COM 599200 – Statistical Learning

Lecture 7 – Moving Beyond Linearity

Y.-W. Peter Hong

Extensions to the Linear Model

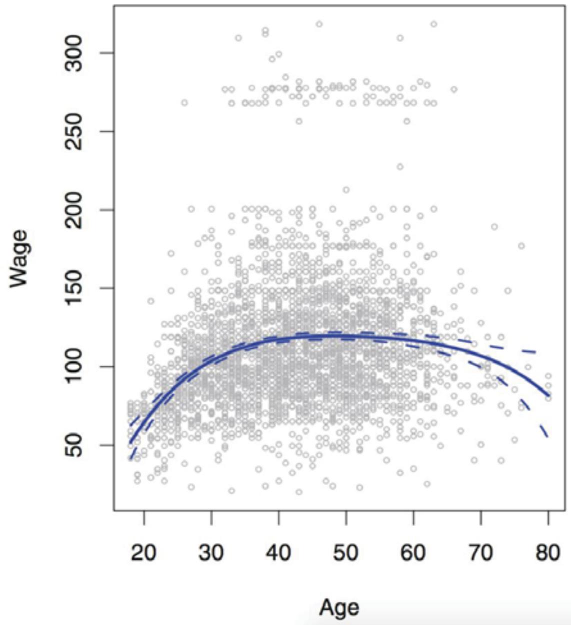
- *Polynomial regression*: Add extra predictors by raising each of the original predictors to a power.
- *Step functions*: Divides the range of a variable into K regions and fit a piecewise constant function in each.
- *Regression splines*: Divides the range of a variable into K regions and fit a polynomial function in each.
- *Smoothing splines*: Similar to regression splines, but minimizes RSS subject to a smoothness penalty.
- *Local regression*: Similar to splines, but regions overlap.
- *Generalized additive models*: Extend the methods above to deal with multiple predictors.

Polynomial Regression

- Extends the linear model $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ to the polynomial model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_d x_i^d + \epsilon_i.$$

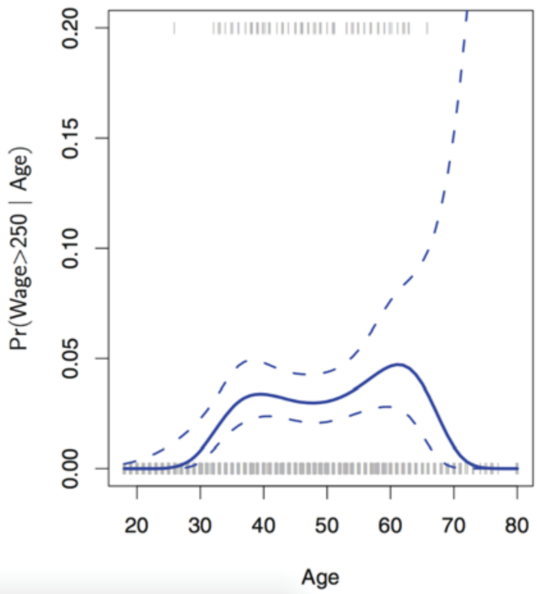
➔ Usually $d \leq 3, 4.$



Polynomial (Logistic) Regression

- Similarly, to classify people into high and low earners groups, we can adopt polynomial logistic regression

$$\Pr(y_i > 250|x_i) = \frac{\exp(\beta_0+\beta_1 x_i+\beta_2 x_i^2+\cdots+\beta_d x_i^d)}{1+\exp(\beta_0+\beta_1 x_i+\beta_2 x_i^2+\cdots+\beta_d x_i^d)}.$$



Step Function (1/2)

- Previous scheme imposes a global structure on $f(X)$.
- Use step functions to break the range of X into bins.
  1. With $c_1, c_2, \ldots, c_K$, construct $K+1$ new variables

