Statistical Learning

https://github.com/ggorr/Machine-Learning/tree/master/ISLR

7 Moving Beyond Linearity

- 7.1 Polynomial Regression
- 7.2 Step Functions
- 7.3 Basis Functions
- 7.4 Regression Splines
- 7.5 Smoothing Splines
- 7.6 Local Regression
- 7.7 Generalized Additive Models
- 7.8 Lab: Non-linear Modeling
- 7.9 Exercises

7.1 Polynomial Regression

- Polynomial regression of degree d

$$y_i = \beta_0 + \beta_1 x_i + \cdots + \beta_d x_i^d + \epsilon$$

Example

- Data

x_i	1.1	1.9	3.2	4.1	4.9
y_i	0.8	0.9	1.5	4.8	3.5

- predictors x, x^2, …, x^d
- solution $\beta = (X^T X)^{-1} X^T y$ where

$$X = \begin{bmatrix} 1 & 1.1 & \cdots & 1.1^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 4.9 & \cdots & 4.9^d \end{bmatrix}, \; y = \begin{bmatrix} 0.8 \\ \vdots \\ 3.5 \end{bmatrix}$$

Example

- Probability from data

x_i	1.1	1.9	3.2	4.1	4.9
$\Pr(y_i > 2 \mid x_i)$	0	0	0	1	1

- Equation

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

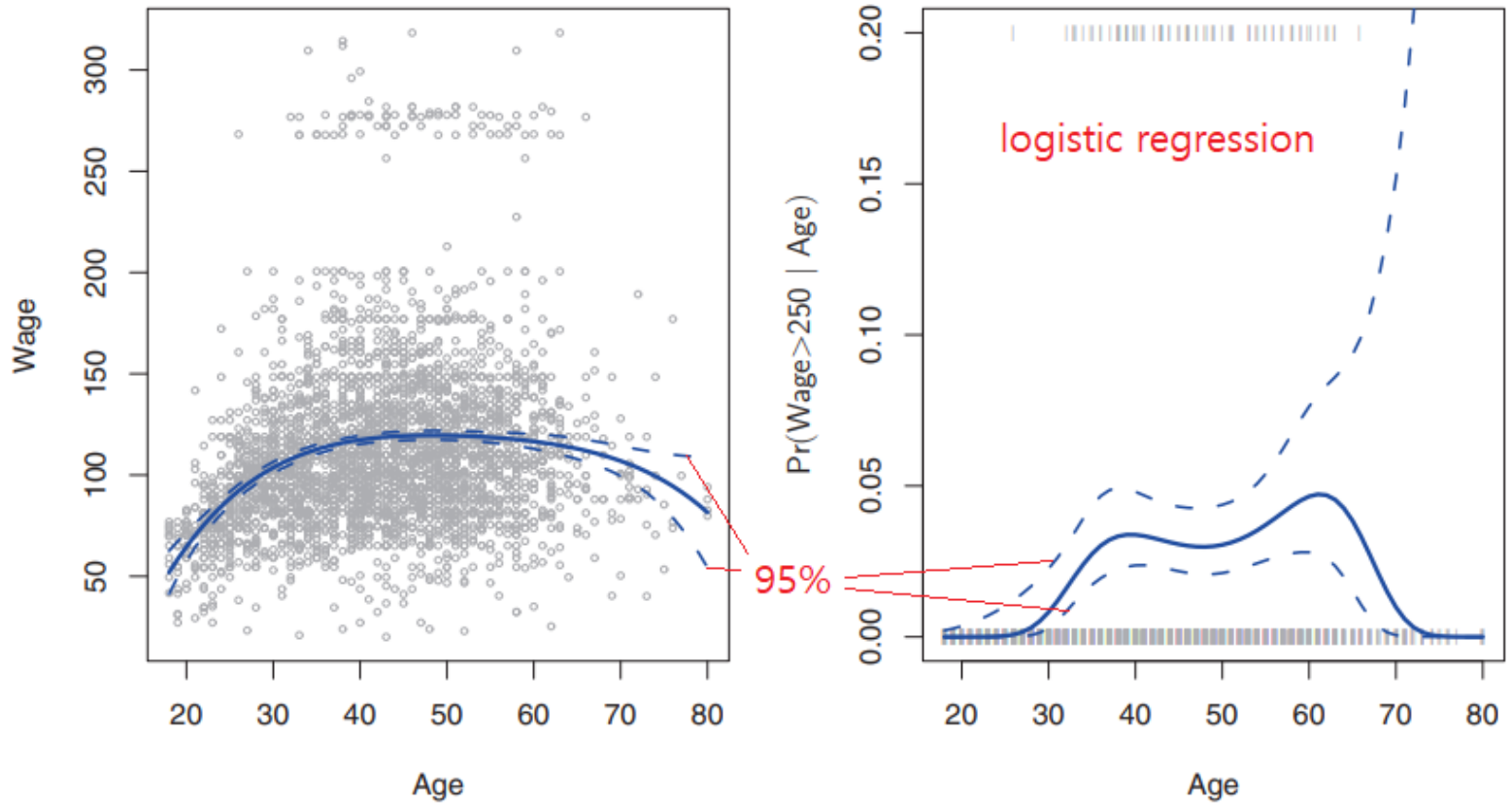
$$\beta_0 + \beta_1 x_i + \cdots + \beta_d x_i^d = \log \frac{p(x_i)}{1 - p(x_i)}$$

