Statistical Methods for Bioinformatics

II-4: Beyond Linearity

Today

- Non-linearity in the (Generalized) Linear Model
  - limitations of polynomial global fits
- Linear Model of Basis Functions
- Splines
  - Cubic Regression Spline and the truncated power-basis function
  - Natural Cubic Regression Spline
  - Smoothing Spline
- Non-parametric regression
  - LOESS
- Example application of non-linear models
- Generalized additive models

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.

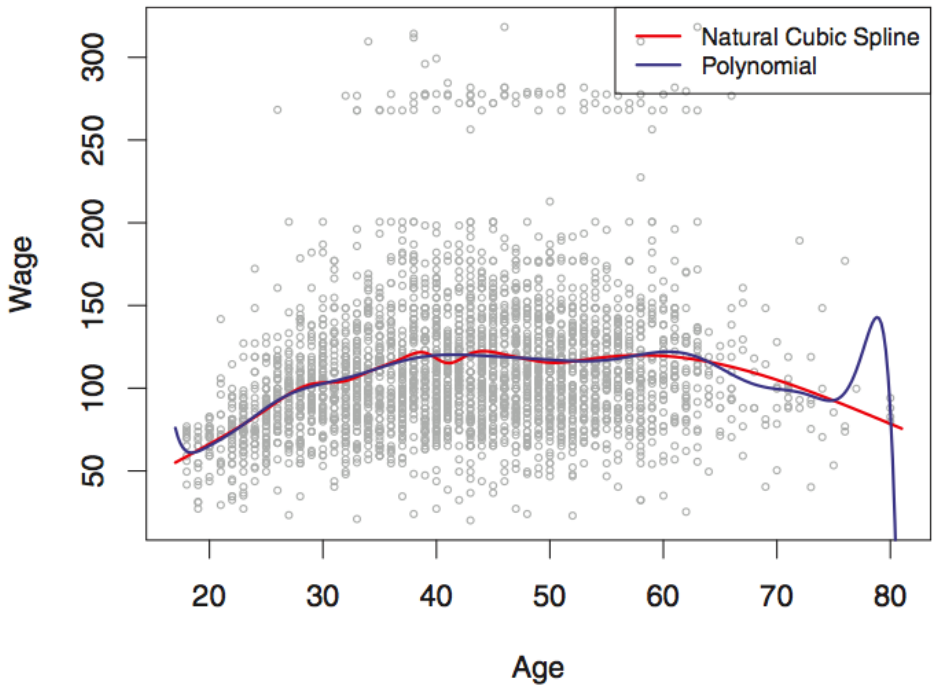
$$g(Y) = \beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m + \varepsilon$$

- 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

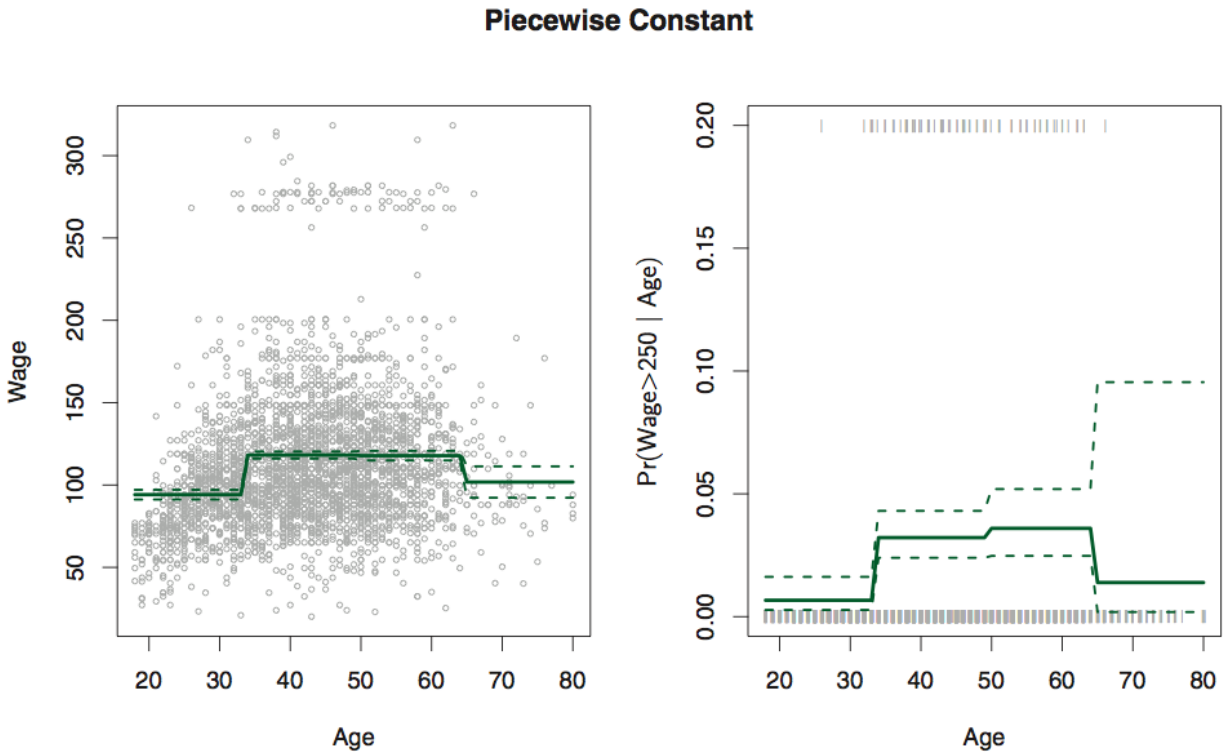
- A polynomial series generates a global fit; i.e. it describes the whole range of the predictor.
- Tweaking the coefficients for one region can cause the function to flap about madly in more remote, data-sparse, regions.



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 behavior, especially near the tails.

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.