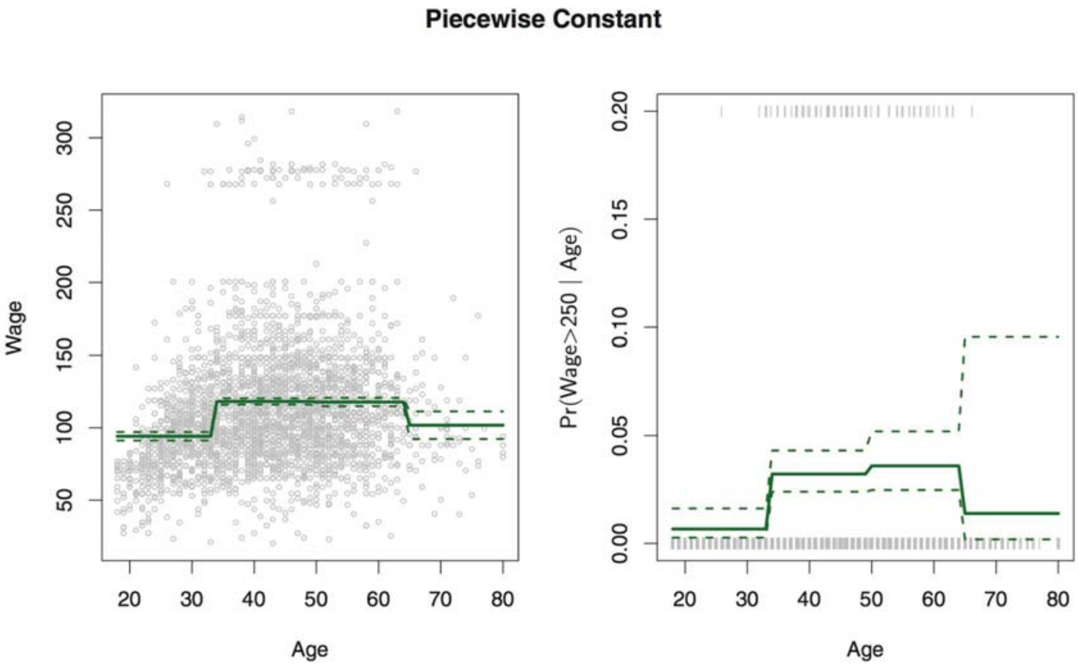
$$\begin{aligned} C_0(X) &= I(X < c_1) \\ C_1(X) &= I(c_1 \leq X < c_2) \\ &\vdots \\ C_{K-1}(X) &= I(c_{K-1} \leq X < c_K) \\ C_K(X) &= I(c_K \leq X) \end{aligned}$$

     where $I(\cdot)$ is an indicator function.
  2. Use least squares to fit the linear model

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \cdots + \beta_K C_K(x_i) + \epsilon_i$$

$$(\text{or } y_i = \beta_0' C_0(x_i) + \beta_1' C_1(x_i) + \beta_2' C_2(x_i) + \cdots + \beta_K' C_K(x_i) + \epsilon_i).$$

Step Function (2/2)



- Similarly, for logistic regression, we have

$$\Pr(y_i > 250|x_i) = \frac{\exp(\beta_0 + \beta_1 C_1(x_i) + \cdots + \beta_K C_K(x_i))}{1 + \exp(\beta_0 + \beta_1 C_1(x_i) + \cdots + \beta_K C_K(x_i))}.$$

Basis Functions

- Polynomial and piecewise-constant regression models are special cases of a basis function approach.
- Fit a linear model of basis functions
  $$y_i = \beta_0 b_0(x_i) + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_K b_K(x_i) + \epsilon_i,$$
  where the basis functions $b_0(\cdot), b_1(\cdot), b_2(\cdot), \ldots, b_K(\cdot)$ are fixed and known.
- E.g., for polynomial regression,
  $$b_j(x_i) = x_i^j,$$
  for piecewise-constant regression,
  $$b_j(x_i) = I(c_j \leq x_i < c_{j+1}).$$

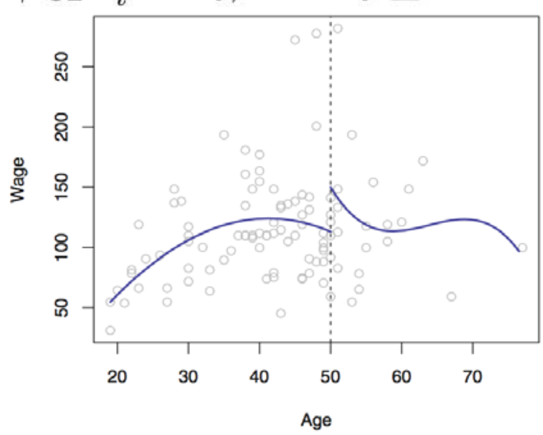
➔ Others: wavelets, Fourier series, regression splines.

Piecewise Polynomial Regression

- Piecewise polynomial regression fits separate low-degree polynomials over different regions of X.
- E.g., by dividing the range of X into regions $\{X < c\}$ and $\{X \geq c\}$, we get a piecewise cubic polynomial
  $$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \epsilon_i, & \text{if } x_i < c \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \epsilon_i, & \text{if } x_i \geq c. \end{cases}$$