where $p(x_i) = \Pr(y_i > 2|x_i)$

- One may find the maximum likelihood solution in Chapter 4

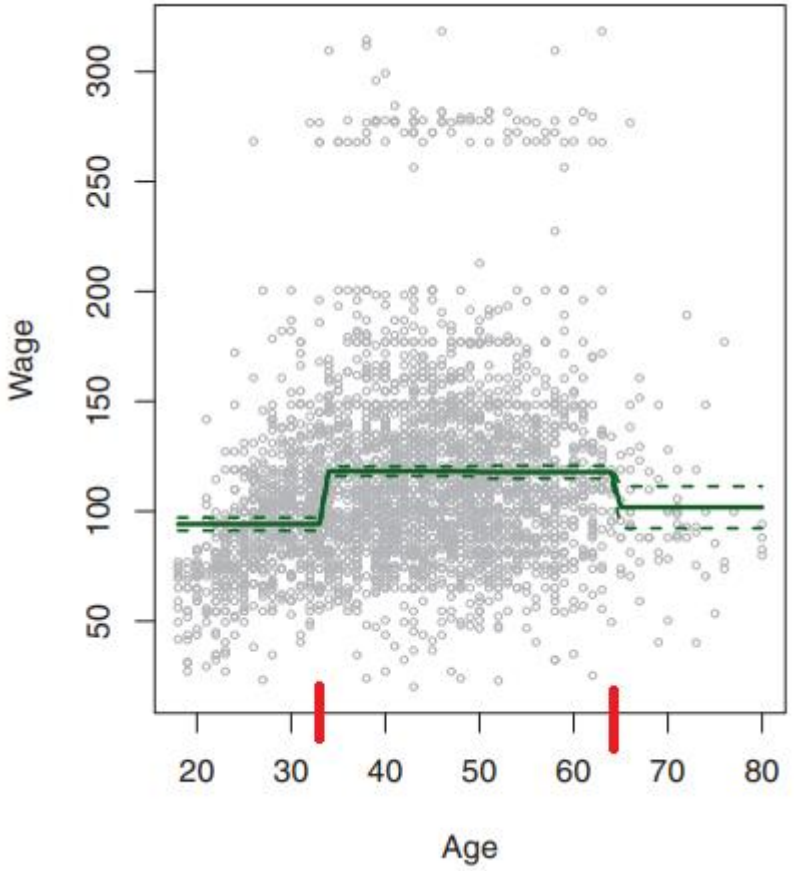
Example

- Polynomial regression of degree 4



7.2 Step Functions

- Divide and average



- Cutpoints

$$c_1, \ldots, c_K \text{ with } c_1 < \cdots < c_K$$

- $K + 1$ intervals

$$(-\infty, c_1), [c_1, c_2), \ldots, [c_{K-1}, c_K), [c_K, \infty)$$

- 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 is the indicator function i.e.