Step functions

In step functions you define a fit per interval. For a constant response prediction per interval:

$$y = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \beta_3 C_3(x_i) + \ldots + \beta_n C_n(x_i) + \varepsilon_i$$

with C (X) indicator variables that become 1 or 0 depending on the value of X, and interval boundaries

$$C(x) = \begin{cases} 1 & bound_{lower} \leq x < bound_{higher} \\ 0 & x < bound_{lower} \vee x \geq bound_{higher} \end{cases}$$

This can give stable fits, with flexibility based on location and number of breaks, but normally quite terrible bias.

Fitting higher order functions per interval

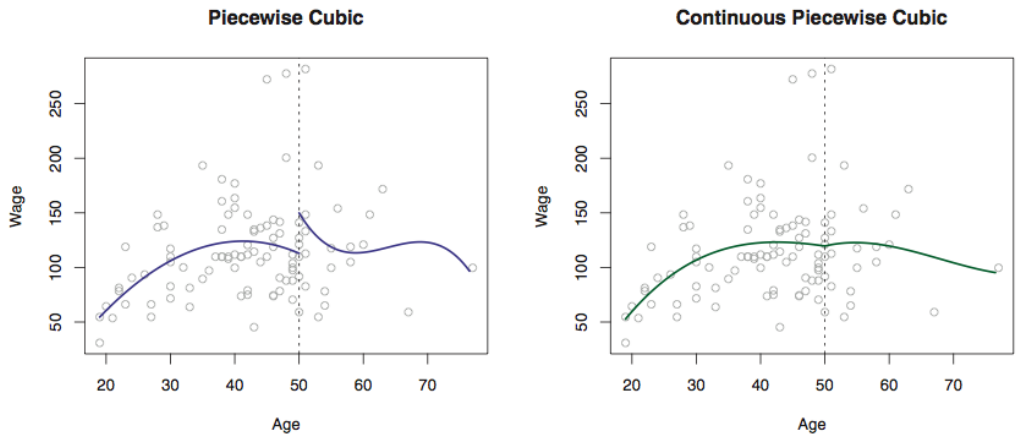
Piecewise polynomial regression: fitting low level polynomial over intervals of X.

$$\begin{cases} \beta_{01} + \beta_{11}x_1 + \beta_{21}x_1^2 + \beta_{31}x_1^3 & x_1 \leq \textit{bound} \\ \beta_{02} + \beta_{12}x_1 + \beta_{22}x_1^2 + \beta_{32}x_1^3 & x_1 > \textit{bound} \end{cases}$$

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

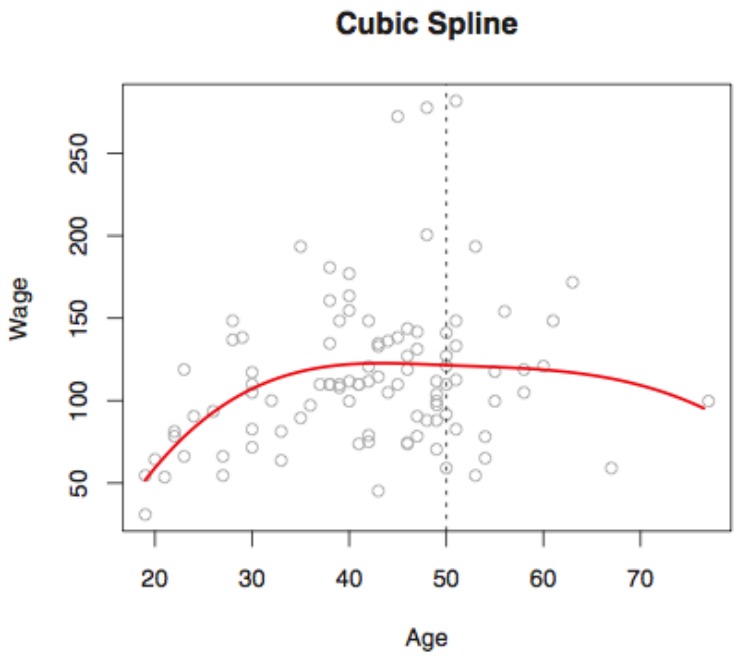
Constraints to obtain smooth functions

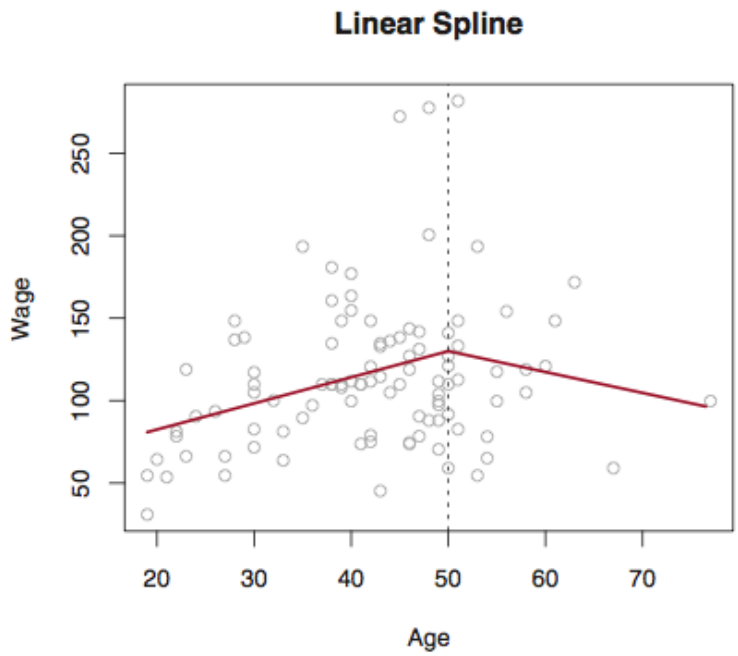
- If we do not insist on continuity we get awkward results
- Just a constraint on the response value at the interval borders still provides unrealistic fits.



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: 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.

