the parameters is easy: they closely predict the fitted curve at the center of the corresponding B-spline.

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

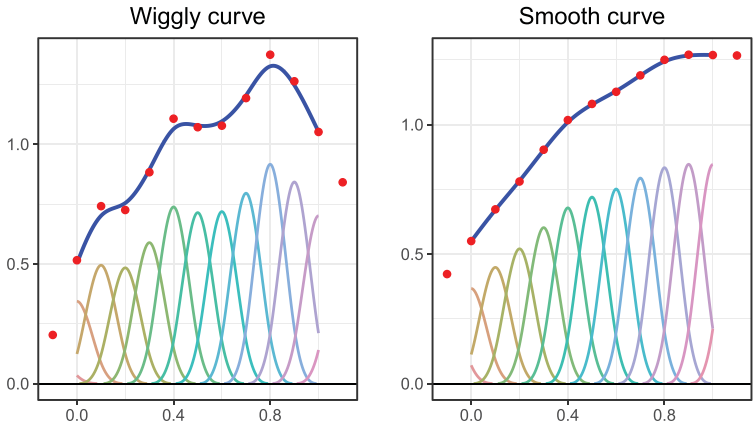

Figure 2.15 The values of the coefficients of the B-splines (red dots) lie close to the fitted curve (blue line). The smoother the curve, the closer the coefficients are to each other. R code in `f-bcoeff.R`

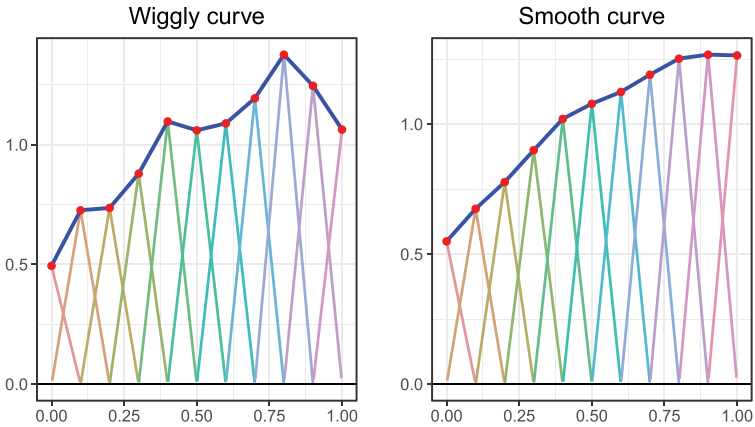

Figure 2.16 The values of the coefficients of linear B-splines (red dots) lie on the fitted curve (blue line). R code in `f-bcoeff-lin.R`

Linear P-splines have an almost perfect interpretation of the coefficients, as Figure 2.16 shows. The fitted curve linearly interpolates the coefficients. A piecewise linear curve with many knots can be acceptable in many applications, especially for visualization. A basis of linear B-splines can be computed in a few line of R code, as shown in Appendix C.1.

In view of the excellent interpretability of explicit coefficients, we should call P-splines a "proper parametric" model.

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

2.10 Whittaker: P-splines without B-splines

We obtain interesting and useful results when we replace B by an identity matrix, leading to $\hat{\mu} = (I + D'D)^{-1}y$. This is only meaningful if the locations of the observations are evenly spaced, and if we are interested in fitted values only on those locations. The number of coefficients will be equal to the number of observations (missing data can be handled with zero weights).

What results is the Whittaker smoother, originally designed for smoothing ("graduation") of life tables (Whittaker, 1923). The original algorithm used third-order differences. This smoother gradually became less used as the popularity of the smoothing spline was rising. P-splines can be considered as "Whittaker on steroids," with a skeleton defined by a discrete penalty, on which B-splines lay the muscles.

An advantage of the Whittaker smoother, when the application allows its use, is the extreme simplicity of the basis. There is neither the need to choose the number of knots nor the degree of the splines. All the pleasant properties of the penalty remain. Interpolation and extrapolation are automatic, and strong smoothing leads to polynomials, among other properties.