$$I(\text{true}) = 1 \text{ and } I(\text{false}) = 0$$

Estimation

- Use least square to fit the model

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

- Note

  only one $C_i(X) = 1$

  hence $C_0(X) + \cdots + C_K(X) = 1$

- Matrix X is sparse

7.3 Basis Functions

- New variables

$$b_1(X), \ldots, b_K(X)$$

- Model

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \cdots + \beta_K b_K(x_i) + \epsilon_i$$

- Example
  - Polynomial regression

$$b_i(X) = X^i$$

7.4 Regression Splines

- 7.4.1 Piecewise Polynomials
- 7.4.2 Constraints and Splines
- 7.4.3 The Spline Basis Representation
- 7.4.4 Choosing the Number and Locations of the Knots
- 7.4.5 Comparison to Polynomial Regression

7.4.1 Piecewise Polynomials

- Knots(=cut points)

$$c_1, \ldots, c_K \text{ with } c_1 < \cdots < c_K$$

- $K + 1$ intervals

$$(-\infty, c_1), [c_1, c_2), \ldots, [c_{K-1}, c_K), [c_K, \infty)$$

- Polynomial regression on each interval

Examples

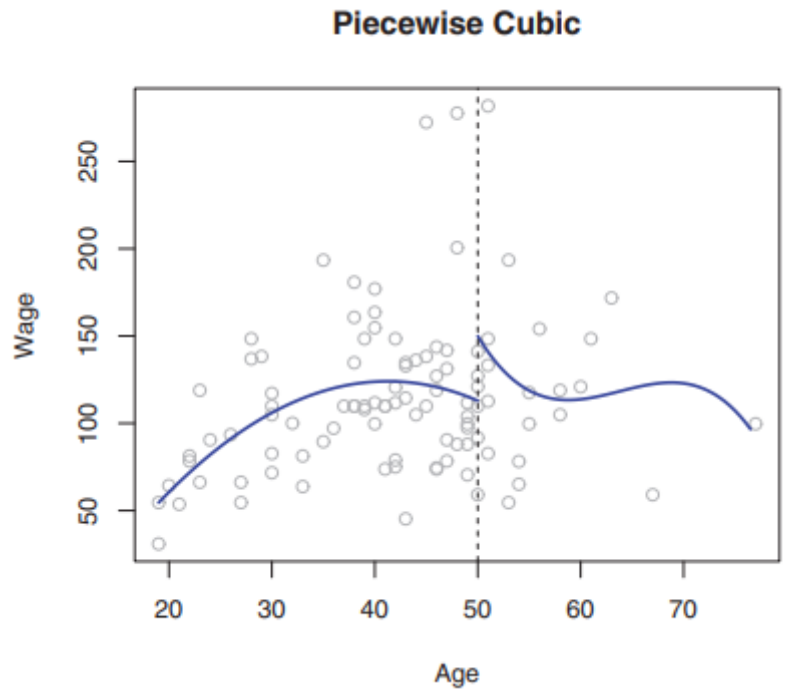
- Example:

  single knot c

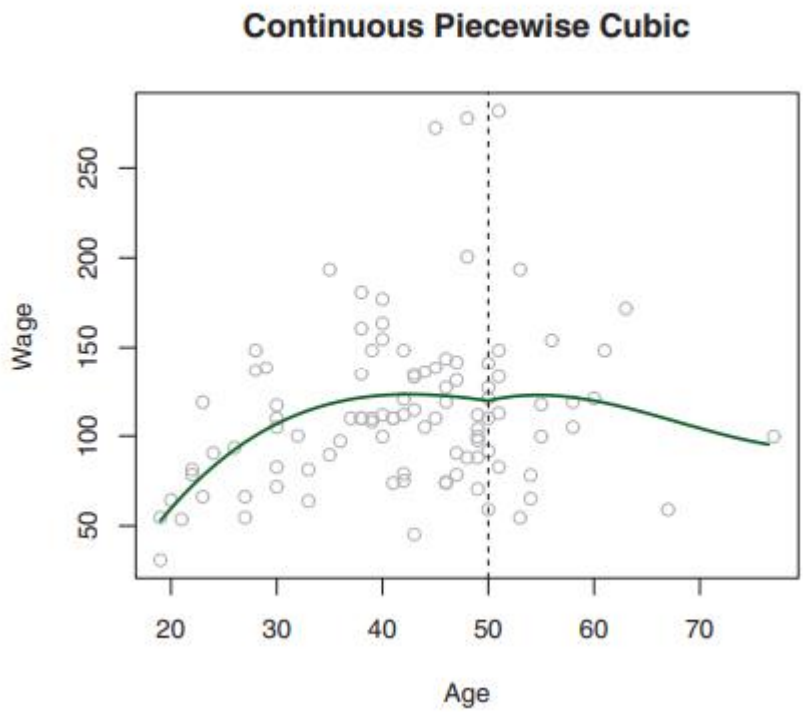
  $$y_i = \beta_{01} + \beta_{11} x_i + \cdots + \beta_{d1} x_i^d + \epsilon \text{ if } x_i < c$$
  $$y_i = \beta_{02} + \beta_{12} x_i + \cdots + \beta_{d2} x_i^d + \epsilon \text{ if } x_i \geq c$$

