Tutorial and Practice Questions 1: Splines

Topics in Advanced Regression

April 2025

Question 1

Simulate data from the following model, where $\mathbf{x}$ is a vector of 100 random values sampled from $U(0,1)$. Define two knots, one at 0.33 and the other at 0.66.

$y_i \sim N(\mu_i, 0.5^2), \qquad \mu_i = 5 + \sin(3\pi(x_i - 0.6))$

For each of the following define the basis functions / matrix and plot these functions. Then fit them to the data above and create a plot to illustrate the fitted curves, lines.

- Piecewise constant
- Piecewise linear
- Piecewise linear, and continuous at knots (broken stick)
- Piecewise cubic polynomial
- Piecewise cubic polynomial, and continuous at knots
- Piecewise cubic polynomial, and continuous up to 1st derivative at knots
- Piecewise cubic polynomial, and continuous up to 2nd derivative at knots
- Cubic spline (using truncated power basis)
- Natural spline (for the moment use the ns() function in packages splines)
- B-spline basis function (for the moment use the bs() function in packages splines, but see exercise below)

Create another plot where you add confidence bands to one of the fitted splines (natural or B-spline).

Some functions / ideas that you may find useful:

- `cut()` will cut the range of a variable x into intervals.
- Interactions in linear models are useful for estimating a different line or curve in each section (not continuous).

- See the summary of constrained least squares below for fitting linear regression models with constraints (continuous, and continuous w.r.t. first derivative).

http://people.duke.edu/~hpgavin/cee201/constrained-least-squares.pdf

Constrained linear least squares

Minimize $(Y - X\beta)'(Y - X\beta)$ subject to constraints $C\beta = b$, using Lagrange multiplier λ: $\lambda'(C\beta - b)$

Solution:

$$\begin{bmatrix} 2X'X & C' \\ C & 0 \end{bmatrix} \begin{bmatrix} \beta \\ \lambda \end{bmatrix} = \begin{bmatrix} 2X'Y \\ b \end{bmatrix}$$

Question 2

Write an R function to create a cubic spline basis based on a truncated power basis. Construct the basis matrix X. Fit a cubic spline to the LIDAR data. Add the fitted spline curve to a plot with the observations. The LIDAR data are available in the `SemiPar` library.

Question 3

Construct the B-spline basis functions (see ESL Appendix pg. 186): Haar basis, linear, quadratic, cubic B-spline basis functions. x between -1, 1. Create basis matrix X. Plot the basis functions.

Penalized Regression Splines

In practice (e.g. in R packages) regression splines are splines where the user chooses the knots and the spline is fitted with or without penalty (in which case they become penalized regression splines).

We can solve the penalized regression problem as follows:

$$\hat{\beta} = (X'X + \lambda\Omega)^{-1}X'Y$$

The main difficulty here is: how to obtain Ω, the penalty matrix? This is one reason for the popularity of P-splines, for which the penalty matrix is easy to calculate (see notes).

Fit P-splines to the LIDAR data. Start with a B-spline basis, calculate the penalty matrix, and fit to the data. Repeat for $\lambda = 0, 0.1, 0.5, 1, 5, 10$. Plot the data and add all of the resulting P-splines to the plot.

Repeat this for a range of λ values between 0 and 20, each time calculating the GCV value. Plot GCV as a function of λ and choose a suitable λ.

Smoothing Splines

In practice (e.g. in R packages) smoothing splines are handled like the regression splines above (basically, a penalized regression spline). A large (enough) number of knots is chosen, and then a roughness penalty is added. The smoothing parameter is often chosen using GCV. So, there is nothing extra you need to do here.

Introduction to Statistical Learning (James et al.)

Chapter 7 Exercises 1, 3, 4

Elements of Statistical Learning (Hastie et al.)

Exercise 5.1 - 5.4