The equations for the Whittaker smoother are extremely sparse. Using sparse matrix algorithms (Eilers, 2003), extremely long data series can be smoothed in a fraction of a second. Unfortunately, calculation of the full hat matrix, $H = (I + \lambda D'D)^{-1}$, is not efficient because the inverse is large and not sparse. For computing its diagonal, Frasso and Eilers (2015) provides efficient `R` code (for $d = 2$), based on an algorithm by Hutchinson and de Hoog (1986). Its computation time and memory use only grow linearly with the length of the data series. It can also be used to compute standard errors, as the covariance matrix of $\hat{\mu}$ is $\hat{\sigma}^2 H$, so the square root of its diagonal gives the standard errors of the fitted $\hat{\mu}$.

The B-spline basis is reduced to a minimum and replaced by the identity matrix. The choice of number and degree of the splines do not play any role. This is what makes the Whittaker smoother a good vehicle for studying discrete penalties.

2.11 Equivalent Kernels

Hastie and Tibshirani (1990) show that linear smoothers can be better understood and more directly comparable if they are expressed as equivalent kernels. An equivalent kernel shows in detail how individual values in the input vector contribute to the smooth output. Consider the P-spline "hat" matrix

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

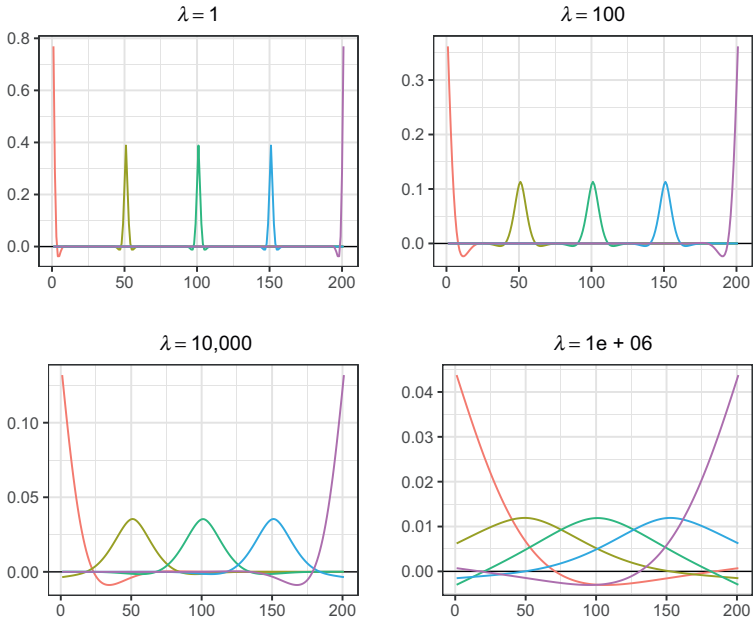

Figure 2.17 Illustration of equivalent kernels. The differently colored curves show the values in rows 1, 51, 101, 151, and 201 of the 201 by 201 hat matrix of the Whittaker smoother with penalty order 2. The titles of the panels show the value of λ. R code in `f-eff-kernels.R`

$H = B(B'B + \lambda D'D)^{-1}B'$ and how it transforms y into $\hat{y} = Hy$. Row i of H tells us how $\hat{y}_i = \sum_j h_{ij} y_j$ is formed as a weighted sum of all observations. The elements in row i form the equivalent kernel. Figure 2.17 shows, for different values of λ, the equivalent kernels in selected rows (1, 51, 101, 151, and 201) of a 201 by 201 hat matrix. It is based on the Whittaker smoother.

When λ is small, most of the elements in a row of H are close to zero, indicating that any element of $\hat{y}$ is only influenced by a few observations close to it. When λ is large, almost all observations contribute to each $\hat{y}_i$. It easy to see that $\sum_j h_{ij} = 1$ for all i, because if e is defined as an m vector of all ones, then $e = He$. Hence $\hat{y}_i$ is a proper weighted mean of y, with the weights in the ith row of H. Borrowing from systems theory, we can interpret $\hat{y} = Hy$ as the description of a linear system, with y as input, and H as describing how it is transformed to the output $\hat{y}$.