  ➔ a single ***knot*** at a point c

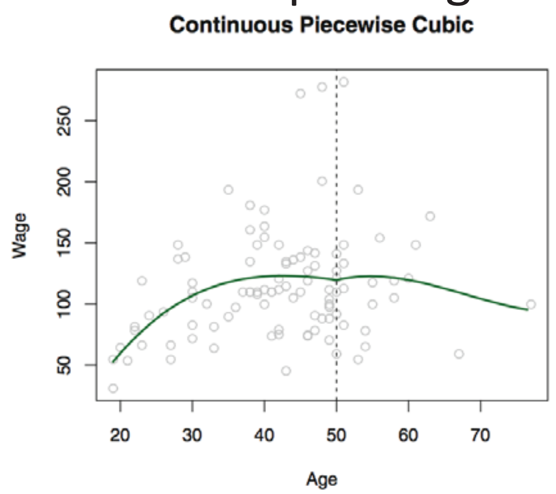
  ➔ 8 degrees of freedom

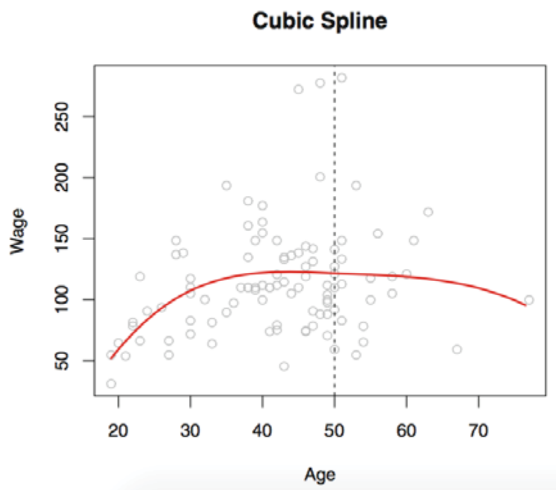


- With K knots, we need to fit to $K+1$ different polynomial models.

Regression Splines

- Previous curve has discontinuity at the knot. ➔ Odd!
- In general, a degree-*d* spline fits a piecewise degree-*d* polynomial to each region, but with continuity in derivatives up to degree *d-1* at each knot.





➔ Each constraint reduces 1 degree of freedom.

The Spline Basis Representation

- **Question:** How can we fit a degree-*d* spline?
- Notice that a cubic spline with K knots at $\xi_1, \ldots, \xi_K$ can be represented as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i,$$

where $b_1(x_i) = x_i,\ b_2(x_i) = x_i^2,\ b_3(x_i) = x_i^3$, and

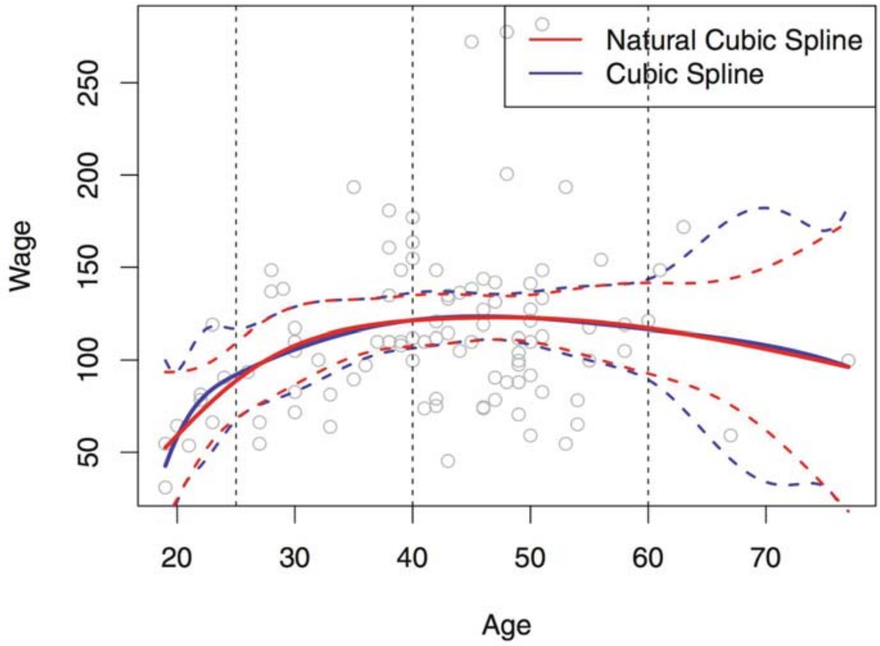
$$b_{3+k}(x_i) = (x - \xi_k)_+^3 \triangleq \begin{cases} (x - \xi_k)^3, & \text{if } x_i > \xi_k, \\ 0, & \text{otherwise,} \end{cases}$$

for $k = 1, \ldots, K$ are *truncated power basis* functions.

➔ $K + 4$ degrees of freedom

- Extends to general degree-*d* splines, but in practice *d* seldom goes beyond 3.

Cubic vs Natural Cubic Splines



- A natural spline adds additional linear constraints to the functions in the boundary regions.