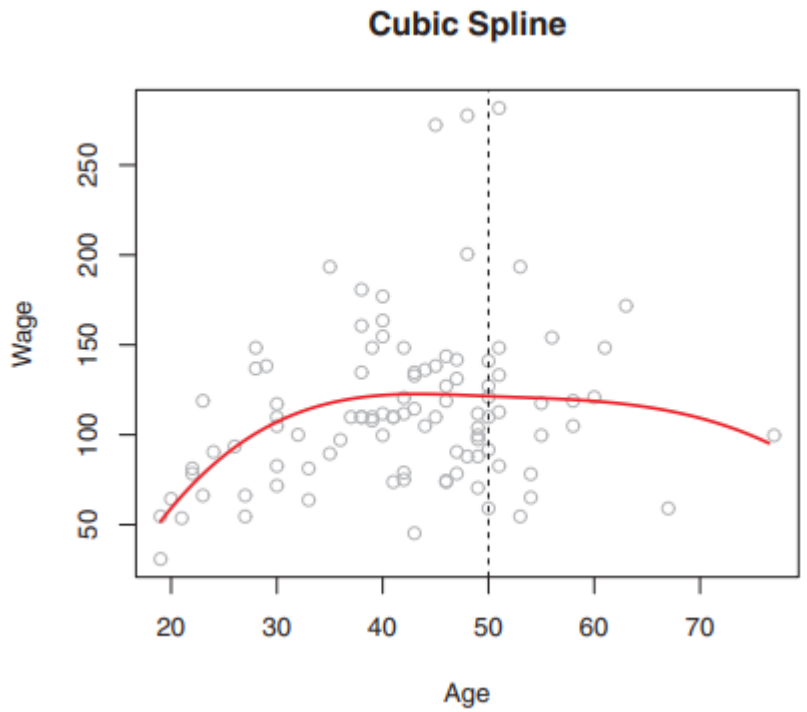
- Example

  A step function is a piecewise polynomial with degree 0


Piecewise Cubic
Wage
50
100
150
200
250
20
30
40
50
60
70
Age

7.4.2 Constraints and Splines





Degree Of Freedom

- Degree of freedom = # of variables

- Example:

  Two cubic polynomials with single knot c

  $$y_i = \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon \text{ if } x_i < c$$

  $$y_i = \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon \text{ if } x_i \geq c$$

  Degree of freedom $= 8$

- Example

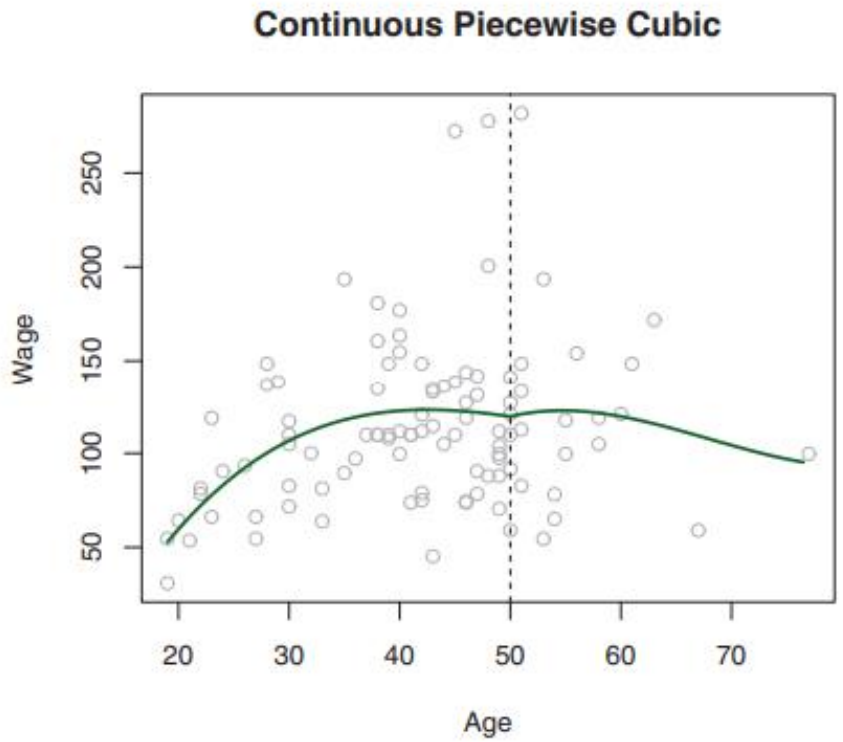
  Two cubic polynomials with single knot c