Note that at the boundaries the equivalent kernels get a strongly asymmetrical shape, but they do not cross the boundaries. This is simply impossible. The vanilla kernel smoother is different. All equivalent kernels have the same shape,

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

implying that those near the boundaries spread out beyond them. See also the discussion of boundary effects in density estimation in Section 3.3.

From its definition follows that H is symmetric. Consider the special case that y consists of all zeros, except for one $y_j = 1$, which is often called an impulse signal. In such a case, Hy is equal to column j of H. Previously, we looked at rows of H that told us how all observations contribute to one element of $\hat{y}$; we now look at how one element of the input vector is distributed over the whole output vector.

In Section 8.5, we will present variations of the penalty and show their effects on equivalent kernels for some of them.

2.12 Smoothing of a Non-normal Response

In many applications, the response will not be normal. Common examples include Poisson distributed counts or binomial responses. P-splines can be directly transplanted into the generalized linear model (GLM) framework. For an introduction to GLM, we recommend Dobson and Barnett (2018) and Fahrmeir and Tutz (2001).

The GLM introduces three parts: the random component, the linear predictor, and the link function. The random component specifies the probability distribution of y. It can be any member of the exponential family, with mean $\mu = E(y)$, but we will restrict ourselves to the Poisson and binomial. The second component is the linear predictor, which we model with B-splines, $\eta = B\alpha$. The last component is the link function g: $\eta = g(\mu)$. Common (canonical) link functions include the logarithmic link (for Poisson) and the logit link (for binomial), which will be our choices throughout this book.

Maximum likelihood estimation is standard for the GLM. Because we want to introduce a penalty, we find it convenient to switch from maximizing the log-likelihood to minimizing the deviance, which is essentially (apart from a constant) minus two times the log-likelihood. We use Poisson smoothing to explain this idea.

2.12.1 Poisson Smoothing

Assuming independent observations, the Poisson likelihood is

$$L = \prod_{i=1}^{m} \frac{e^{-\mu_i} \mu_i^{y_i}}{y_i!}.$$

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

Neglecting the (constant) contribution from $y_i!$, the log-likelihood is

$$\ell(\mu; y) = \log(L) = \sum_{i=1}^{m} (y_i \log(\mu_i) - \mu_i). \tag{2.18}$$

For convenience, we choose to work with the deviance, which is $\text{dev}(\mu; y) = -2\ell(\mu; y) + 2\ell(y; y)$. Using deviance, a penalty can be attached in a similar way as it was to the sum of squares of residuals in (2.6). We now have an objective function in the form

$$\begin{aligned} Q &= \text{dev}(\mu; y) + \lambda ||D\alpha||^2 \\ &= 2 \sum_{i=1}^{m} (y_i \log(y_i/\mu_i) - (y_i - \mu_i)) + \lambda ||D\alpha||^2, \end{aligned} \tag{2.19}$$

which for Poisson responses and the log link becomes a function of α through $\mu = \exp(\eta) = \exp(B\alpha)$. The goal is to minimize Q with respect to α, which is the solution to $\partial Q/\partial \alpha = 0$. Re-expressing with the chain rule, we find

$$0 = \frac{\partial Q}{\partial \alpha} = \frac{\partial Q}{\partial \mu} \frac{\partial \mu}{\partial \eta} \frac{\partial \eta}{\partial \alpha}.$$

