Statistical Methods for Bioinformatics

II-4: Beyond Linearity

3 May 2016

Beyond Linearity

- When a predictor has a non-linear relationship with the response variable the default approach is to transform the predictor to maintain the basic linear form.

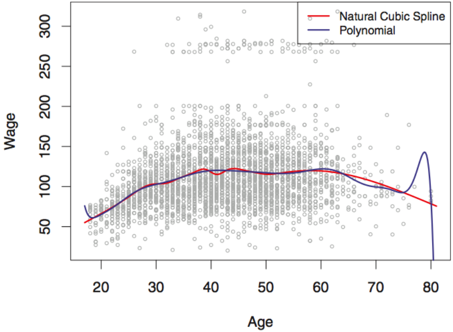
$$g(Y) = \beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m + \varepsilon$$

- A simple transformation may suffice e.g. log or root transformations
- The traditional approach is to use polynomial expansions

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \varepsilon$$

The problem with polynomials

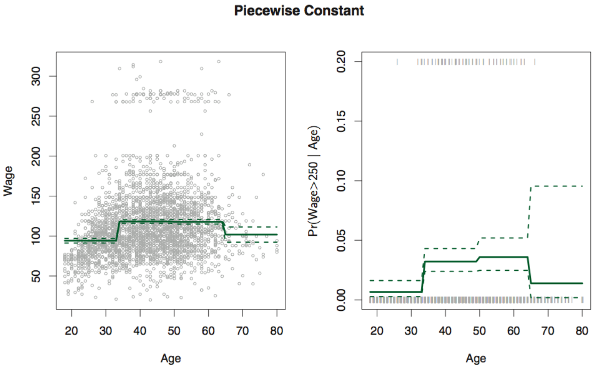
- A polynomial series generates a global fit; i.e. it describes the whole range of the predictor.
- Tweaking the coefficients for one region can cause the function to flap about madly in more remote, data-sparse, regions.



On the Wage data set, a natural cubic spline with 15 degrees of freedom is compared to a degree-15 polynomial. Polynomials can show wild

Alternatives: splitting up

- We can break up the range of X into bins; an ordered categorical variable with estimated means.



The Wage data. Left: Solid curve: fitted value from a least squares regression of wage (in thousands) using step functions of age. Dotted curves indicate 95 % confidence interval. Right: Model of binary event wage>250k using logistic regression with step functions of age; showing posterior probability.

Basis functions

Polynomial and piecewise constant-regression functions are expression of the general model:

$$y = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + \ldots + \beta_n b_n(x_i) + \varepsilon_i$$

with $b(.)$ some defined basis function

- $b_j(x) = x^j$ in the case of polynomials.

This approach allows to fit flexible functions, while holding on to the linear model with its many advantages, such as paramater estimation approaches and error/significance inference.

Regression splines as basis functions

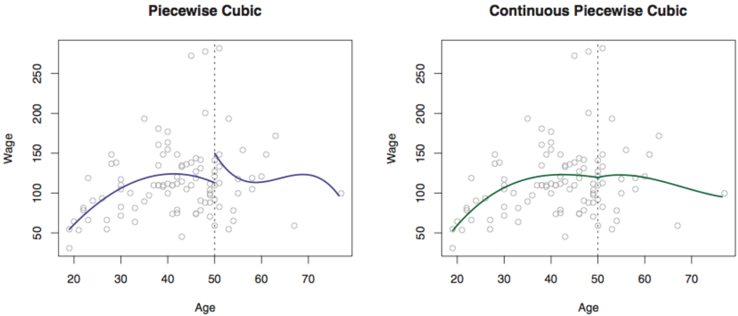
- Piecewise polynomial regression: fitting low level polynomial over intervals of X.

$$\begin{cases} \beta_{01} + \beta_{11}x_I + \beta_{21}x_I^2 + \beta_{31}x_I^3 & x_I < c \\ \beta_{02} + \beta_{12}x_I + \beta_{22}x_I^2 + \beta_{32}x_I^3 & x_I \geq c \end{cases}$$

Adding more intervals (knots) makes the function more flexible.

Constraints

- If we do not insist on continuity we get awkward results
- Even a constraint on the response value at the interval borders provides unrealistic fits.

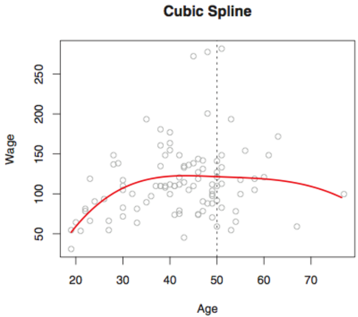


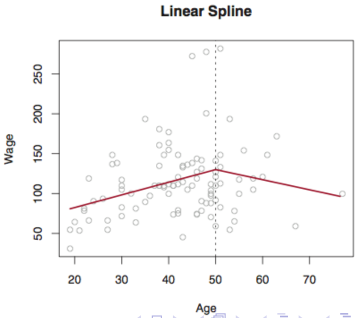
Piecewise Cubic

Continuous Piecewise Cubic

Constraints

- Ensuring continuity to the second derivative gives smoother transitions and reduces the degrees of freedom needed for the fit
- A spline of degree D is a function formed by connecting polynomial segments of degree D so that:
  - the function is continuous,
  - the function has $D - 1$ continuous derivatives (the Dth derivative is constant between knots)





What is a spline?