  $$y_i = \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon \text{ if } x_i < c$$
  $$y_i = \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon \text{ if } x_i \geq c$$

  Constraint: continuous

  $$\beta_{01} + \beta_{11}c + \beta_{21}c^2 + \beta_{31}c^3 = \beta_{02} + \beta_{12}c + \beta_{22}c^2 + \beta_{32}c^3$$

  Degree of freedom $= 8 - 1 = 7$



- Example

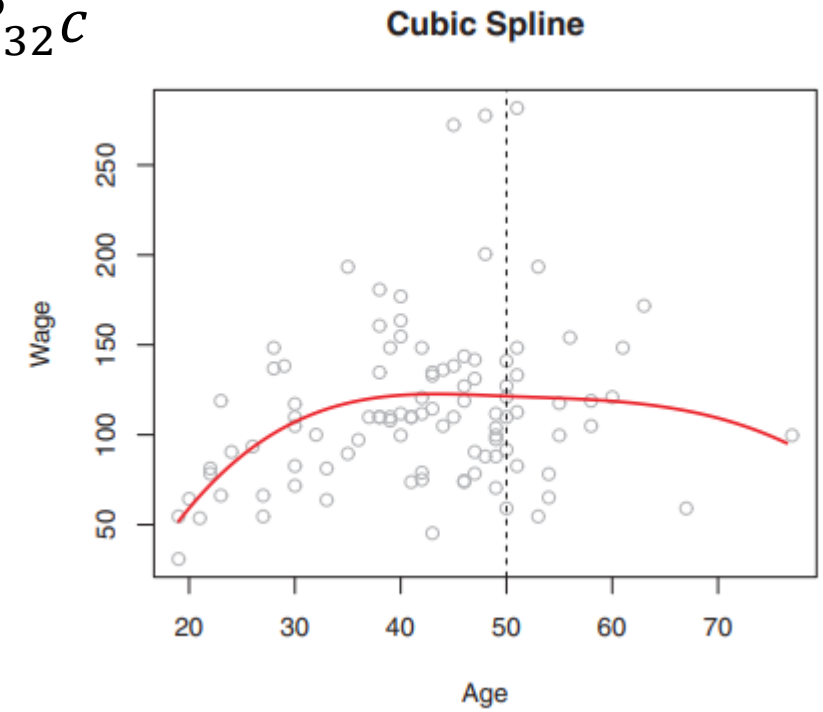
Two cubic polynomials with single knot c

$$y_i = \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon \text{ if } x_i < c$$
$$y_i = \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon \text{ if } x_i \geq c$$

Constraint: smooth

$$\beta_{01} + \beta_{11}c + \beta_{21}c^2 + \beta_{31}c^3 = \beta_{02} + \beta_{12}c + \beta_{22}c^2 + \beta_{32}c^3$$
$$\beta_{11} + 2\beta_{21}c + 3\beta_{31}c^2 = \beta_{12} + 2\beta_{22}c + 3\beta_{32}c^2$$
$$2\beta_{21} + 6\beta_{31}c = 2\beta_{22} + 6\beta_{32}c$$