Note that the third term simplifies to $B = \partial \eta/\partial \alpha$, and for Poisson (log link) the second term is $\mu = \partial \mu/\partial \eta$. The partial derivatives then yield the penalized likelihood equations

$$B'(y - \mu) = \lambda D' D\alpha, \tag{2.20}$$

which remain nonlinear in α. The Newton–Raphson method applies a first-order Taylor approximation to the derivative on the left-hand side of (2.20) (about $\tilde{\alpha}$),

$$B'(y - \mu) \approx B'(y - \tilde{\mu}) - B'\tilde{W}B(\alpha - \tilde{\alpha}),$$

with $\tilde{\eta} = B\tilde{\alpha}$, $\tilde{\mu} = \exp(\tilde{\eta})$, and $\tilde{W} = \text{diag}(\tilde{\mu})$, effectively providing a second-order approximation at a first-order price. The matrix $H = -B'WB$ is the so-called Hessian matrix, which is $H = -B'\partial \mu/\partial \alpha'$. Here the Newton–Raphson is equivalent to the Method of Scoring because $E(H) = H$. This finally gives the linear system of equations

$$(B'\tilde{W}_t B + \lambda D' D)\tilde{\alpha}_{t+1} = B'\tilde{W}_t(\tilde{\eta}_t + \tilde{W}_t^{-1}(y - \tilde{\mu}_t)). \tag{2.21}$$

This system has to be solved iteratively. The subscript t indicates the iteration number. Starting values are only needed for the linear predictor η, which set

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

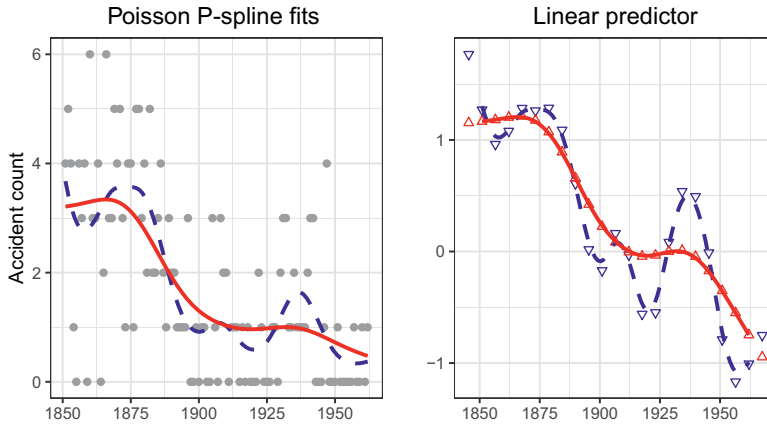

Figure 2.18 Yearly counts of British coal mining disasters, smoothed with a Poisson model. P-spline fits with a cubic B-spline basis with 20 segments and a second-order penalty, for both $\lambda = 1$ and $\lambda = 100$ (left). The corresponding linear predictors and the B-spline coefficients are also shown (right). R code in `f-coal-smooth.R`

up the GLM weights and working vector to initialize the iterations. We set $\tilde{\eta}_0 = \log(y + 1)$, giving

$$\tilde{\alpha}_{t+1} = (B'\tilde{W}_t B + \lambda D'D)^{-1} B'\tilde{W}_t \tilde{z}_t, \tag{2.22}$$

with $\tilde{z}_t = \tilde{\eta}_t + \tilde{W}_t^{-1}(y - \tilde{\mu}_t)$ as a working dependent vector. We find that (2.22) reflects an iterative re-weighted (penalized) least squares solution. Usually convergence is quick, needing only a handful of iterations. The working vector is a foundational to effective dimension, standard error bands, among other things, as outlined in Section 2.12.3 ahead.

The stage is now set to fit a smooth curve through a series of counts. We use data on the annual number of major accidents in British coal mines, from 1851 to 1962. They are based on the data set `coal` in R. The observations, the fitted values, and the linear predictors are plotted in Figure 2.18, for both light and heavy penalization. The coefficients of the B-splines are plotted too, emphasizing again how they form the skeleton of the smooth linear predictor.

The generalized linear P-spline smoother is ideal for histograms, which are also a series of counts. See Section 3.3 for details, including the choice of domain, the influence of the bin width, and the optimal choice of λ.

