Appendix: Derivation of Cubic Spline Algorithm

Derivation of Cubic Spline Algorithm

Cubic Spline: Problem Set-Up

Given $\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$, find a piecewise polynomial $(x) = p_i(x)$ on $[x_i, x_{i+1}]$ with

$$p_i(x) = c_{i,1} + c_{i,2}(x - x_i) + c_{i,3}(x - x_i)^2 + c_{i,4}(x - x_i)^3,$$

satisfying[1]

1. $(x_i) = y_i$ for $i = 1, \ldots, n$;
2. $p_i(x_{i+1}) = p_{i+1}(x_{i+1})$ for $i = 1, \ldots, n-2$;
3. $p_i'(x_{i+1}) = p_{i+1}'(x_{i+1})$ for $i = 1, \ldots, n-2$;
4. $p_i''(x_{i+1}) = p_{i+1}''(x_{i+1})$ for $i = 1, \ldots, n-2$.

[1] Let us not worry about the boundary conditions yet.

Connection to Hermite Cubic Interpolation

Key Observation. If $c_{i,j}$'s are set to be

$$c_{i,1} = y_i\,, \qquad c_{i,3} = \frac{3y[x_i, x_{i+1}] - 2\sigma_i - \sigma_{i+1}}{\Delta x_i}\,,$$
$$c_{i,2} = \sigma_i\,, \qquad c_{i,4} = \frac{\sigma_i + \sigma_{i+1} - 2y[x_i, x_{i+1}]}{(\Delta x_i)^2}\,, \qquad (\star)$$

as in the Hermite cubic interpolation for some constants $\sigma_1, \sigma_2, \ldots, \sigma_n$ to be determined, then (x) satisfies the first three requirements from the previous slide.

Reduction. Determine $\sigma_1, \sigma_2, \ldots, \sigma_n$ so that the fourth requirement is satisfied.

Derivation of a Linear System for Cubic Splines

Exercise. Using ($\star$), write out the fourth requirement $p''_{i-1}(x_i) = p''_i(x_i)$ in terms of $\sigma_{i-1}, \sigma_i, \sigma_{i+1}$, where $i \in \mathbb{N}[2, n-1]$.

Derivation of a Linear System for Cubic Splines (cont')

Exercise. Express the system of $n-2$ equations for $\sigma_1, \ldots, \sigma_n$ as a matrix equation $X\boldsymbol{\sigma} = \mathbf{r}$, where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)^{\mathrm{T}}$ and $X \in \mathbb{R}^{n \times n}$ and $\mathbf{r} \in \mathbb{R}^n$ are to be found[2].

[2] Since two equations are still missing, leave the first and last rows of X and $\mathbf{r}$ empty for now. The answer is provided in the next slide.

Tridiagonal System for Cubic Splines

Notation. $\Delta x_i = x_{i+1} - x_i$ and $\nabla x_i = \Delta x_{i-1} + \Delta x_i = x_{i+1} - x_{i-1}$.

$$X = \begin{bmatrix} * & * & * & \cdots & * & * & * \\ \Delta x_2 & 2\nabla x_2 & \Delta x_1 & & & & \\ & \Delta x_3 & 2\nabla x_3 & \Delta x_2 & & & 0 \\ & & \ddots & \ddots & \ddots & & \\ 0 & & & \Delta x_{n-2} & 2\nabla x_{n-2} & \Delta x_{n-3} & \\ & & & & \Delta x_{n-1} & 2\nabla x_{n-1} & \Delta x_{n-2} \\ * & * & * & \cdots & * & * & * \end{bmatrix},$$

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_{n-2} \\ \sigma_{n-1} \\ \sigma_n \end{bmatrix}, \quad \text{and} \quad \mathbf{r} = \begin{bmatrix} * \\ 3\left(y[x_1, x_2]\Delta x_2 + y[x_2, x_3]\Delta x_1\right) \\ 3\left(y[x_2, x_3]\Delta x_3 + y[x_3, x_4]\Delta x_2\right) \\ \vdots \\ 3\left(y[x_{n-3}, x_{n-2}]\Delta x_{n-2} + y[x_{n-2}, x_{n-1}]\Delta x_{n-3}\right) \\ 3\left(y[x_{n-2}, x_{n-1}]\Delta x_{n-1} + y[x_{n-1}, x_n]\Delta x_{n-2}\right) \\ * \end{bmatrix}.$$

Remarks

An $n \times n$ matrix A is said to be **diagonally dominant** if

$$|a_{ii}| > \sum_{j=1, j \neq i}^{n} |a_{ij}| \quad \text{for each } i = 1, \ldots, n.$$

- It is known that a diagonally dominant matrix is nonsingular.
- Note that X is a tridiagonal matrix which is diagonally dominant, and thus invertible.

Implementation of Boundary Conditions

- (**clamped cubic spline**) If slopes at each end are known, fill in the first and the last equation of $X\boldsymbol{\sigma} = \mathbf{r}$ with

$$\sigma_1 = y_1', \quad \sigma_n = y_n'.$$

- (**natural cubic spline**) If the second derivatives at the endpoints are known, then use

$$2\sigma_1 + \sigma_2 = 3y[x_1, x_2] - \frac{1}{2}\Delta x_1 y_1''$$
$$\sigma_{n-1} + 2\sigma_n = 3y[x_{n-1}, x_n] + \frac{1}{2}\Delta x_{n-1} y_n''.$$

Implementation of Boundary Conditions (cont')

- (**periodic boundary condition**) If the data points come from a periodic function with period $P = x_n - x_1$ so that $\sigma_1 = \sigma_n$, then use

$$\Delta x_1 \sigma_{n-1} + 2\nabla x_1 \sigma_1 + \Delta x_{n-1}\sigma_2 = 3\left(y[x_{n-1}, x_n]\Delta x_1 + y[x_1, x_2]\Delta x_{n-1}\right)$$
$$\sigma_1 - \sigma_n = 0.$$

  Here, take $\nabla x_1 = x_2 - x_0 = x_2 - (x_{n-1} - P)$.

Implementation of Boundary Conditions (cont')

- (***not-a-knot* boundary condition**) If nothing is known about the endpoints, require $p_1(x) \equiv p_2(x)$ and $p_{n-2}(x) = p_{n-1}(x)$:

$$(\Delta x_2)^2 \sigma_1 + \left((\Delta x_2)^2 - (\Delta x_1)^2\right) \sigma_2 - (\Delta x_1)^2 \sigma_3 = 2\left(y[x_1, x_2] (\Delta x_2)^2 - y[x_2, x_3] (\Delta x_1)^2\right),$$

$$- (\Delta x_{n-1})^2 \sigma_{n-2} + \left((\Delta x_{n-2})^2 - (\Delta x_{n-1})^2\right) \sigma_{n-1} + (\Delta x_{n-2})^2 \sigma_n = 2\left(y[x_{n-1}, x_n] (\Delta x_{n-2})^2 - y[x_{n-2}, x_{n-1}] (\Delta x_{n-1})^2\right).$$

Exercise. Derive the equations shown above.