Moving beyond linearity *

- **Polynomial regression** extends the linear model by adding extra predictors, obtained by raising each of the original predictors to a power => i.e cubic regression (x, x^2, x^3). This approach provides a simple way to provide a non linear fit to data.
- **Step functions** cut the ~~range~~ range of variable into K distinct regions in order to produce a qualitative variable. This has the effect of fitting a piecewise constant function
- **Regression splines** are more flexible than polynomials and step functions, and in fact are an extension of the two. They involve dividing the range of x into K distinct regions. Within each region, a polynomial function is fit to the data. However these polynomials are constrained so that they join smoothly at the region boundaries, or knots. Provided that the interval is divided into enough regions, this can produce an extremely flexible fit
- **Smoothing splines** are similar to regression splines, but arise in a slightly different situation. Smoothing splines result from minimizing a residual sum of squares criterion subject to a smoothness penalty.
- **Local regression** is similar to splines, but differs in an important way. The regions are allowed to overlap, and indeed they do so in a very smooth way.
- **Generalized additive models** allow us to extend the methods above to deal with multiple predictors

Steps function

Polynomial functions put a global structure but not step functions.

linear => $y = \beta_0 + \beta_1 C_1(x) + \beta_2 C_2(x) + \dots + \beta_K C_K(x) + \varepsilon$ with $C_0(x) = I(x < c_1)$, $C_1(x) = I(c_1 \leq x < c_2)$, ...

logistic $P(y|x=x) = \frac{\exp(\beta_0 + \beta_1 C_1(x) + \dots)}{1 + \exp(\beta_0 + \beta_1 C_1(x) + \dots)}$

→ Predicts a response of $\beta_0 + \beta_j$ for $c_j \leq x < c_{j+1}$ so β_j represents the average increase in the response for x in $c_j < x < c_{j+1}$ relative to $x < c_1$

Polynomials and piecewise regression are a special case of basis function

$y = \beta_0 + \beta_1 b_1(x) + \dots + \beta_K b_K(x) + \varepsilon$ where $b_1(.), \dots b_K(.)$ are fixed and known. Hence all inferences tools for linear models of ch3 can be used). We can also use wavelets, Fourier to construct regression splines as basis functions.

Regression spline

- Instead of fitting a high-degree polynomial over the entire range of x, piecewise polynomial regression involves fitting separate low-degree polynomials over different regions of x (points where coeff change are knots). i.e $y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \varepsilon_i & \text{if } x_i < c \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \varepsilon_i & \text{if } x_i > c \end{cases}$

More knots => more flexible.

Each constraint (continuity, continuity first deriv, and second deriv) that we impose on the piecewise cubic polynomials effectively frees up one degree of freedom. The 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. Therefore

But how to do it?

The most direct way to represent a cubic spline is to start off with a basis for a cubic polynomial (x, x^2, x^3) and then add one truncated power basis function per knot $\Rightarrow h(x,\xi) = (x-\xi)^3_+ \Rightarrow \begin{cases} (x-\xi)^3 & \text{if } x > \xi \\ 0 & \text{otherwise} \end{cases}$

Hence we perform least squares regression with an intercept and $3+k$ predictors, of the form $x, x^2, x^3, h(x,\xi_1), \dots h(x,\xi_k)$ where $\xi_1, \dots \xi_k$ are the knots. The spline can have high variance at the boundary, so we can add a constraint at the boundary.

knots place → uniform
→ more knots where wig rapidly, knots number → CV

$\Rightarrow$ hence splines introduce flexibility by increasing the number of knots but keeping the degree fixed.

Smoothing splines

- we want g that make RSS small, but that is also smooth $\Rightarrow \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \underbrace{\int g''(t)^2 dt}_{\text{to realize variability}}$
- if $\lambda \to \infty \Rightarrow$ perfectly smooth $\Rightarrow$ straight line, close $\lambda \Rightarrow$ CV.
  hence λ control the bias-variance trade-off of the smoothing spline.
- we have seen that a smoothing spline is simply a natural cubic spline with knots at every unique value of x_i. It might seem that a smoothing spline will have far too many degrees of freedom since a knot at each data points allows a great deal of flexibility. But λ controls the roughness, and as $\lambda \to \infty$ $df \to n \to 2$.

Local regression

- It involves computing the fit at a target point x_0 using only the nearby training observations.

small s $\Rightarrow$ more local and wiggly ! weights k_{i0} will differ for each value of x_0
(use CV)

1. Gather the fraction $s = k/n$ of training points whose x_i are closest to x_0
2. Assign a weight $k_{i0} = k(x_i, x_0)$ to each point in this neighborhood, so that the point furthest from x_0 has weight zero and the closest has the biggest weight. All but these k nearest neighbors get weights zero
3. Fit weighted least squares regression of the y_i on the x_i using the weights by finding $\hat{\beta}_0, \hat{\beta}_1$ that minimize $\sum_{i=1}^{n} k_{i0}(y_i - \beta_0 - \beta_1 x_i)^2$
4. The fitted value at x_0 is given by $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$

Generalized additive model

- GAM provide a general framework for extending a standard linear model by allowing non linear functions of each of the variables, while maintaining additivity

$y_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \varepsilon_i$ we can use spline, local regression, polynomials to create any

- GAM allow us to fit a non-linear f_j to each x_j, so that we can automatically model non-linear relationship that standard linear.
- Because the model is additive, we can examine the effect of each x_j on y individually while holding the others fixed. But we can miss out because of (only using) additivity, but we can add manually $x_j \times x_k$ or $f_{jk}(x_j, x_k)$
- $\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + f_1(x_1) + f_2(x_2) + \dots + f_p(x_p)$