P-splines have a property that we call the conservation of moments, which is of great value for density estimation. Let c be the vector of counts in the bins

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

of a histogram of length m, and let $\hat{\mu}$ be the smooth fit. Given the penalty of order d, it holds for all $\lambda \geq 0$:

- For $d = 1$, $\sum_{i=1}^{m} y_i = \sum_{i=1}^{m} \hat{\mu}_i$ (proper density);
- For $d = 2$, the above holds and also $\sum_{i=1}^{m} x_i c_i = \sum_{i=1}^{m} x_i \hat{\mu}_i$ (same mean);
- For $d = 3$, the above hold and also $\sum_{i=1}^{m} x_i^2 c_i = \sum_{i=1}^{m} x_i^2 \hat{\mu}_i$ (same variance).

The conservation property can be extended to higher moments using higher-order penalties. Notice that $d = 3$ is especially useful because both the variance and mean of the smoothed histogram are the same as those of the raw histogram, even with strong smoothing. Most other algorithms would increase the variance. Moreover, the polynomial limits for large λ have a clear interpretation: when $d = 2$, the linear predictor becomes linear in x, and we get an exponential distribution. For $d = 3$, the smooth approaches the normal distribution.

Even with narrow bins, the counts in histograms generally do not form a long data series. The length may be around 100 or so. This is small enough to use the identity matrix as the P-spline basis, as described for the Whittaker smoother in Section 2.10.

In some hazard modeling applications, the goal of smoothing of counts is not to get a smooth estimate of μ itself, but rather of a smooth hazard (or intensity), say h. Then $E(y_i) = \mu_i = h_i u_i$, where u is the exposure, usually a population size. The smoothing of the hazard can be achieved by taking $h = \exp(B\alpha)$ and $U = \text{diag}(u)$, which leads to $\mu = U \exp(B\alpha)$. The classical approach is to use $\mu = \exp(B\alpha + \log u)$, where $\log(u)$ is called the offset. This clearly runs into problems when some elements of u are zero. It can occur that both y and u are zero in some places. A typical example is extrapolation. Multiplication by U avoids logarithms of zeros.

2.12.2 Binomial Smoothing

For binomial responses, y denotes the number of *successes* among t independent trials ($0 \leq y \leq t$). We estimate a smooth curve for the binomial parameter π, which represents the probability of observing a success for any given trial. It follows that $\mu = t\pi$. The same statements hold true for Bernoulli responses, but now with the restrictions $y = 0$ or 1 and $t = 1$, and $P(y = 1) = \pi = \mu$. In the GLM framework, the response distribution is now set to the binomial, and we choose the (canonical) link function to be the logit. The model is

$$g(\mu) = \log\left(\frac{\pi}{1-\pi}\right) = \eta = B\alpha, \tag{2.23}$$

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

where π is the binomial "success" probability. Equivalently,

$$\pi = \frac{1}{1 + \exp(-B\alpha)},$$

and for m independent binomials, each with t_i trials ($i = 1 : m$)

$$Q = 2\sum_{i=1}^{m}\left(y_i \log\left[\frac{y_i}{\mu_i}\right] + (t_i - y_i)\ \log\left[\frac{t_i - y_i}{t_i - \mu_i}\right]\right) + \lambda||D\alpha||^2. \quad (2.24)$$

Neglecting the parts of Q that do not depend on μ, the following simplification results

$$Q = -2\sum_{i=1}^{m}[y_i \log \mu_i + (t_i - y_i)\log(t_i - \mu_i)] + \lambda||D\alpha||^2, \quad (2.25)$$

which further simplifies in the (ungrouped) Bernoulli setting. Using the same linearization technique that was presented for Poisson smoothing, we arrive at equations similar to (2.21), but now with $W = \text{diag}(\mu(1 - \pi))$. A convenient choice for starting values is $\tilde{\pi}_0 = (y_i + 1)/(t_i + 2)$ or $\tilde{\eta}_0 = \log(\tilde{\pi}_0/(1 - \tilde{\pi}_0))$.

We revisit the kyphosis case study presented in Hastie and Tibshirani (1990) (data set `kyphosis` in `R`). The binary response is presence (1) or absence (0) of postoperative spinal deformity in children. There are 81 observations (17 present, 64 absent). We model the probability of occurrence as a function of age. The data and the fitted curves for the probabilities are plotted in Figure 2.19 (left panel) for two levels of smoothing ($\lambda = 1$ and $\lambda = 100$). The corresponding linear predictors are also displayed in the right panel.

