EPFL, Autumn 2025
Yoav Zemel
Guillaume Obozinski

Ho Yun

Statistical Machine Learning

Exercise sheet 8

Exercise 8.1 (Basis for Cubic Splines)

(a) It turns out that the derivative of a B-Spline is another B-Spline of lower order.

$$\tfrac{d}{dx}B_{j,d}(x) = d\left[\tfrac{B_{j,d-1}(x)}{\tau_{j+d}-\tau_j} - \tfrac{B_{j+1,d-1}(x)}{\tau_{j+d+1}-\tau_{j+1}}\right]$$

Using this fact, show that a cubic spline f with K interior knots, say $\xi_1, \ldots, \xi_K$ can be uniquely represented as a truncated power series as follows:

$$f(x) = \sum_{j=0}^{3} \beta_j x^j + \sum_{k=1}^{K} \theta_k (x-\xi_k)_+^3$$

Hint: And to prove linear independence, try differentiation.

Solution: For convenience, we shall assume throughout that the domain of all the functions involved is $[\xi_0, \xi_{K+1})$.

It suffices to show this for an arbitrary $B_{j,3}(x)$ Now, it follows from the differentiation formula that the third derivative $D_3 B_{j,3}(x) \in \text{Span}\{B_{j,0}\}_j$. Since,

$$B_{j,0}(x) = \mathbf{1}_{\{\xi_j \le x < \xi_{j+1}\}} = \mathbf{1}_{\{x \ge \xi_j\}} - \mathbf{1}_{\{x \ge \xi_{j+1}\}} = (x-\xi_j)_+^0 - (x-\xi_{j+1})_+^0$$

we have $D_3 B_{j,3}(x) \in \text{Span}\left[\{1\} \cup \{(x-\xi_k)_+^0\}_{k=1}^K\right]$. The additional element 1 here ensures that

$$\text{Span}\left[\{1\} \cup \{(x-\xi_k)_+^0\}_{k=1}^K\right] = \text{Span}\{\mathbf{1}_{\{\xi_k \le x < \xi_{k+1}\}}\}_{k=1}^K$$

So,

$$D_3 B_{j,3}(x) = c_0 + \sum_{k=1}^{K} c_k (x-\xi_k)_+^0$$

Integrating thrice, gives

$$B_{j,3}(x) = \sum_{i=0}^{3} \beta_i x^i + \tfrac{1}{6}\sum_{k=1}^{K} c_k (x-\xi_k)_+^3$$

To show that the representation is unique, assume that $f(x) = 0$ everywhere. Then, differentiating thrice at $x < \xi_1$, $\xi_j \le x < \xi_{j+1}$ and $x \ge \xi_K$ gives $6\beta_3 = 0$

$$f'''(x) = 6\beta_3 + 6\sum_{k=1}^{j} \theta_k \mathbf{1}_{\{x \ge \xi_k\}} = 0$$

for $j = 1, \ldots, K$. It follows, that $\beta_3 = 0$ and $\theta_k = 0$ for $j = 1, \ldots, K$. Using this, one can deduce that $\beta_0 = \beta_1 = \beta_2 = 0$ and the conclusion follows.

(b) Spline predictors often have high variance in the outer range. For this reason, it is desirable to require that the spline be linear before the first and after the last boundary point. This additional constraint brings down the variance. Such splines are known as natural splines. Show that the cubic spline f is a natural spline if and only if the corresponding coefficients satisfy the following condition:

$$\beta_2 = 0, \beta_3 = 0, \sum_{k=1}^{K} \theta_k = 0 \text{ and } \sum_{k=1}^{K} \xi_k \theta_k = 0.$$

Solution: Clearly, if f linear before $x = \xi_1$ and after $x = \xi_K$, then it follows that $\sum_{j=0}^{3} \beta_j x^j$ is linear and so is $\sum_{j=0}^{3} \beta_j x^j + \sum_{k=1}^{K} \theta_k (x - \xi_k)^3$ which is equal to

$$\left(\beta_3 + \sum_{k=1}^{K} \theta_k\right) x^3 + \left(\beta_2 - 3\sum_{k=1}^{K} \theta_k \xi_k\right) x^2 + \left(\beta_1 + 3\sum_{k=1}^{K} \theta_k \xi_k^2\right) x + \left(\beta_0 - \sum_{k=1}^{K} \theta_k \xi_k^3\right)$$

The conclusion follows.