Degree of freedom $= 8 - 3 = 5$

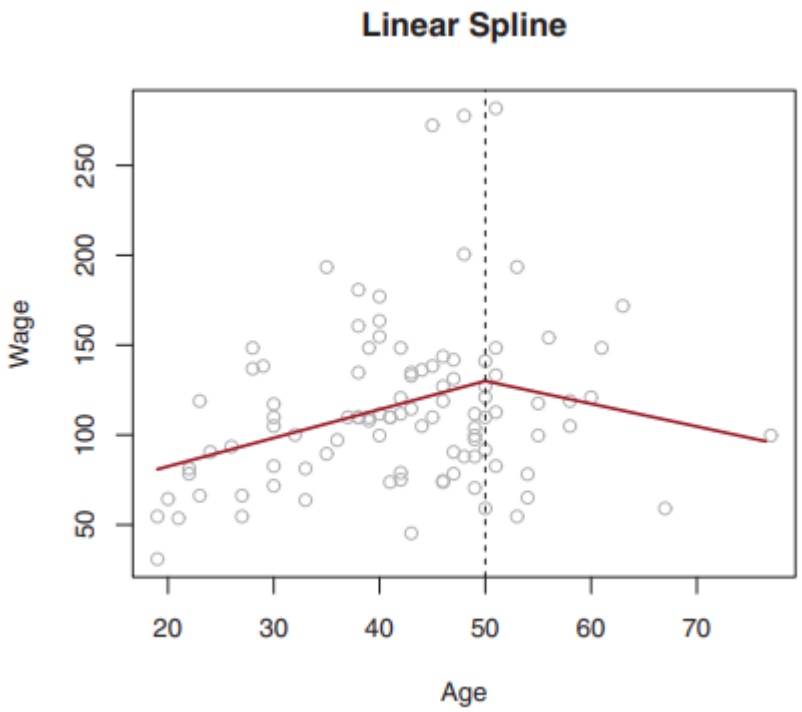


Cubic Spline

- Piecewise cubic polynomials
- Continuity
  - Continuous, 1st derivative is continuous, 2nd derivative is continuous
- Degree of freedom $= K + 4$

Linear Spline

- Piecewise linear
- Continuity
  - Continuous
- Degree of freedom $= K + 2$



Spline

- Piecewise degree-d polynomials
- Continuous derivatives up to degree $d - 1$
- Degree of freedom $= K + d + 1$

7.4.3 The Spline Basis Representation

- Cubic spline with K knots has degree of freedom $K + 4$
- Can a cubic spline be modeled as

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

for some basis functions $b_1, \ldots, b_{K+3}$?

Answer

- Truncated power basis function

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

- Model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 h(x_i, c_1) + \cdots + \beta_{K+3} h(x_i, c_K) + \epsilon$$

Proof in the case of degree 3

- Two cubic polynomials

$$y = \beta_{01} + \beta_{11}x + \beta_{21}x^2 + \beta_{31}x^3$$
$$y = \beta_{02} + \beta_{12}x + \beta_{22}x^2 + \beta_{32}x^3$$

Let

$$f(x) = (\beta_{02} - \beta_{01}) + (\beta_{12} - \beta_{11})x + (\beta_{22} - \beta_{21})x^2 + (\beta_{32} - \beta_{31})x^3$$