Natural Cubic Splines (1/2)

- A natural cubic spline is similar to the cubic spline but requires the function to be linear beyond the boundary knots.
- Recall the truncated power series representation for cubic splines

$$f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \sum_{k=1}^{K} \beta_{k+3}(X - \xi_k)_+^3.$$

- The linear boundary conditions for natural cubic splines yield $\beta_2 = \beta_3 = 0$ and

$$\sum_{k=1}^{K} \beta_{k+3} = 0, \quad \sum_{k=1}^{K} \xi_k \beta_{k+3} = 0.$$

Natural Cubic Splines (2/2)

- Hence, a natural cubic spline with K knots is represented by the K basis functions

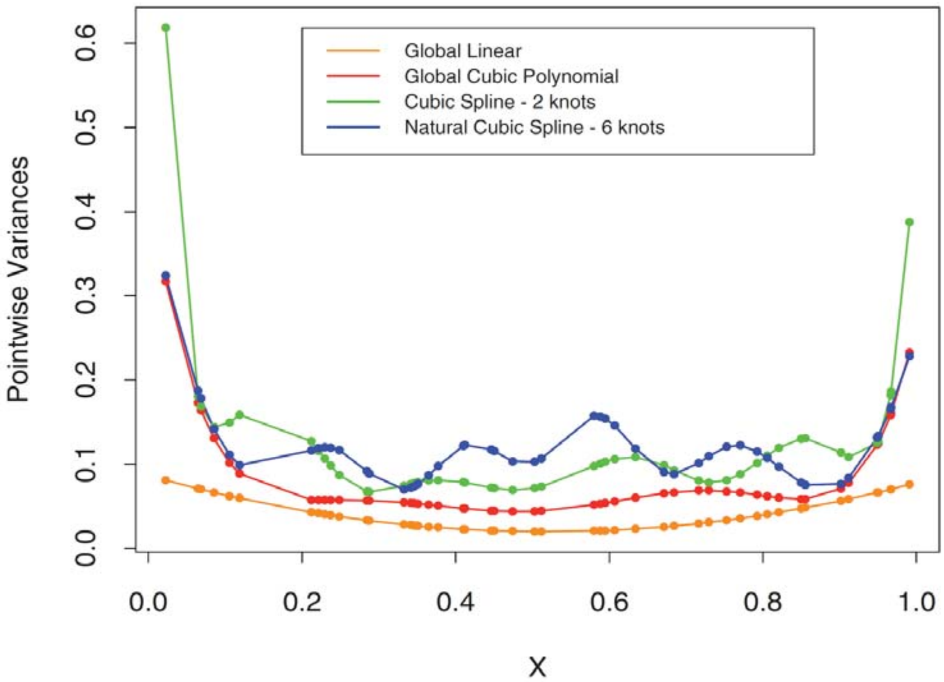
$$b_0(X) = 1,\ b_1(X) = X,$$
$$b_{k+1}(X) = d_k(X) - d_{K-1}(X),$$

for $k = 1, \ldots, K-2$, where

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

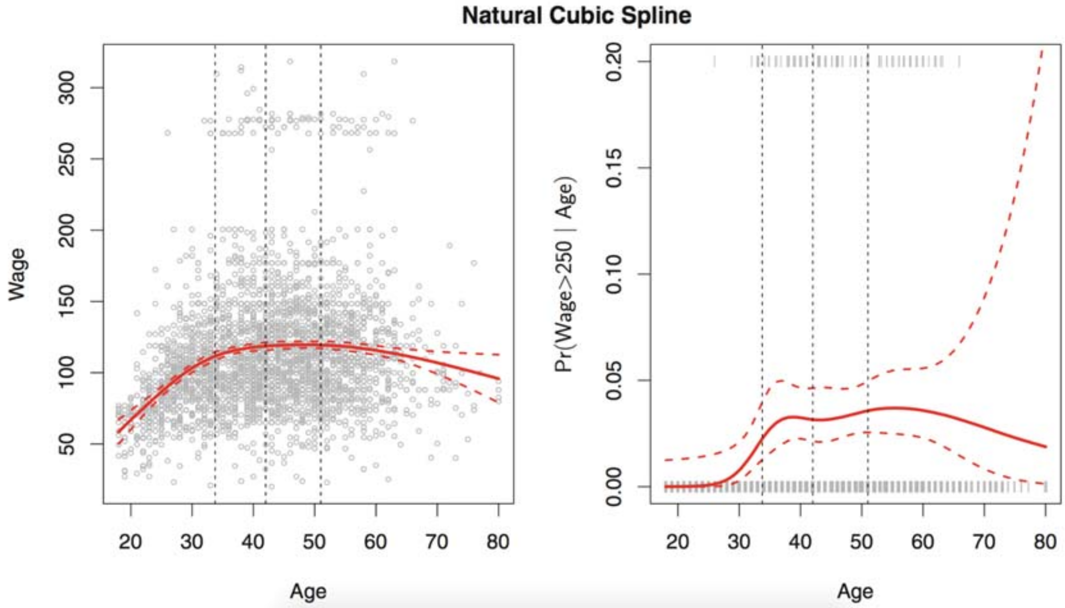
Example

- Simulated data with 50 data points with $X \sim \mathcal{U}([0,1])$ and Gaussian noise with constant variance.