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

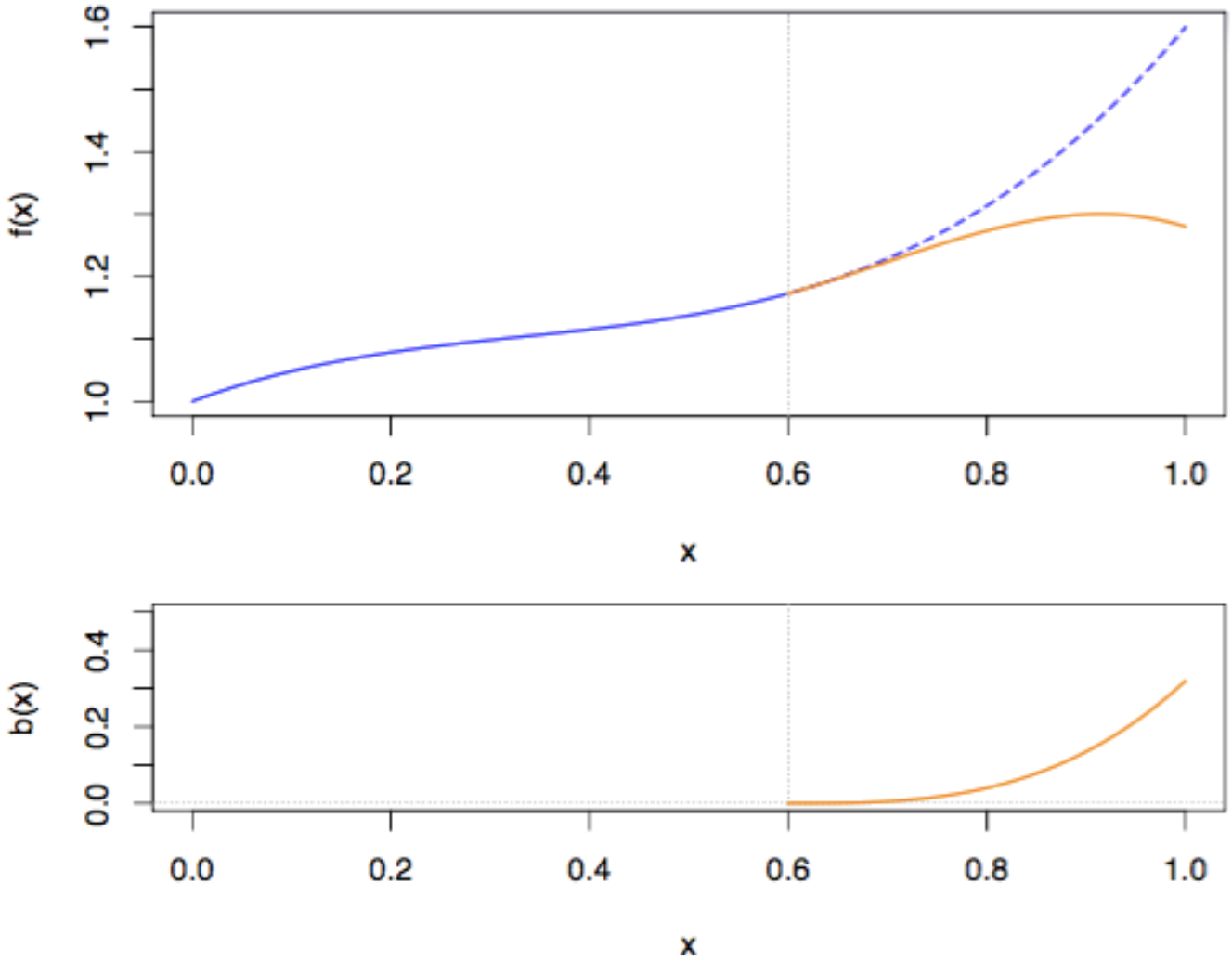


image by Trevor Hastie, Robert Tibshirani

Practicalities around regression splines

- In principle you could go to higher degree splines, e.g. with 4th degree polynomials. In practice, this is hardly ever warranted.
- The truncated power function is not too useful in practice due to numerical instability issues.
  - Powers of large numbers can cause problems with overflow/rounding
  - The B-spline basis is more suitable (stable), esp. with many knots (but of a more complicated form)
  - B-splines are equivalent to the formulation shown here
- In R you can fit a cubic regression spline with the **gam** package using the **bs** function

Question: why this comment?

'Unfortunately, splines can have high variance at the outer range of the predictors—that is, when X takes on either a very small or very large value'

Natural Splines: additional constraints

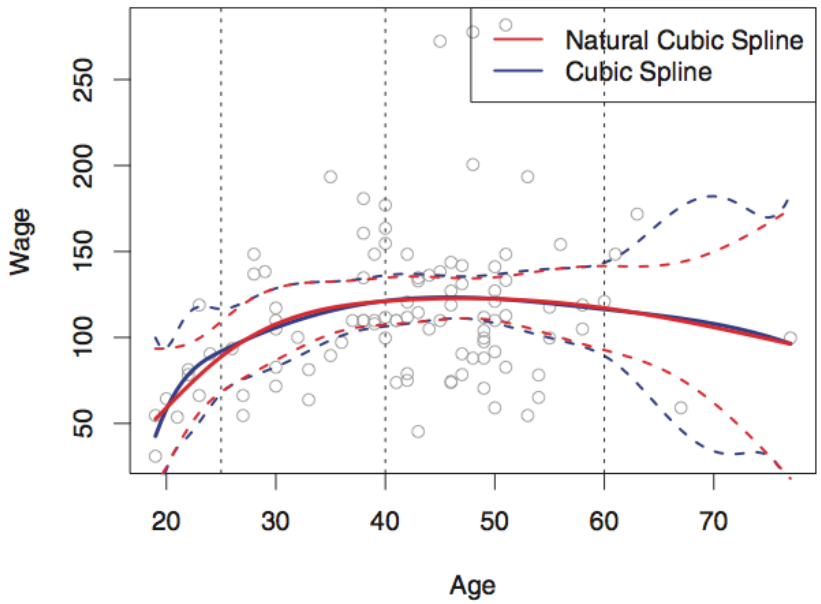
- We know the behavior of polynomials fit to data tends to be erratic near the boundaries
- Locally fit polynomials fit behave even more wildly there, and inference beyond the range is unreliable
- A "natural" cubic spline adds constraints, so that the function is linear beyond the boundary knots.
  - The following holds for a spline **g** fit on **n** observations in ascending order $x_0 \cdots x_n$:

$$g''(x_0) = g''(x_n) = 0$$

  - Boundary knots are required, but 4 degrees of freedom are saved to a cubic spline **with the same number of knots**

