Today

1. Basis Functions
2. Regression Splines
3. Smoothing Splines
4. Generalized Additive model (GAM)

1. Basis Functions

- Polynomial and piecewise-constant regression models are special cases of a basis function approach.
- Instead of fitting a linear model of X, we fit the model

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_2) + \cdots + \beta_K b_K(x_i) + \varepsilon_i \quad i=1,\cdots,n$$

Notes:

① $b_k(X)$: a family of functions or transformations of X.

② $b_1(x) \cdots b_K(x)$ are fixed and known.

Eg: For polynomial regression, $b_j(x_i) = x_i^j$

For piecewise-constant regression $b_j(x_i) = I(c_j < x_i < c_{j+1})$

- Use least squares to estimate the coefficients

2. Regression splines

* piecewise polynomials
  - instead of fitting a high-degree polynomial over the entire range of X, we fit low-degree polynomials over different regions of X.

Eg: A piecewise polynomial with a single knot at a point c has forms

$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \varepsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \varepsilon_i & \text{if } x_i \geq c. \end{cases}$$

⇒ Each of these polynomial functions can be fit using least squares
See Figure 7.3
- Better to add constraints to the polynomials
  i.e. continuity of the polynomials
  continuity of the first derivative
  Continuity of the second derivative
- A cubic spline with K knots uses a total of K+4 d.f.
- General definition of a degree-d spline is that it is a piecewise degree-d polynomial, with continuity in derivatives up to degree d-1 at each knot

* The spline Basis Representation
- use the basis model to represent a regression spline
e.g. a cubic spline with K knots can be modeled as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \varepsilon_i$$

where

$$b_1(x_i) = x_i$$
$$b_2(x_i) = x_i^2$$
$$b_3(x_i) = x_i^3$$
$$b_{k+3}(x_i) = (x_i - \xi_k)_+^3 \quad k=1,\ldots K$$

$$\left((x_i - \xi_k)_+^3 = \begin{cases} (x_i - \xi_k)^3 & \text{if } x_i > \xi_k \\ 0 & \text{otherwise} \end{cases} \right)$$

Notes: ① In other words, in order to fit a cubic spline to a data set with K knots, we perform least squares regression with an intercept and 3+K predictors, of the form $X, X^2, X^3, (X-\xi_1)_+^3, \ldots (X-\xi_K)_+^3$

② A total of K+4 coefficients

③ Fitting a cubic spline with K knots uses K+4 d.f.

④ Splines can have high variance at the outer range of the predictors

⑤ A natural cubic spline extrapolates linearly beyond the boundary knots. i.e. the function is required to be linear at the boundary.

- this add $4 = 2 \times 2$ constraints
  - allow us to put more internal knots for the same degrees of freedom as a regular cubic spline.

* Knot placement

- one strategy is to decide K, # of knots, and then place them at appropriate quantiles of the observed X.
  e.g. specify the desired degrees of freedom, and then let software place the corresponding number of knots at uniform quantiles of the data.
- How many knots? or # of degrees of freedom?
  use CV.

3. Smoothing splines

- consider this criterion for fitting a smooth function $g(x)$ to some data

$$\underset{g \in S}{\text{minimize}} \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

Notes ① the first term is RSS, and tries to make $g(x)$ match the data at each x_i

② The second term is roughness penalty and controls how wiggly $g(x)$ is. It is modulated by the tuning parameter $\lambda \geq 0$

i.e. the smaller λ, the more wiggly the function, eventually interpolating y, when $\lambda = 0$

as $\lambda \to \infty$, the function $g(x)$ becomes linear.

~~* choose λ~~

⇒ The solution is a natural cubic spline, with a knot at every unique value of x_i. The roughness penalty still controls the roughness via λ.

Notes: ① Smoothing splines avoid the knot-selection issue, leaving a single λ to be chosen.

② The ~~ag~~ algorithm details are too complex to describe here.

③ In R: smooth.splines( )

④ The vector of n fitted values can be written as $\hat{g}_\lambda = S_\lambda y$ where S_λ is a $n \times n$ matrix (determined by x_i and λ)

⑤ The effective degrees of freedom are given by

$$df_\lambda = \sum_{i=1}^{n} \{S_\lambda\}_{ii}$$

⇒ then we can specify df rather than λ!

R: smooth.spline(age, wage, df=10)

⑥ The leave-one-out CV error is given by

$$RSS_{cv}(\lambda) = \sum_{i=1}^{n} \left(y_i - \hat{g}_\lambda^{(-i)}(x_i)\right)^2 = \sum_{i=1}^{n} \left[\frac{y_i - \hat{g}_\lambda(x_i)}{1 - \{S_\lambda\}_{ii}}\right]^2$$

In R: smooth.spline(age, wage)

4. Generalized Additive Model

- Allows for flexible non-linearities in several variables, but retains the additive structure of linear models.

$$y_i = \beta_0 + \beta_1 b_1(x_{i1}) + \cdots + \beta_K b_K(x_{i1}) + \varepsilon$$

$$\Rightarrow y_i = \beta_0 + f_1(x_{i1}) + \cdots + f_p(x_{ip}) + \varepsilon$$

Notes: ① Can fit a GAM using, e.g. natural splines

lm(wage ~ ns(age, df=5) + ns(year, df=5) + education)

② coefficients not that interesting; fitted functions are.

③ Can mix terms — some linear, some nonlinear.

④ use anova( ) to compare models

⑤ Can use smoothing splines or local regression as well

gam(wage ~ s(year, df=5) + lo(age, span=.5) + education)

⑥ GAMs are additive, although low-order interactions can be included in a natural way using, e.g. bivariate smoothers or interactions of the form ns(age, df=5) : ns(year, df=5)

⑦ GAM for classification

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 f_1(X_1) + \cdots + f_p(X_p)$$

e.g. gam(I(wage)>250) ~ year + s(age, df=5) + education, family = binomial)