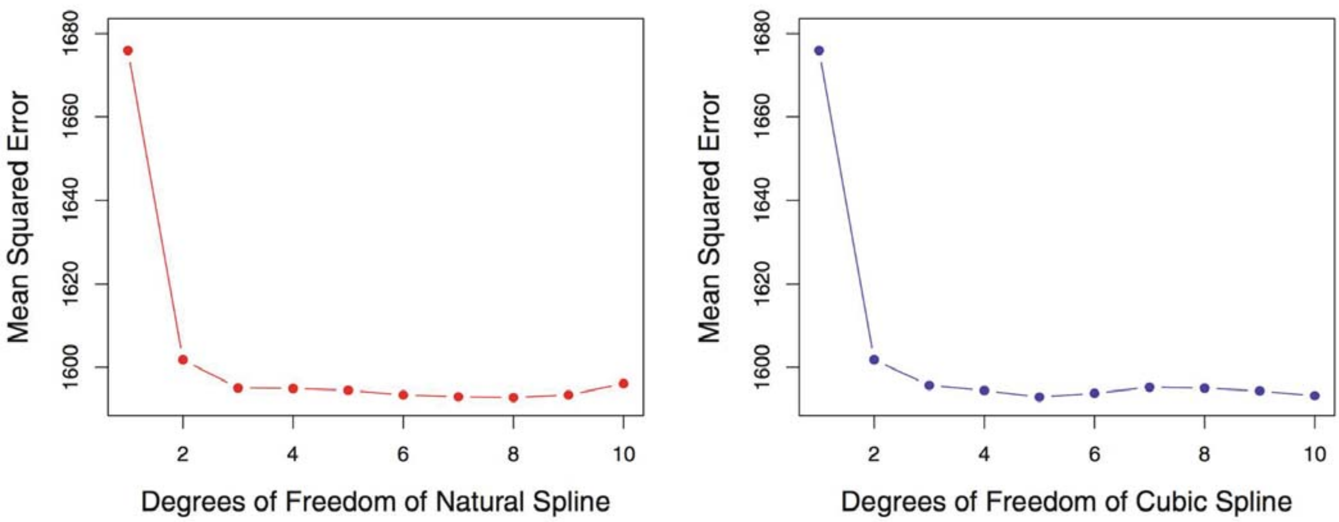
Choosing the Locations of Knots

- **Question:** Where should we place the knots?
  - Place more knots in places that may vary more.
  - In practice, it is common to place them uniformly.



Choosing the Number of Knots

- **Question:** How many knots should we use?
  - By cross-validation.



Smoothing Splines

- **Goal:** Find function g that minimizes the RSS

$$\sum_{i=1}^{n} \left(y_i - g(x_i)\right)^2$$

while being smooth.

- A **smoothing spline** is a function g that minimizes

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

where $\lambda \geq 0$ is a tuning parameter.

➔ $|g''(t)|$ indicates how "wiggly" $g(t)$ is around t.

✓ When $\lambda = 0$, g will exactly interpolate all data points.

✓ When $\lambda \to \infty$, g is a straight line.

Smoothing Splines as Natural Cubic Splines

- The solution g to the smoothing spline optimization problem is (i) a piecewise cubic polynomial with knots at the points $x_1, \ldots, x_n$, (ii) continuous in first and second derivatives at each knot, and (iii) linear in the region outside of the extreme knots.

  ➔ It is a natural cubic spline, but fitted differently!

- Specifically, for a natural cubic spline, we can write

$$g(X) = \sum_{j=0}^{K-1} \beta_j b_j(X)$$

where $b_0(X) = 1,\ b_1(X) = X,$ and
$b_{k+1}(X) = d_k(X) - d_{K-1}(X),$ for $k = 1, \ldots, K-2.$

Fitting a Smoothing Spline

- Then, the smoothing spline problem reduces to

$$\mathrm{RSS}(\beta, \lambda) = (\mathbf{y} - \mathbf{B}\beta)^T(\mathbf{y} - \mathbf{B}\beta) + \lambda\beta^T\mathbf{\Omega}\beta,$$

where $\{\mathbf{B}\}_{ij} = b_{j-1}(x_i)$ and $\{\mathbf{\Omega}\}_{jk} = \int b''_{j-1}(t)b''_{k-1}(t)dt$.

➔The solution is $\hat{\beta} = (\mathbf{B}^T\mathbf{B} + \lambda\mathbf{\Omega})^{-1}\mathbf{B}^T\mathbf{y}$ and the fitted smoothing spline is