Natural Splines: additional constraints

- As can be seen below, the variability in the fit is reduced in the boundary regions (Confidence Intervals are shown)



In R you can fit a cubic regression spline with the **gam** package using the **ns** function

Natural Splines: expressed in base functions

A natural cubic spline model with K knots is represented by K basis functions:

$$y = \beta_0 + \beta_1 X + \beta_2 b_{k+2}(X) + \beta_3 b_{k+3}(X) + \ldots + \beta_K b_K(x_i) + \varepsilon_i$$

with $b_{k+2}(X) = d_k(X) - d_{k-1}(X)$ with $(X - \xi_k)^3_+$ the truncated base function as before:

$$d_k(X) = \frac{(X - \xi_k)^3_+ - (X - \xi_K)^3_+}{\xi_K - \xi_k}$$

(from Elements of Statistical Learning)

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

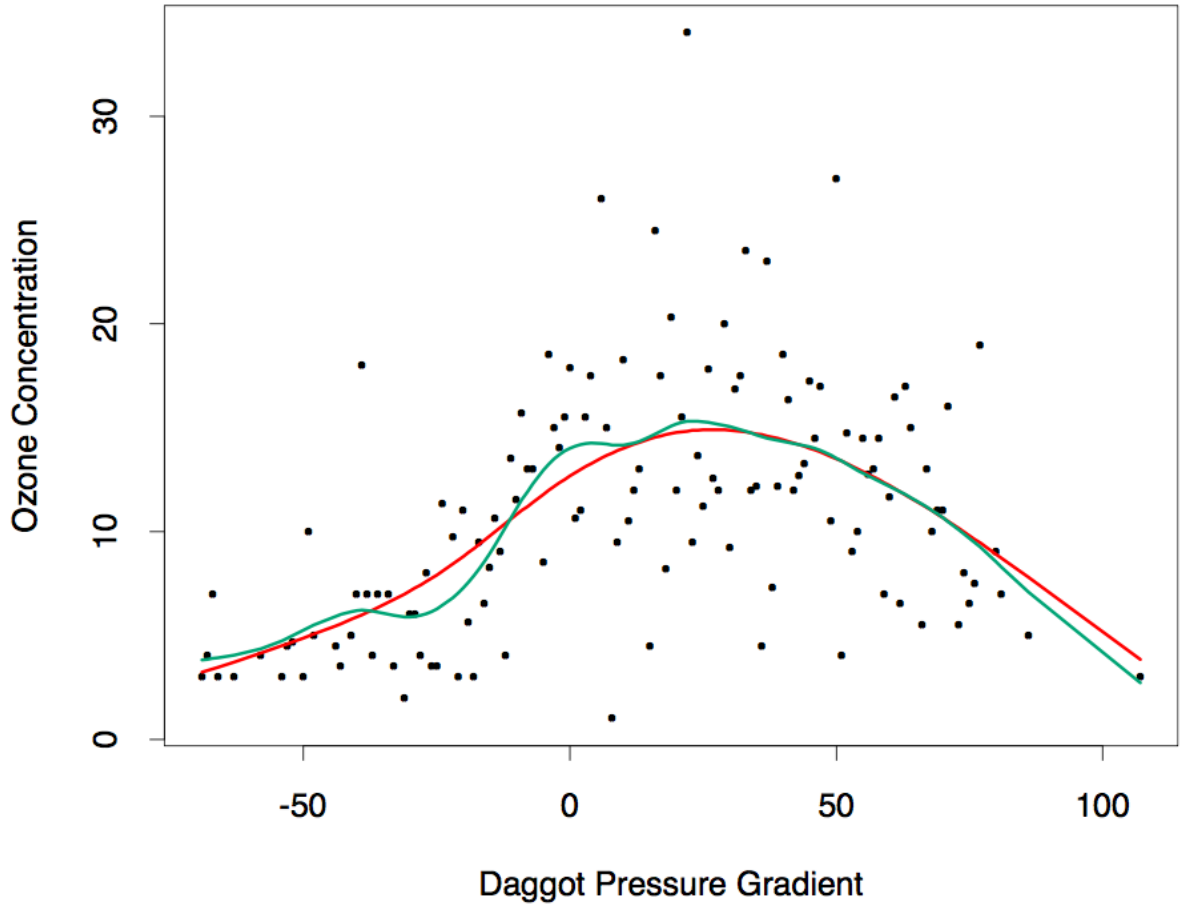
- We try to fit a function g that fits the data as good as possible, but it should avoid overlearning. A reasonable demand is for the function to be "smooth". We use the following optimization function.

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

- 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)



Question: What does this comment refer to

"In other words, the function g(x) that minimizes (7.11) is a natural cubic spline with knots at x1, . . . , xn! However, it is not the same natural cubic spline that one would get if one applied the basis function approach described in Section 7.4.3 with knots at x1, . . . , xn—rather, it is a shrunken version of such a natural cubic spline, where the value of the tuning parameter λ in (7.11) controls the level of shrinkage."

