Smoothing Spline Notes

David W. Scott

April 26, 2000

Problem: Given values $\{f_1, f_2, \ldots, f_{m+1}\}$, find the smoothest interpolating cubic spline.

Let the m bins be defined by the mesh $\{t_1, t_2, \ldots, t_{m+1}\}$, and denote the k-th bin by $B_k = [t_k, t_{k+1})$. Let $h_k = t_{k+1} - t_k$. Define the cubic spline piecewise as

$$f(x) = a_k + b_k(t - t_k) + c_k(t - t_k)^2 + d_k(t - t_k)^3 \qquad x \in B_k\,. \tag{1}$$

Sometimes we write $f_k(x)$ to indicate which bin $f(x)$ refers.

The optimization criterion may be computed as

$$\min \int_{t_1}^{t_{m+1}} f''(t)^2 dt = \sum_{k=1}^{m} \left[4h_k c_k^2 + 12h_k^2 c_k d_k + 12h_k^3 d_k^2\right]\,. \tag{2}$$

Let $\mathbf{x} = (c_1, \ldots, c_m, d_1, \ldots, d_m)^T$. Then the criterion (2) may be written as $\mathbf{x}^T U \mathbf{x}$, where the matrix U is given by

$$U = \begin{pmatrix} Diag(4h_1, \ldots, 4h_m) & Diag(6h_1^2, \ldots, 6h_m^2) \\ Diag(6h_1^2, \ldots, 6h_m^2) & Diag(12h_1^3, \ldots, 12h_m^3) \end{pmatrix}\,. \tag{3}$$

Next we consider the constraints arising from the spline definition. We attempt to build as many constraints into the problem in a natural fashion to simplify the quadratic minimization. (Whether we succeed or not is a question.)

First, the spline must match the $m+1$ values at the knots. This leads to

$$f_k(t_k) = a_k = f_k \qquad k = 1, \ldots, m \tag{4}$$

with a special equation for the last bin:

$$f_m(t_{m+1}) = a_m + h_m b_m + h_m^2 c_m + h_m^3 d_m = f_{m+1} \tag{5}$$

The next three sets of equations constrain the spline at the $m-1$ interior knots:

$$f_k^{(r)}(t_{k+1}) = f_{k+1}^{(r)}(t_{k+1}) \qquad k = 1, 2, \ldots, m-1 \quad and \quad r = 0, 1, 2. \tag{6}$$

From equation (1), these are seen to be

$$a_k + h_k b_k + h_k^2 c_k + h_k^3 d_k = a_{k+1} \qquad r = 0 \quad k = 1, \ldots, m-1 \tag{7}$$

$$b_k + 2h_k c_k + 3h_k^2 d_k = b_{k+1} \qquad r = 1 \quad k = 1, \ldots, m-1 \tag{8}$$

$$2c_k + 6h_k d_k = 2c_{k+1} \qquad r = 2 \quad k = 1, \ldots, m-1 \tag{9}$$

We attempt to simplify these a bit. Note that equation (5) is an extension of equation (7) with $k = m$, noting equation (4). These may be re-written as

$$b_k + h_k c_k + h_k^2 d_k = \frac{f_{k+1} - f_k}{h_k} \qquad k = 1, \ldots, m\,. \tag{10}$$

Equation (9) may be simplified to

$$c_k - c_{k+1} + 3h_k d_k = 0 \qquad k = 1, \ldots, m-1\,. \tag{11}$$

The variables b_k and b_{k+1} may be eliminated from equation (8) by using equation (10) twice, once for subscript k and once for subscript $k+1$. This gives

$$h_k c_k + h_{k+1} c_{k+1} + 2h_k^2 d_k + h_{k+1}^2 d_{k+1} = \frac{f_{k+2} - f_{k+1}}{h_{k+1}} - \frac{f_{k+1} - f_k}{h_k} \qquad k = 1, \ldots, m-1 \tag{12}$$

Thus the $4m-2$ constraints may be expressed as $2m-2$ linear constraints on the variable $\mathbf{x}$, which contains just the c_k's and d_k's. In matrix form, we have

$$V\mathbf{x} = \mathbf{w}\,, \tag{13}$$

where V is an $(2m-2) \times 2m$ matrix, and w is a vector of length $2m-2$. The last $m-1$ values in $\mathbf{w}$ are 0, while the first $m-1$ are given by

$$w_k = \frac{f_{k+2} - f_{k+1}}{h_{k+1}} - \frac{f_{k+1} - f_k}{h_k} \qquad k = 1, \ldots, m-1. \tag{14}$$

The matrix V is given by

$$V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}\,, \tag{15}$$

where the $(m-1) \times m$ matrices V_{ij} are

$$V_{11} = \begin{bmatrix} h_1 & h_2 & & & \\ & h_2 & h_3 & & \\ & & \ddots & & \\ & & & h_{m-1} & h_m \end{bmatrix} \tag{16}$$

$$V_{12} = \begin{bmatrix} 2h_1^2 & h_2^2 & & & \\ & 2h_2^2 & h_3^2 & & \\ & & \ddots & & \\ & & & 2h_{m-1}^2 & h_m^2 \end{bmatrix} \tag{17}$$

$$V_{21} = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & & \\ & & & 1 & -1 \end{bmatrix} \tag{18}$$

and

$$V_{22} = \begin{bmatrix} 3h_1 & & & & 0 \\ & 3h_2 & & & 0 \\ & & \ddots & & \\ & & & 3h_{m-1} & 0 \end{bmatrix}. \tag{19}$$

The Lagrangian associated with our problem is

$$L(\mathbf{x}, \lambda) = \mathbf{x}^T U \mathbf{x} + \lambda^T (V\mathbf{x} - \mathbf{w}) = \mathbf{x}^T U \mathbf{x} + (\mathbf{x}^T V^T - \mathbf{w}^T)\lambda\,. \tag{20}$$

Taking the gradient with respect to $\mathbf{x}$ and λ, we get the following pair of matrix equations:

$$V\mathbf{x} = \mathbf{w} \qquad \textit{and} \tag{21}$$

$$2U\mathbf{x} + V^T\lambda = 0\,. \tag{22}$$

Now it is easy to see that U is positive definite, and we assume that V is of full rank, so that $(VV^T)^{-1}$ exists. This is why we eliminated the b_k's from the canonical problem.

Assuming U^{-1} exists, we may solve for $\mathbf{x}$ in equation (22) and get

$$\mathbf{x} = -\frac{1}{2} U^{-1} V^T \lambda\,, \tag{23}$$

which when substituted into equation (21) gives

$$-\frac{1}{2} V U^{-1} V^T \lambda = \mathbf{w}\,. \tag{24}$$

It seems reasonable to assume that $VU^{-1}V^T$ is invertible, so that

$$\lambda^* = -2\left(VU^{-1}V^T\right)^{-1}\mathbf{w}\,. \tag{25}$$

From equations (23) and (25), it follows that

$$\mathbf{x}^* = U^{-1}V^T\left(VU^{-1}V^T\right)^{-1}\mathbf{w}\,. \tag{26}$$

We recover the parameters c_k and d_k from $\mathbf{x}$, and using equation (10), we solve for b_k. The a_k's of course are simply the f_k values.

An Splus function to implement these formulae is attached. But it doesn't seem to give reasonable results at the moment. Try running

$$smo.spline(c(0,1,0)) \quad and \quad smo.spline(c(0,1,1,0)).$$