$$\hat{g}_\lambda(X) = \sum_{j=0}^{K-1} \hat{\beta}_j b_j(X).$$

- The fitted values at the training data points are

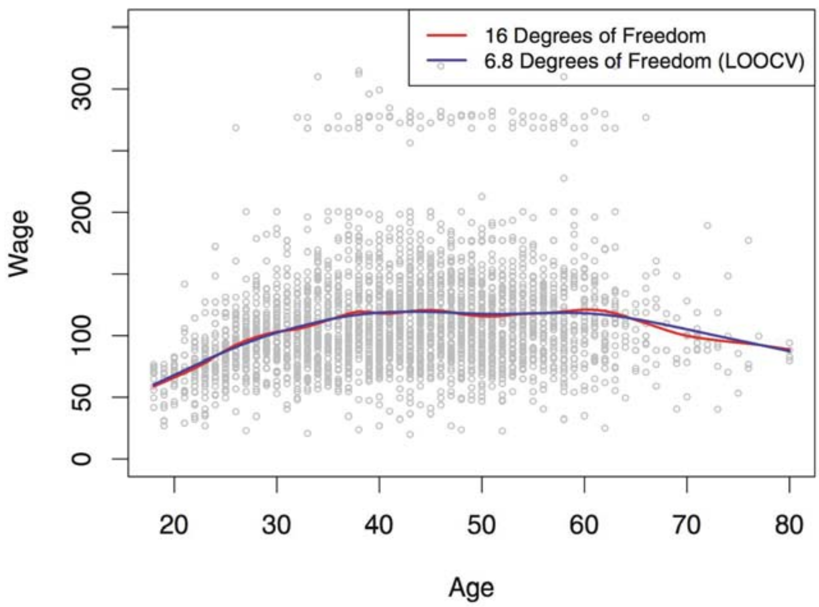
$$\hat{\mathbf{y}} = \mathbf{B}(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{\Omega})^{-1}\mathbf{B}^T\mathbf{y}\big(= \mathbf{S}_\lambda\mathbf{y}\big).$$

- The effective degrees of freedom is $\mathrm{df}_\lambda = \mathrm{tr}(\mathbf{S}_\lambda)$.

Choosing the Smoothing Parameter

- Use CV! E.g., LOOCV yields estimated test MSE

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



Local Regression

- **Local regression** at a point x_0 is obtained by taking the weighted least squares fit using nearby observations, i.e.,

$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$$

where $\hat{\beta}_0$ and $\hat{\beta}_1$ are chosen to minimize

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

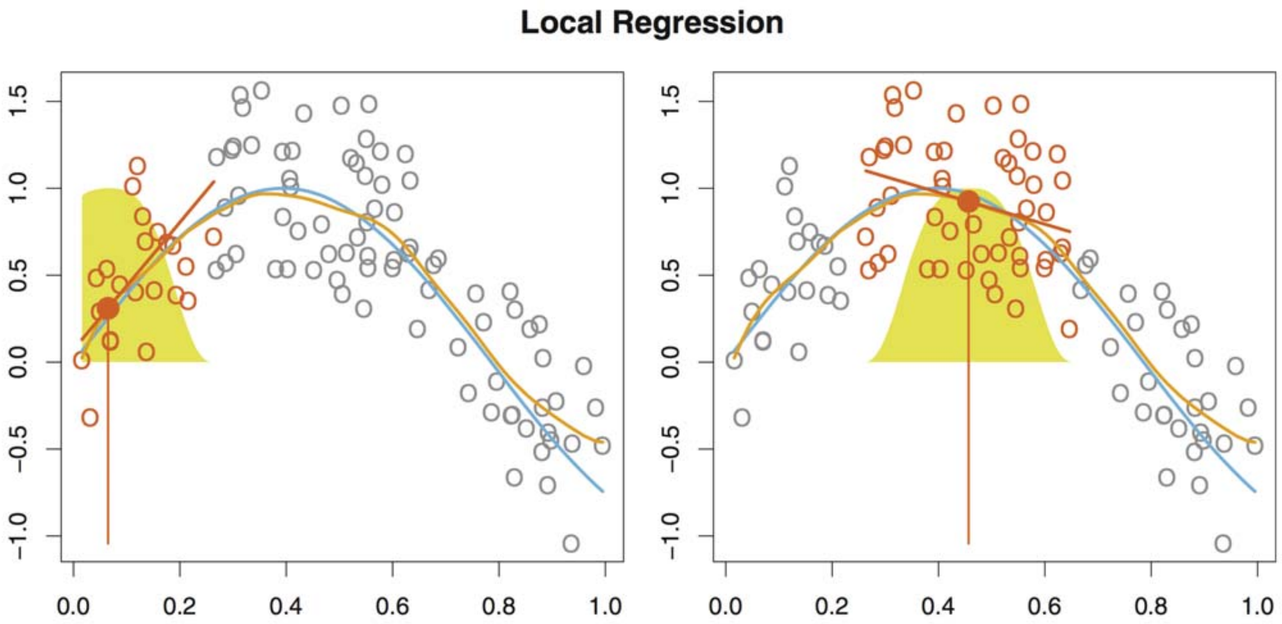
- Here, $K_{i0} = K(x_i, x_0)$ is nonzero only for the k closest points, and $K_{i0} \geq K_{j0}$ if x_i is closer to x_0 than x_j.

➔Similar to k-nearest neighbor approach.

➔The choice of k is important.