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

Non-parametric methods

- Normal linear regression assumes e.g. normal distribution of errors.
  - Non-parametric covers techniques that do not rely on data belonging to any particular distribution. E.g. the Mann–Whitney U test for the hypothesis two samples are from the same population and is based on ranking your values. The test can be more powerful than a t-test on non-normal distributions .
- Polynomial expansions to fit a complex function still assume a single functional can generalize the predictor-response relationship.
  - Non-parametric methods make no (less) assumptions on the form of the functional

The simplest non-parametric regression

- A prediction for a value in a range is based on a **local weighted average** based on the nearby points.
- The function that defines the weights for the weighted average is dubbed a "kernel", e.g. a Gaussian kernel.
- The result is a smooth function
- package np in R

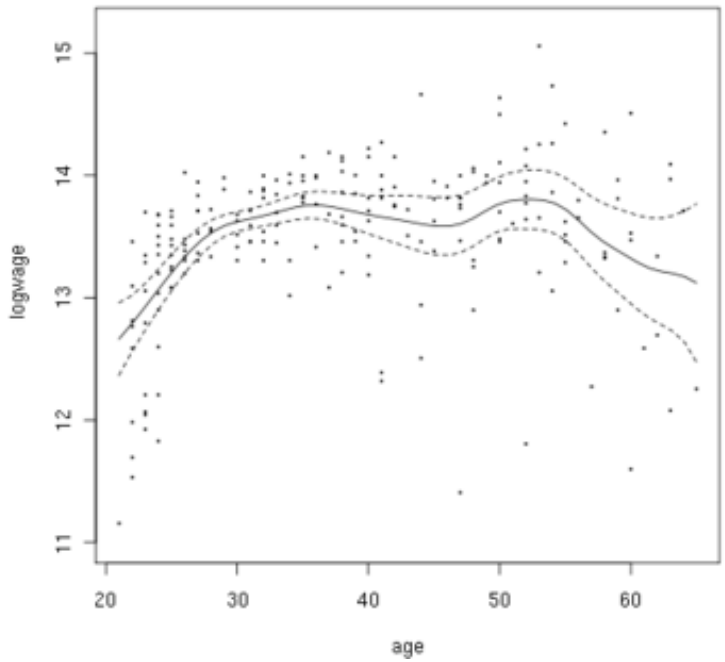
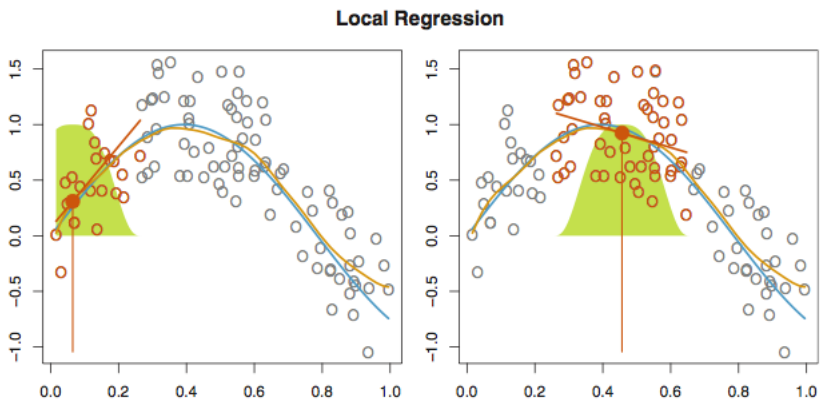


image from Wikipedia (http://en.wikipedia.org/wiki/Kernel_regression)

Local Regression; LOESS

- The predictor and response relationship is modeled with **local linear fits:** for a given x we fit $f(x) = \beta_0 + \beta_1 x$
- A weighted least squares fit is made for these simple linear models. The observations are sampled from around x and are weighted through a specified kernel function. The observations close to the value to be predicted are given most weight.
- LO(W)ESS stands for Locally Weighted Scatterplot Smoothing



Local regression , where the blue curve represents generating function, orange curve corresponds to the local regression estimate f(x). The yellow bell-shape superimposed on the plot indicates weights assigned to each point, decreasing to zero with distance from the target point.

Local Regression; LOESS

Algorithm 7.1 *Local Regression At* $X = x_0$

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 highest weight. All but these k nearest neighbors get weight zero.

3. Fit a *weighted least squares regression* of the y_i on the x_i using the aforementioned weights, by finding $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize

$$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2. \tag{7.14}$$

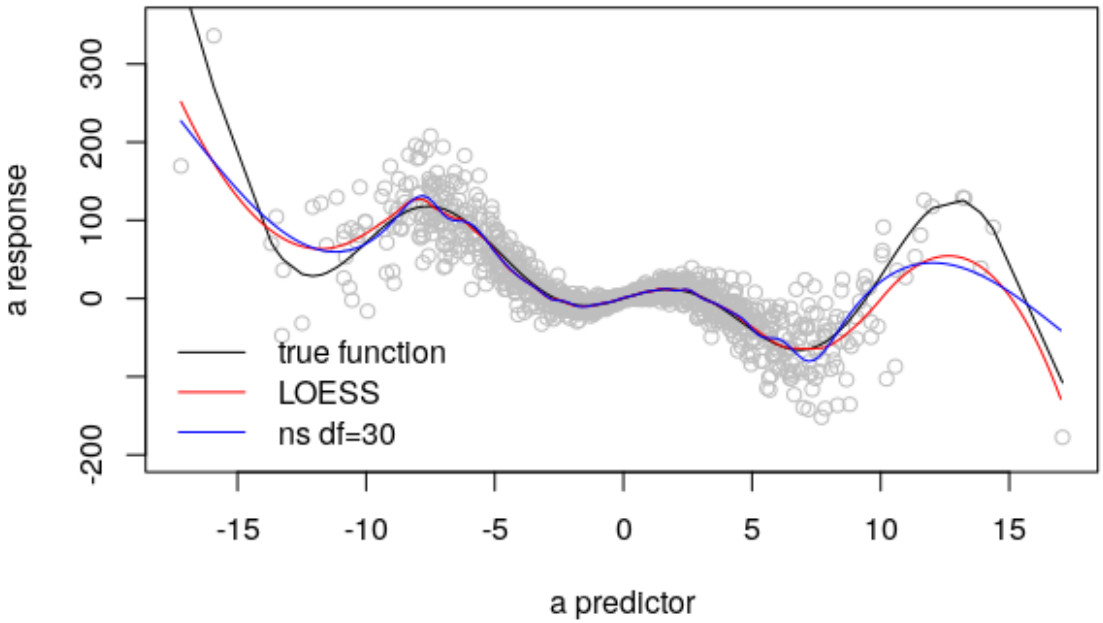
4. The fitted value at x_0 is given by $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$.