2.12.3 GLM Effective Dimension and Standard Errors

Upon convergence, it is useful to view (2.21) as weighted linear smoothing of the working dependent variable $\hat{z} = \hat{\eta} + \hat{W}^{-1}(y - \hat{\mu})$, with weights $\hat{W}$. We can then interpret

$$H = B(B'\hat{W}B + \lambda D'D)^{-1}B'\hat{W} \quad (2.26)$$

as the "effective" hat matrix and

$$\text{ED} = \text{trace}(H) = \text{trace}((B'\hat{W}B + \lambda D'D)^{-1}B'\hat{W}B)$$

as the effective dimension. For the covariance matrix of $\hat{\alpha}$, we find the Bayesian form to yield

$$\text{cov}(\hat{\alpha}) = \phi(B'\hat{W}B + \lambda D'D)^{-1}, \quad (2.27)$$

where ϕ denotes a scale parameter. With normal responses (and the identity link function), a scale parameter, $\phi = \sigma^2$ was presented. For the binomial and

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

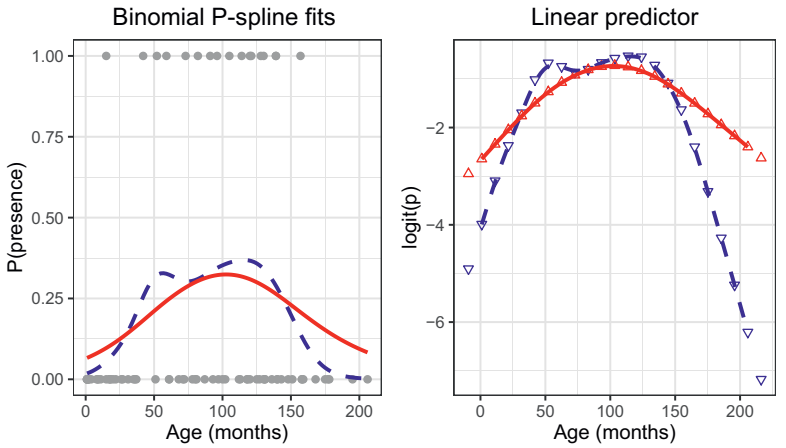

Figure 2.19 Kyphosis binomial response fit with basis using 20 segments (left). The dashed (solid) line indicates light (heavy) penalization using $\lambda = 1$ and 100, respectively. The corresponding linear predictor, and the P-spline coefficients are also shown (right). R code in `f-kypho-smooth.R`

Poisson forms – that are generally associated with the exponential family – we find $\phi = 1$ (see table 2.1 of Fahrmeir and Tutz [2001]). A common problem with Poisson modeling is the presence of overdispersion or $\phi > 1$. In such cases, we can estimate the scale parameter with $\hat{\phi} = \text{Dev}(\hat{\mu}; y)/(m - \text{ED})$. Other estimators of ϕ replace deviance with the Pearson chi-square statistic.

The covariance of $\hat{\eta}$ follows immediately as $\Gamma = B\text{cov}(\hat{\alpha})B'$, and approximate standard errors for the linear predictor can be constructed using the square root of the diagonal elements of Γ. Thus, twice lower and upper bounds for η follow from $\hat{\eta} \pm 2se(\hat{\eta})$. Representing these limits as (L, U), the corresponding limits for $\hat{\mu}$ follow as $(h(L), h(U))$, where $h(\cdot)$ denotes the inverse link function.

In addition to the inherent GLM weights, prior weights can also be useful in generalized linear smoothing. Assuming that such prior weights are now provided in the vector v and $V = \text{diag}(v)$, then $V\tilde{W}$ should replace $\tilde{W}$ everywhere in the fitting algorithm, and $V\hat{W}$ should replace $\hat{W}$ in the effective hat matrix and derived results.