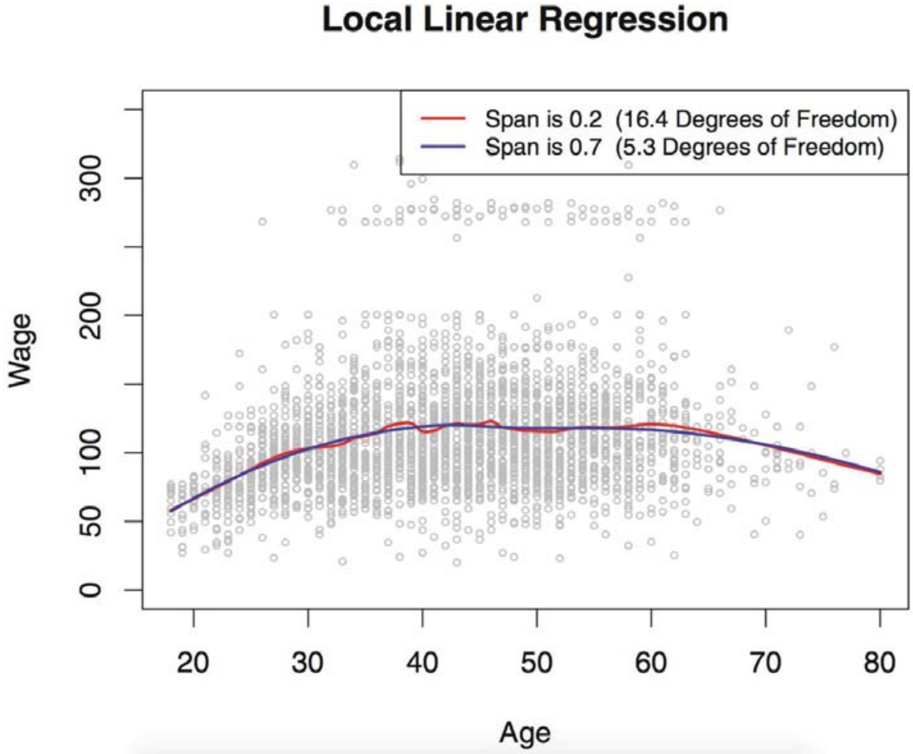
Example

- Simulated dataset (blue line is true $f(X)$)



Example

- Wage dataset for $k/n = 0.2$ and 0.7.



Generalized Additive Models - Regression

- **General additive models (GAMs)** uses nonlinear fitted models for each of the variables as building block for fitting an additive model.
- For regression, GAM extends the standard linear regression model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \epsilon_i$$

to the following

$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i$$

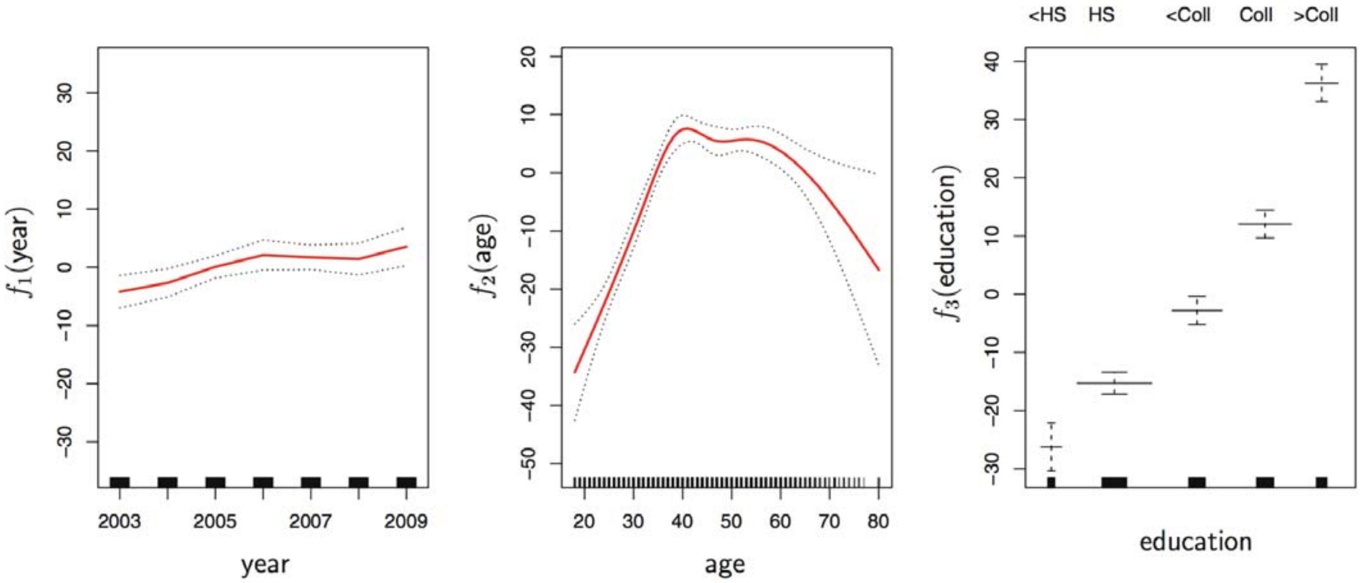
where $f_j(x_{ij})$ is *a (smooth) nonlinear function* used to replace the linear component $\beta_j x_{ij}$.