Local Regression

- Choices:
  - The weighting function
    - a continuous, bounded, and symmetric real function
    - a running mean is known as the box kernel
    - a (truncated) Gaussian is a natural candidate
  - The weighting function comes with range parameter
    - e.g. the span, the fraction of the dataset considered by the kernel
  - Type of regression function
- Advantages: v. flexible fit
- Disadvantages:
  - Requires dense data to work well
  - No closed functional definition
    - a memory-based procedure

Local Regression vs Splines

- Which works better?



Specifics

- Standard errors can be estimated for every point, however bootstrap estimates are often preferred
- The degrees of freedom used by the smoother can be estimated very similarly to how we did it for the smoothing spline:
  - The vector of estimated values f can be expressed as: $\hat{f} = Sy$, S is a $n \times n$ matrix defined by our smoother and y are our observations.
- The used degrees of freedom by $df = trace(S)$, the sum of the diagonal values of the matrix $df = \sum_{i=1}^{n} \{S\}_{ii}$.

Question

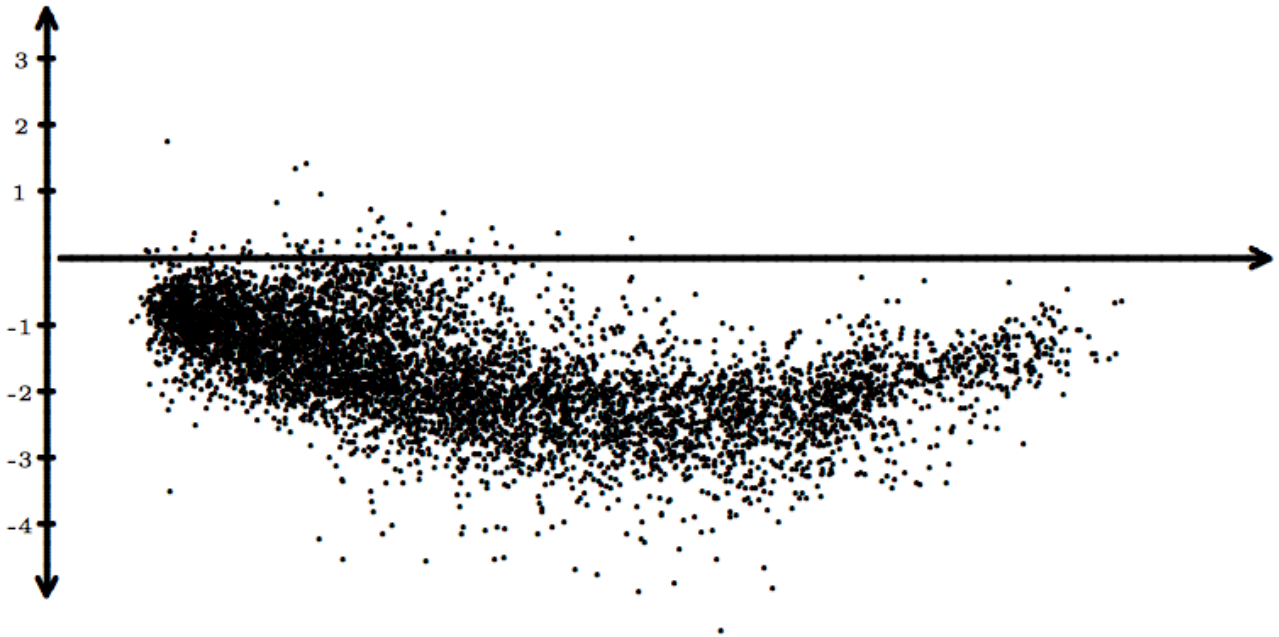
From Page. 281, I don't really understand 'we need all the training data each time we wish to compute a prediction.' Why we need all the training data?

An Application of Non-Linear Models

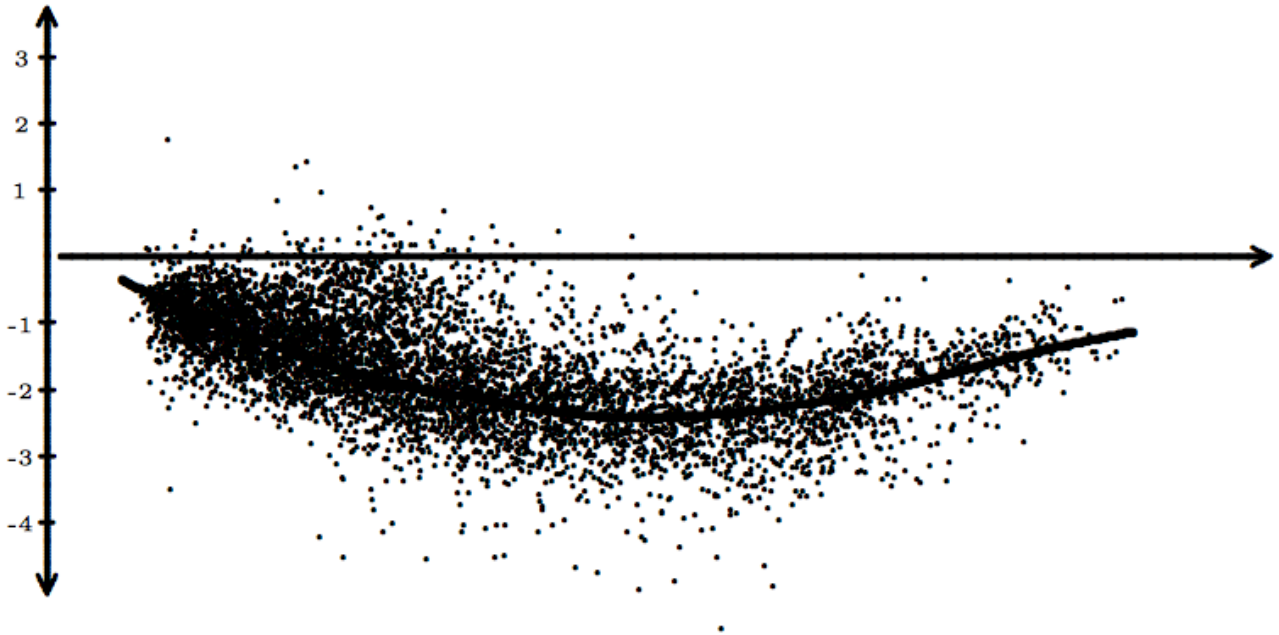
- One important use is to remove Systematic Experimental Bias from data; or calibration.
- An example: Spotted microarrays consist of spotted DNA samples in a regular pattern on a solid surface. Read out of relative abundances of mRNA by hybridization of cDNA tagged with a fluorescent dye. To compare two conditions, two dyes are used: e.g. Cy3 (green) and Cy5 (red).
- We are typically interested in the ratio between the signals as a measure of differential expression between conditions
- However the green dye often has a tendency to be stronger than the red dye. The magnitude of this effect varies from array to array. If we can measure this bias we can correct for it.
- A standard method of displaying microarray data that visualizes the spread between the two channels shows a G(g) as the Cy3 intensity for a gene g, and R(g) is the Cy5 intensity for g, and we plot $M = \log2(G(g)/R(g))$ on the vertical axis, against $A = (\log2(G(g) + \log2(R(g)))/2$ on the horizontal axis