- Historically: a flexible ruler used to draw curves.Thin wooden strips to interpolation from the key points of a design into smooth curves. The strips are held in place at defined points using weights called ”ducks”. Between the fixed points would assume shapes defined by minimum strain energy.
- In statistics etc: a ”spline” is a smooth, piecewise polynomial approximation of a continuous function.

Form of a cubic spline

- A cubic spline with k knots can be modelled as:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + \ldots + \beta_{K+3} b_{+3}(x_i) + \varepsilon_i$$

- One representation starts with a normal cubic polynomial: x, x^2, x^3, then add truncated power basis functions per knot:

$$h(x, \xi) = (x - \xi)^3_+ = \begin{cases} (x - \xi)^3 & \text{if } x > \xi \\ 0 & \text{otherwise,} \end{cases}$$

- Limited increase in use of degrees of freedom: a cubic spline with K knots uses K+4 degrees of freedom.

The truncated power basis function in action

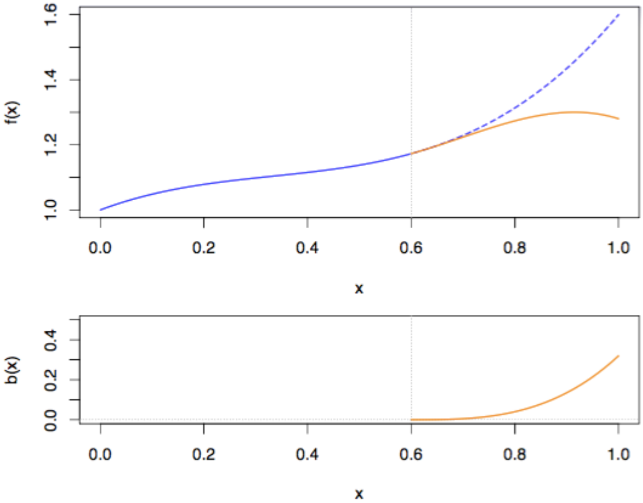
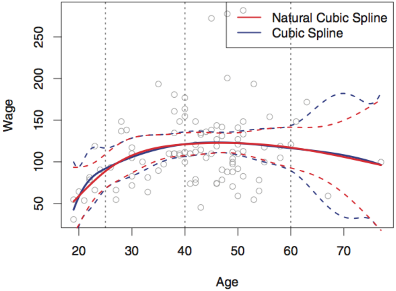


image by Trevor Hastie, Robert Tibshirani

Natural Splines: additional constraints

- We know the behavior of polynomials fit to data tends to be erratic near the boundaries
- Locally fit polynomials fit beyond the boundary knots behave even more wildly than the corresponding global polynomials in that region.
- A “natural” cubic spline adds constraints, so that the function is linear beyond the boundary knots.
  - 4 degrees of freedom are saved



Decisions with Regression Splines

1. select the order of the spline
2. the number of knots
3. placement of knots

- One approach is to parameterize a family of splines by degrees of freedom, and have the observations determine the positions of the knots.
- In practice it is common to place knots in a uniform fashion
- Decide form by cross-validation

Smoothing splines: roughness penalty

- Purpose:
  - Provide a good fit to the data to explore and present the relationship between the explanatory variable and the response variable
  - To obtain a curve estimate that does not display too much rapid fluctuation
- How to make a compromise between the two rather different aims in curve estimation?
- Smoothing splines penalize for roughness quantified by:

$$\int g''(t)^2$$

Smoothing splines

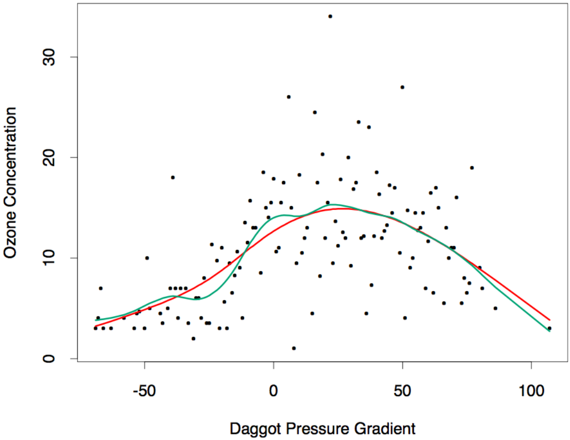
- We try to fit a function g that fits the data as good as possible, but it should avoid overlearning. A reasonable demand is for the function to be "smooth". We use the following optimization function.

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

- if $\lambda = 0$ you'll get a perfect match to the training data, if $\lambda \to \infty$ then you'll get a function without inflections: a line.
- Remarkably, it can be shown that this formula has an explicit, finite-dimensional, unique minimizer which is a natural cubic spline with knots at the unique values of the xi, i = 1,...,N

Smoothing splines: the λ parameter

- The smoothing parameter controls the variance/bias balance
  (image from The Elements of Statistical Learning)



Smoothing splines: the λ parameter

- The smoothing parameter constrains the degrees of freedom of the fit. $df(\lambda)$ decreases from n for $\lambda = 0$ to 2 as $\lambda \to \infty$. Assume the estimated fit $\hat{g}_\lambda = S_\lambda Y$, then the effective degrees of freedom is given by $df_\lambda = \sum_{i=1}^{n} \{S_\lambda\}_{ii}$
- Cross-validation is a good way to estimate an adequate λ. There is a very computationally efficient Leave-One Out Cross-Validation solution:

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

- Similar efficient LOOCV solutions exist for the regression splines

Exercises

Do Lab chapter 7 till 7.8.3
Read remainder of chapter 7 and chapter 8.

- Exercises 1 and 9