(E.g., natural splines, smoothing splines … etc)

Example: GAMs with Natural Splines

$$\text{wage} = \beta_0 + f_1(\text{year}) + f_2(\text{age}) + f_3(\text{education}) + \epsilon$$

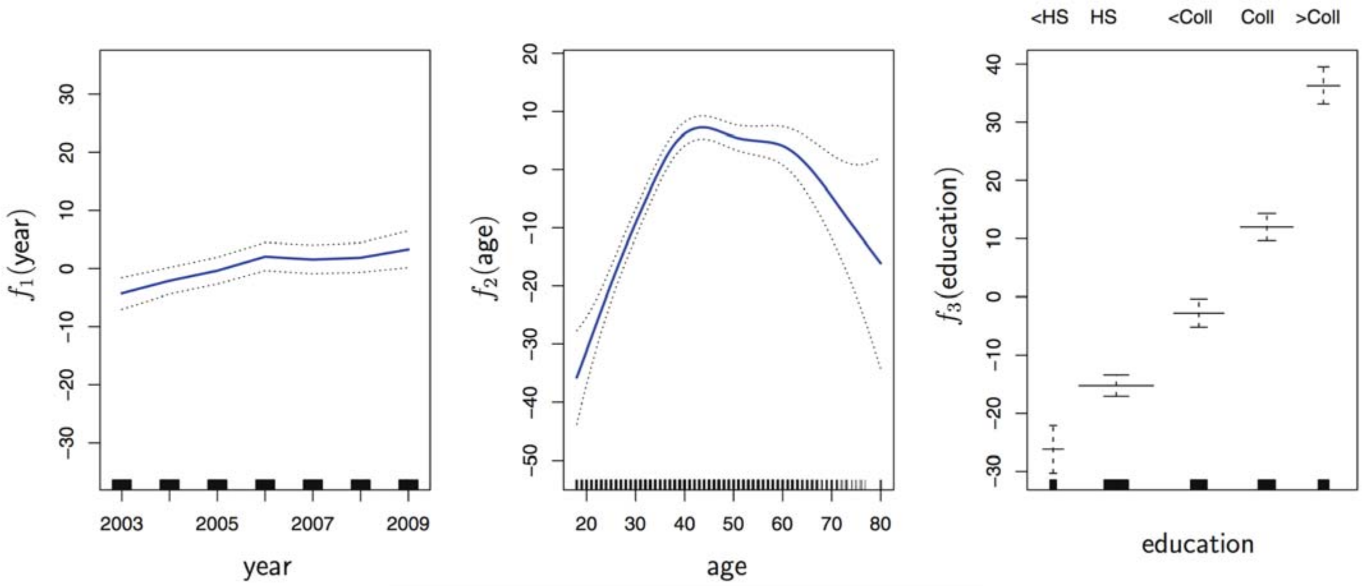
- f_1, f_2: natural splines; f_3: separate constants



➔ Regression onto spline basis and dummy variables.

Example: GAMs with Smoothing Splines

- f_1, f_2: *smoothing* splines; f_3: separate constants



Backfitting Algorithm

1. Initialize: $\hat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n} y_i,\ \hat{f}_j \equiv 0, \forall i, j.$
2. Iterate:

   For $j = 1, \ldots, p$,

$$\hat{f}_j \leftarrow \mathcal{S}_j \left[\left\{ y_i - \hat{\beta}_0 - \sum_{k \neq j} \hat{f}_k(x_{ik}) \right\}_{i=1}^{n} \right]$$

$$\hat{f}_j \leftarrow \hat{f}_j - \frac{1}{n} \sum_{i=1}^{n} \hat{f}_j(x_{ij})$$

   until the functions $\hat{f}_j$ change less than a prespecified threshold.

➔ Here, $\mathcal{S}_j$ is the regression operator of j-th variable.

Pros and Cons of GAMs

- **Pros:**
  - Nonlinear relationships can be automatically identified by fitting nonlinear f_j to each X_j.
  - Nonlinear fits may yield more accurate predictions.
  - Additive models allow us to examine the effect of each X_j on Y individually.
  - The smoothness of f_j can be summarized via degrees of freedom.
- **Cons:**
  - The additive restriction misses out on many interactive terms (but can be addressed by adding interaction functions, e.g., $f_{jk}(X_j, X_k)$).

Generalized Additive Models – Classification

- Suppose that $Y \in \{0, 1\}$ and let $p(X) = \Pr(Y = 1|X)$.
- GAMs extend the logistic regression model

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

  to the following

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + f_1(X_1) + f_2(X_2) + \cdots + f_p(X_p).$$

- Use backfitting procedure in conjunction with a likelihood maximizer. (E.g., adopt iteratively reweighted least squares (IRLS) algorithm where weighted least squares is solved with backfitting.)