(c) Show that

$$\hat{f}(x) = h(x)^T\hat{\boldsymbol{\beta}} = \sum_{i=1}^{N} K(x, x_i)\hat{\boldsymbol{\alpha}}_i \tag{5.76}$$

and $\hat{\boldsymbol{\alpha}} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}$.

(d) How would you modify your solution if $M < N$?

Ex. 5.17 Show how to convert the discrete eigen-decomposition of $\mathbf{K}$ in Section 5.8.2 to estimates of the eigenfunctions of K.

Ex. 5.18 The wavelet function $\psi(x)$ of the symmlet-p wavelet basis has vanishing moments up to order p. Show that this implies that polynomials of order p are represented exactly in V_0, defined on page 176.

Ex. 5.19 Show that the Haar wavelet transform of a signal of length $N = 2^J$ can be computed in $O(N)$ computations.

Appendix: Computations for Splines

In this Appendix, we describe the B-spline basis for representing polynomial splines. We also discuss their use in the computations of smoothing splines.

B-splines

Before we can get started, we need to augment the knot sequence defined in Section 5.2. Let $\xi_0 < \xi_1$ and $\xi_K < \xi_{K+1}$ be two *boundary* knots, which typically define the domain over which we wish to evaluate our spline. We now define the augmented knot sequence τ such that

- $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_M \leq \xi_0$;
- $\tau_{j+M} = \xi_j,\ j = 1, \cdots, K$;
- $\xi_{K+1} \leq \tau_{K+M+1} \leq \tau_{K+M+2} \leq \cdots \leq \tau_{K+2M}$.

The actual values of these additional knots beyond the boundary are arbitrary, and it is customary to make them all the same and equal to ξ_0 and ξ_{K+1}, respectively.

Denote by $B_{i,m}(x)$ the ith B-spline basis function of order m for the knot-sequence τ, $m \leq M$. They are defined recursively in terms of divided differences as follows:

$$B_{i,1}(x) \quad = \quad \begin{cases} 1 & \text{if } \tau_i \leq x < \tau_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{5.77}$$

for $i = 1, \ldots, K + 2M - 1$. These are also known as Haar basis functions.

$$B_{i,m}(x) \quad = \quad \frac{x - \tau_i}{\tau_{i+m-1} - \tau_i} B_{i,m-1}(x) + \frac{\tau_{i+m} - x}{\tau_{i+m} - \tau_{i+1}} B_{i+1,m-1}(x)$$

for $i = 1, \ldots, K + 2M - m$.

$$\tag{5.78}$$

Thus with $M = 4$, $B_{i,4},\ i = 1, \cdots, K + 4$ are the $K + 4$ cubic B-spline basis functions for the knot sequence ξ. This recursion can be continued and will generate the B-spline basis for any order spline. Figure 5.20 shows the sequence of B-splines up to order four with knots at the points $0.0, 0.1, \ldots, 1.0$. Since we have created some duplicate knots, some care has to be taken to avoid division by zero. If we adopt the convention that $B_{i,1} = 0$ if $\tau_i = \tau_{i+1}$, then by induction $B_{i,m} = 0$ if $\tau_i = \tau_{i+1} = \ldots = \tau_{i+m}$. Note also that in the construction above, only the subset $B_{i,m},\ i = M - m + 1, \ldots, M + K$ are required for the B-spline basis of order $m < M$ with knots ξ.

To fully understand the properties of these functions, and to show that they do indeed span the space of cubic splines for the knot sequence, requires additional mathematical machinery, including the properties of divided differences. Exercise 5.2 explores these issues.

The scope of B-splines is in fact bigger than advertised here, and has to do with knot duplication. If we duplicate an interior knot in the construction of the τ sequence above, and then generate the B-spline sequence as before, the resulting basis spans the space of piecewise polynomials with one less continuous derivative at the duplicated knot. In general, if in addition to the repeated boundary knots, we include the interior knot ξ_j $1 \leq r_j \leq M$ times, then the lowest-order derivative to be discontinuous at $x = \xi_j$ will be order $M - r_j$. Thus for cubic splines with no repeats, $r_j = 1,\ j = 1, \ldots, K$, and at each interior knot the third derivatives $(4-1)$ are discontinuous. Repeating the jth knot three times leads to a discontinuous 1st derivative; repeating it four times leads to a discontinuous zeroth derivative, i.e., the function is discontinuous at $x = \xi_j$. This is exactly what happens at the boundary knots; we repeat the knots M times, so the spline becomes discontinuous at the boundary knots (i.e., undefined beyond the boundary).