On the scale of the linear predictor, the features of automatic interpolation and extrapolation, as well as polynomial limits for large λ, follow for the GLM as they did for normal responses on the linear predictor. For example, when smoothing binomial data with the logit link function and a second-order penalty, the limiting result with heavy smoothing is the same as linear logistic regression.

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

2.13 Notes and Details

This chapter lays the foundation for the rest of the book. P-splines combine a rich and evenly spaced B-spline basis with a simple difference penalty on the coefficients. Smoothing becomes an extension of (generalized) linear regression, with a single tuning parameter, avoiding tinkering with the size and number of B-splines. Interpolation and extrapolation are essentially automatic.

The difference penalty of P-splines is a simplification of the work of O'Sullivan (1986). He derived a matrix, very similar to our discrete difference matrix D, based on the integrated square of the second derivative of the fitted curve. Whereas pure differences are trivial to construct, O'Sullivan's approach is complicated for higher-order derivatives. Recently published algorithms simplify the computations (Wand and Ormerod, 2008; Wood, 2017). As long as knots are evenly spaced, we do not see any advantages in these works. Wand and Ormerod (2008) claim better performance of what they termed O-splines, but Eilers et al. (2015) show that a wrong B-spline basis had been used for their comparisons.

We emphasize that the discrete difference penalty is not meant at all to be an approximation to a continuous penalty. It is simple, and it is powerful, and it is all we need. It also puts no demands on the degree of the B-spline. To base a penalty on, say, fourth derivatives of the fitted curve, the B-splines should have at least degree 4, or else the derivative disappears. P-splines do not have this limitation. In the next chapter we will see for histogram smoothing that splines of zero order combine perfectly with a third-order difference penalty.

The penalty can be interpreted as the condition that the B-spline parameters α closely obey a linear difference equation. The coefficients of the equation form rows of the Pascal triangle with alternating signs. Many specialized variations are possible; some of them are discussed in Chapter 8.

A few years after we published our paper on P-splines, Ruppert and Carroll (2000) proposed to use a mixed model with truncated power functions (TPF) for smoothing, an approach that is extended by Ruppert et al. (2003). Quantiles of x are used for the knots. This approach underestimates the power of a (difference) penalty. Indeed, if there is no penalty, quantiles as knots guarantee support for all basis functions, which is a good thing. A penalty is a more elegant solution. In Eilers and Marx (2010), we show that evenly spaced knots are to be preferred for TPF.

In many publications we noticed the temptation to optimize the number of splines. Based on the TPF model with quantile-based knots, Ruppert (2002) and Ruppert et al. (2003) present formulas that boil down to one spline per four unique x, with a certain maximum. This advice has been cited many times for

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press

guidance on penalized B-splines. We propose to simply forget optimizing the number of splines and just take a large number of them.

Asymptotic results on P-splines have been obtained by Hall and Opsomer (2005), Kauermann et al. (2009), Li and Ruppert (2008), and Claeskens et al. (2009). They all consider the limited situation in which more data become available on the same domain. A more realistic setting is smoothing of (seasonal) time series, that grow in length as more observations are collected. It is clear that the number of P-splines must then grow too.

The R function `lsfit()` finds the solution of $B'WB\hat{\alpha} = B'Wy$ for given B, y and $W = \text{diag(w)}$. It can also be used for penalized regression through data augmentation, i.e., $\hat{\alpha} = (B'_+W_+B_+)^{-1}B'_+W_+y_+$, where

$$B_+ = \begin{bmatrix} B \\ \sqrt{\lambda}D \end{bmatrix} \quad y_+ = \begin{bmatrix} y \\ 0 \end{bmatrix} \quad \text{and} \quad w_+ = \begin{bmatrix} w \\ 1 \end{bmatrix}.$$

In this way, standard and widely available fitting algorithms that only accept B, y, and w can be tricked into solving penalized regression, including P-splines. If there are multiple penalties or additive structures, they too can be handled by additional augmentations. In this book, we have used `lsfit()` with data augmentation in some of our programs.

https://doi.org/10.1017/9781108610247.003 Published online by Cambridge University Press