M versus A plot

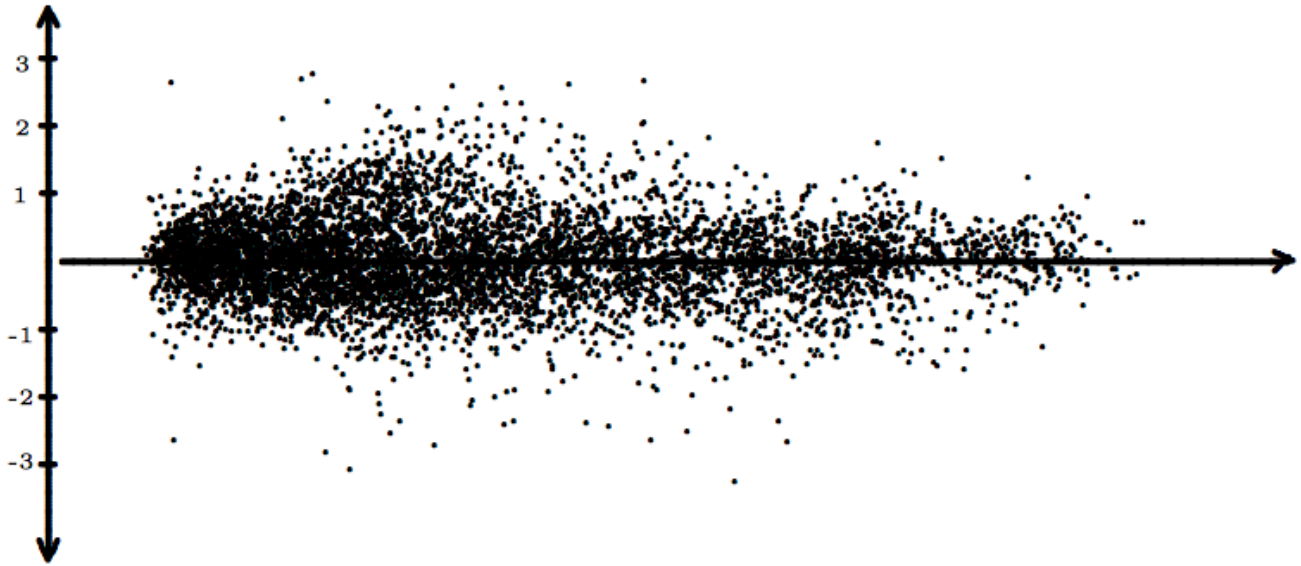
M is log fold (vertical axis), A is abundance (on the horizontal axis)



M versus A LOESS fit



M versus A LOESS fit subtracted



Details

- When one may not assume that most of the genes are unchanged between the two conditions, applying this method may normalize out true biological differences.
- Another issue of normalization involves the spread of the M values across the array, which may depend on the array itself and not on the biology.
- In real experiments there are normally many biases and random effects.

High dimensionality

- Can we fit non-linearly when p is large (and n<p)?

Generalized Additive Models

- Generalized Additive Models (GAMs) extend the Generalized Linear Model so that non-linear responses can be included, maintaining the additive form between components.

$$g(y_i) = \beta_0 + \sum_{j=1}^{p} \beta_j f_j(x_{ij}) + \varepsilon_i$$

becomes

$$g(y_i) = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \varepsilon_i$$

- For natural/regression splines the non-linear function can be represented as a normal set of basis functions and we can use a normal least squares approach and a general linear model!
- Other functionals push to alternative fitting procedures, as the back-fitting procedure (exercise 11)

GAM fitting

A normal lm OLS fit is defined as:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