The local support of B-splines has important computational implications, especially when the number of knots K is large. Least squares computations with N observations and $K + M$ variables (basis functions) take $O(N(K+M)^2 + (K+M)^3)$ flops (floating point operations.) If K is some appreciable fraction of N, this leads to $O(N^3)$ algorithms which becomes

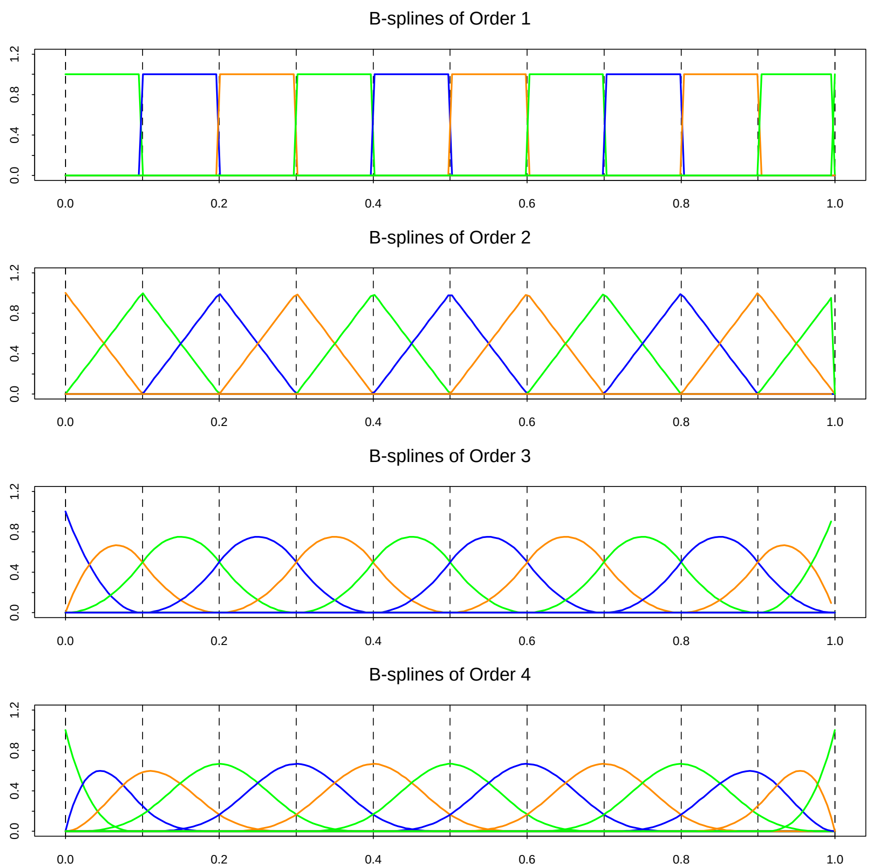

FIGURE 5.20. *The sequence of B-splines up to order four with ten knots evenly spaced from 0 to 1. The B-splines have* local support*; they are nonzero on an interval spanned by $M+1$ knots.*

unacceptable for large N. If the N observations are sorted, the $N \times (K+M)$ regression matrix consisting of the $K+M$ B-spline basis functions evaluated at the N points has many zeros, which can be exploited to reduce the computational complexity back to $O(N)$. We take this up further in the next section.

Computations for Smoothing Splines

Although natural splines (Section 5.2.1) provide a basis for smoothing splines, it is computationally more convenient to operate in the larger space of unconstrained B-splines. We write $f(x) = \sum_1^{N+4} \gamma_j B_j(x)$, where γ_j are coefficients and the B_j are the cubic B-spline basis functions. The solution looks the same as before,

$$\hat{\boldsymbol{\gamma}} = (\mathbf{B}^T\mathbf{B} + \lambda\boldsymbol{\Omega}_B)^{-1}\mathbf{B}^T\mathbf{y}, \tag{5.79}$$

except now the $N \times N$ matrix $\mathbf{N}$ is replaced by the $N \times (N+4)$ matrix $\mathbf{B}$, and similarly the $(N+4) \times (N+4)$ penalty matrix $\boldsymbol{\Omega}_B$ replaces the $N \times N$ dimensional $\boldsymbol{\Omega}_N$. Although at face value it seems that there are no boundary derivative constraints, it turns out that the penalty term automatically imposes them by giving effectively infinite weight to any non zero derivative beyond the boundary. In practice, $\hat{\gamma}$ is restricted to a linear subspace for which the penalty is always finite.

Since the columns of $\mathbf{B}$ are the evaluated B-splines, in order from left to right and evaluated at the *sorted* values of X, and the cubic B-splines have local support, $\mathbf{B}$ is lower 4-banded. Consequently the matrix $\mathbf{M} = (\mathbf{B^TB} + \lambda\boldsymbol{\Omega})$ is 4-banded and hence its Cholesky decomposition $\mathbf{M} = \mathbf{LL^T}$ can be computed easily. One then solves $\mathbf{LL}^T\boldsymbol{\gamma} = \mathbf{B}^T\mathbf{y}$ by back-substitution to give $\boldsymbol{\gamma}$ and hence the solution $\hat{f}$ in $O(N)$ operations.

In practice, when N is large, it is unnecessary to use all N interior knots, and any reasonable *thinning* strategy will save in computations and have negligible effect on the fit. For example, the `smooth.spline` function in S-PLUS uses an approximately logarithmic strategy: if $N < 50$ all knots are included, but even at $N = 5,000$ only 204 knots are used.