(c) An alternative to fitting splines under these additional constraints is to use a basis for natural cubic splines. Show that the following functions form a basis for natural cubic splines:

$$\nu_1(x) = 1, \nu_2(x) = x \text{ and } \nu_{k+2}(x) = d_k(x) - d_{K-1}(x)$$

where,

$$d_k(x) = \tfrac{(x-\xi_k)_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_k}$$

Hint: To prove linear independence, try differentiation.

Solution: Clearly, from (b) we have

$$\theta_K = -\sum_{k=1}^{K-1} \theta_k \text{ and } (\xi_{K-1} - \xi_K)\theta_{K-1} = -\sum_{k=1}^{K-2} (\xi_k - \xi_K)\theta_k$$

Substituting this in the power series, together with the constraints $\beta_2 = \beta_3 = 0$ from (b) we have

$$\begin{aligned} f(x) &= \beta_0 + \beta_1 x + \sum_{k=1}^{K-1} \theta_k (\xi_K - \xi_k) d_k(x) \\ &= \beta_0 + \beta_1 x + \sum_{k=1}^{K-2} \theta_k (\xi_K - \xi_k)(d_k(x) - d_{K-1}(x)) \end{aligned}$$

To show that $\{\nu_k\}_{k=1}^{K}$ are linearly independent assume that $f(x) = 0$. Then, differentiating thrice the former expression above at $\xi_j \leq x < \xi_{j+1}$ to get

$$f'''(x) = 6\sum_{k=1}^{K-1} \theta_k \left[\mathbf{1}_{\{x \geq \xi_k\}} - \mathbf{1}_{\{x \geq \xi_K\}}\right] = 6\sum_{k=j}^{K-1} \theta_k = 0$$

Thus, $\theta_j = 0$ for $1 \leq j \leq K-2$. And using this, it follows that $\beta_0 = \beta_1 = 0$. So, we are done.

Practical exercise

Exercise 8.2 (Smoothing spline practical exercise) In this exercise we will use some well known R package to apply smoothing splines to a real data problem.

(a) Read Chapters 7.5, 7.8.1 and 7.8.2 in [ISL] on smoothing splines to familiarize yourself with the packages in R that you will need for this exercise.

(b) Load the *triceps* data which is available in the `MultiKink` package. The data contains the measurement of the triceps skin fold of 892 females (variable triceps) and we want to model its association with age.

(c) Plot the scatter for triceps and age.

(d) Fit a piecewise linear regression with knots at 5,10,20,30 and 40. Use the `fitted` function to give fitted lines from your model fit.

(e) Using the same knots as above, fit a quadratic piecewise regression

(f) Fit a cubic polynomial spline and a natural cubic spline for your problem with the same knots.
Hint: The function `bs()` in the `splines` package generates the B-spline basis matrix for a polynomial spline, and the function `ns()` in the same library generates the B-spline basis matrix matrix for a natural cubic spline.

(g) Compare the number of regression parameters for the natural spline with that of the polynomial spline. Which is larger, and why?

(h) Fit a natural spline with 6 degrees of freedom and compare it with the natural spline using knots $= c(5, 10, 20, 30, 40)$. What is the difference?

(i) Calculate the MSE (or the root mean squared error) for the models using natural cubic splines with degrees of freedom from 2 (linear model) up to 20. You can use the library `caret`.

(j) Fit a natural cubic smoothing spline (with knots at all the datapoints). Suggest a way to choose the penalization parameter λ for your natural cubic smoothing spline.

Exercise 8.3 (Thin Plate Splines) In this exercise we will work with the `RMprecip` dataset in the `fields` package. Fit a thin plate spline with the x and y covariate on *elevation* using the `Tps()` function and plot your surface using the `surface( , drop.z=TRUE)` function. *Hint: this is a very easy exercise with just one line of code.*