For a GAM OLS is not defined in general

Backfitting

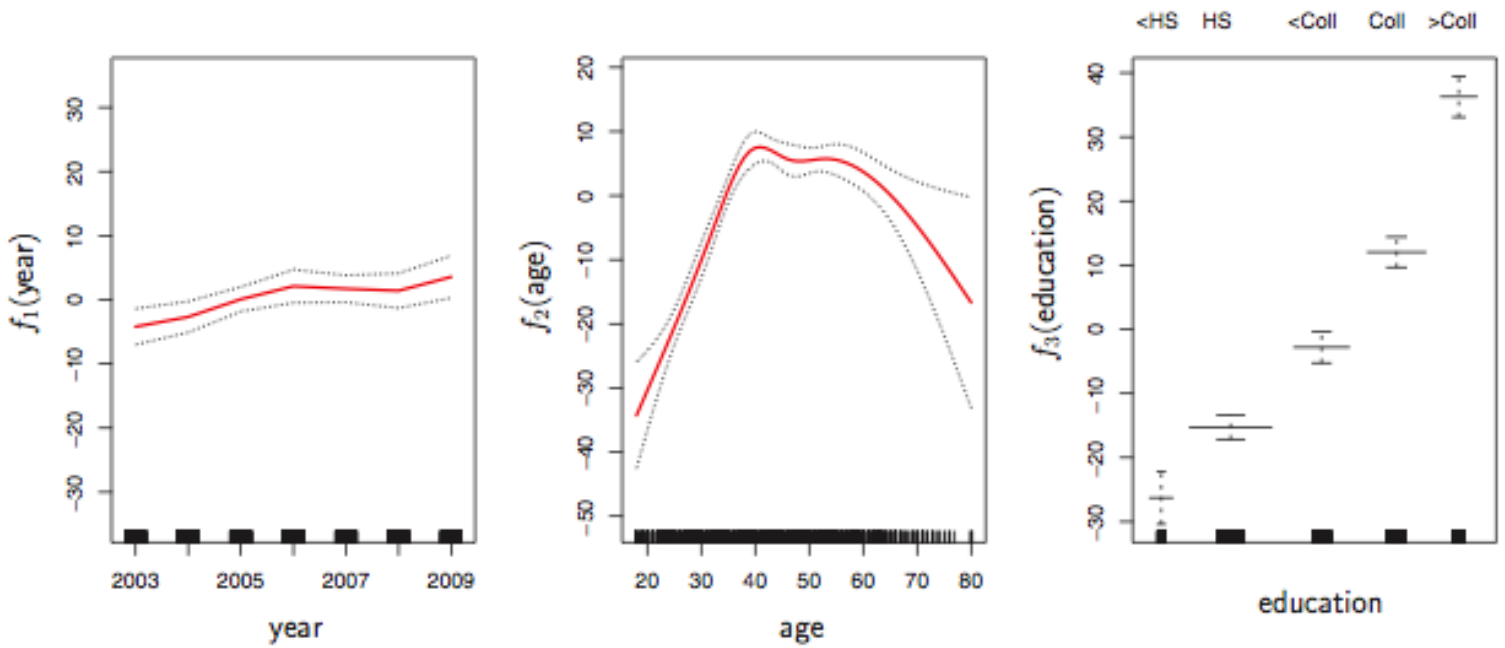
1. Initialize: $\beta_0 = \bar{y}$, $f_j = f_j^0$, $j = 1, \cdots, p$
2. Cycle: $j = 1, \cdots, p, 1, \cdots, p, \cdots$

$$f_j = S_j(y - \beta_0 - \sum_{k \neq j} f_k | x_j)$$

Repeat till changes in f minimal.

Generalized Additive Models

- Why the additive format?



For the Wage data, plots of the relationship between each feature and the response, wage, in the fitted model $wage = \beta_0 + f_1(year) + f_2(age) + f_3(education) + \varepsilon$. Each plot displays the fitted function and point-wise standard errors. The first two functions are natural splines in year and age, with four and five degrees of freedom, respectively. The third function is a step function, fit to the qualitative variable education.

GAM for Classification

A more general notation for part of the GAM formulation is

$$g(E(y)) = \beta_0 + \sum_{j=1}^{p} f_j(x_j)$$

where a link function g connects the predictions to a specified exponential error function distribution (e.g. Poisson, Gaussian, Binomial). Hence GAMs can also be used for classification problems:

$$log(\frac{p(y_i)}{1 - p(y_i)}) = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \varepsilon_i$$

Generalized Additive Models

- The GAM allows flexible fits, with relaxed assumptions, to better represent relationships in the data. (lower bias)
- This comes at some loss of interpretability.
  - Ease of understanding, summarization, communication
  - Parameterized methods give easily interpreted, simple predictions
- Overfitting can be a serious problem; though solutions exist!
  - Control degrees of freedom
  - Cross-validation
  - Compare GAM fits to GLM fits: is the decrease in bias higher than the increase in variance? Are your non-linear models significantly better?
- It is usually preferable to rely on a simple well understood model for predicting future cases, than on a complex model that is difficult to interpret and summarize.
- How about interactions between variables?

Classical comparisons of (G)LMs for model selection

- In the lab they refer to doing ANOVA's to compare linear models.
- Classical model selection approach: The General Linear F-Test. F stands for Fisher.

F-test for linear models

- You compare two linear models: a complete model, also called the unrestricted model, and a reduced model (restricted). In the reduced model one or more of the coefficients in the start model are 0. For example:

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

and a reduced (or *nested*) model with some coefficients *0*:

$$y = \beta_0 + \beta_2 X_2$$

- You want to test that the hypothesis that the removed coefficients are 0: $H_0 : \beta_1 = 0$
- The basis for the comparison is the Residual Sum of Squares of the fits, and an assumption on the normality of the residuals.

F-test definition

- Calculated the $RSS = \sum_i (y_i - \hat{y}_i)^2$ for the complete (c) and reduced (r) models, note the number of used degrees of freedom (df) and the remaining degrees of freedom for the start model ($n - df_c$). Calculate the F-statistic:

$$F = \frac{RSS_r - RSS_c}{df_c - df_r} \Big/ \frac{RSS_c}{n - df_c}$$

- This statistic has an F distribution with parameters ($df_c - df_r$,$n - df_c$)
- Note $RSS_c \leq RSS_r$
- For linear regression, this is equivalent to the ANOVA F-test.
- Can be used to step by step reduce a full model, a kind of Stepwise Backward Selection with hypothesis testing.

GAM evaluation

How to compare models of different complexities:

- ANOVA (if nested)
  - Can compare linear vs non-linear components
    m1=lm(wage $\sim$ ns(year, df $= 5$) + ns(age, df $= 5$))
    m2=lm(wage $\sim$ year + ns(age, df $= 5$))
    anova(m1,m2)
  - GLM vs GAM
    m3=gam(wage$\sim$ s(year, df $= 5$) + ns(age, df $= 5$))
    anova(m3,m2)
- AIC
- Cross-Validation

To do:

Preparation for next week

Read chapter 8 + videos
Send in any question day before class

Exercises

- Lab chapter 7
- Chapter 7, exercise 1, 2, 5, 10 & 11