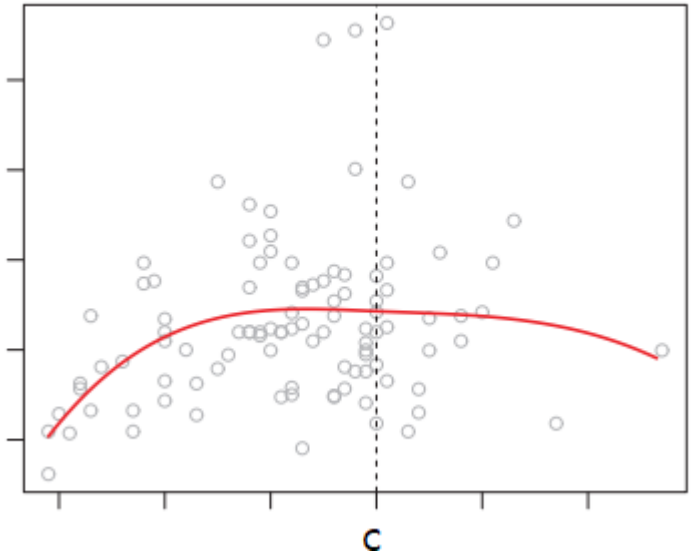
with

$$f(c) = f'(c) = f''(c) = 0$$

Then

$$f(x) = a(x - c)^3$$

for some a



Equivalently

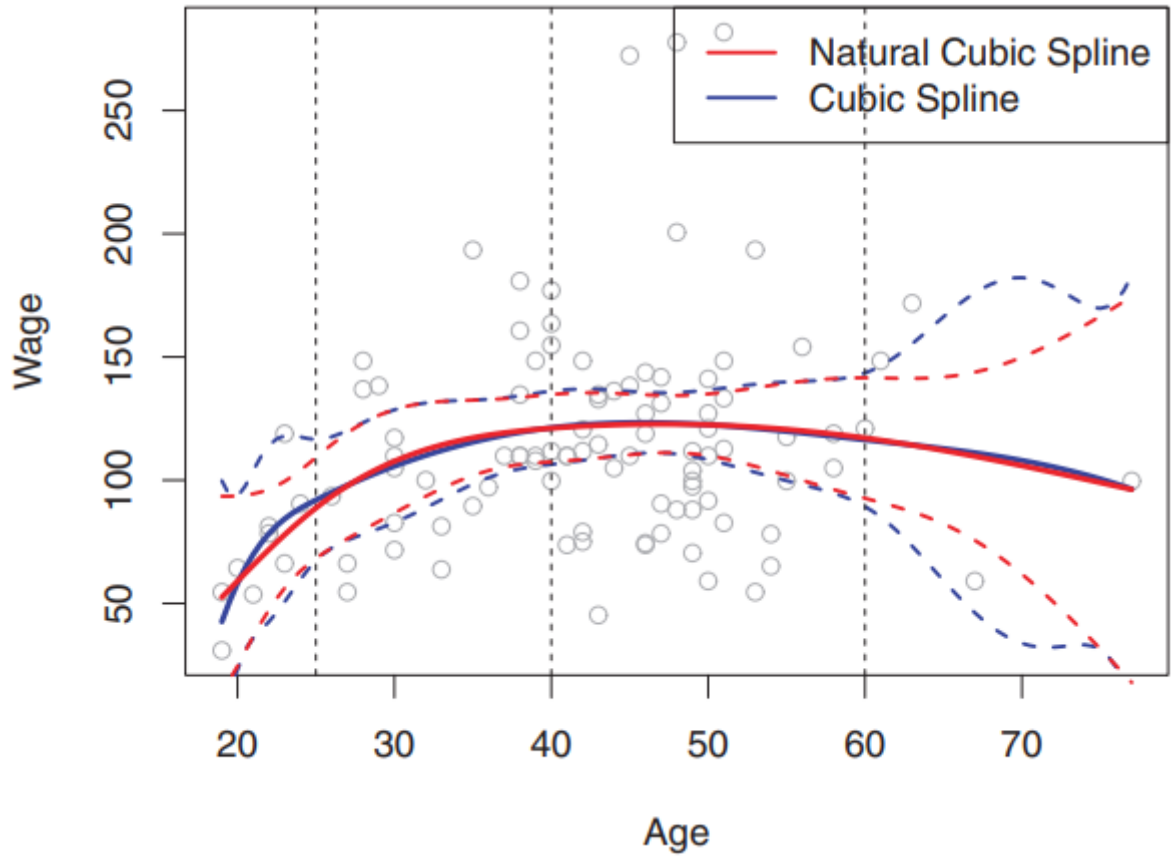
$$\beta_{02} + \beta_{12}x + \beta_{22}x^2 + \beta_{32}x^3 = \beta_{01} + \beta_{11}x + \beta_{21}x^2 + \beta_{31}x^3 + a(x-c)^3$$

Thus we have a model of the form

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 h(x_i, c) + \epsilon$$

Natural Spline

- A regression spline with boundary constraints
  - linear at the boundary
    that is, linear if $x \leq c_1$ or $c_k \leq x$



Model of Natural Spline

- Let

$$d_k(x) = \frac{(x - c_k)_+^3 - (x - c_K)_+^3}{c_K - c_k}$$

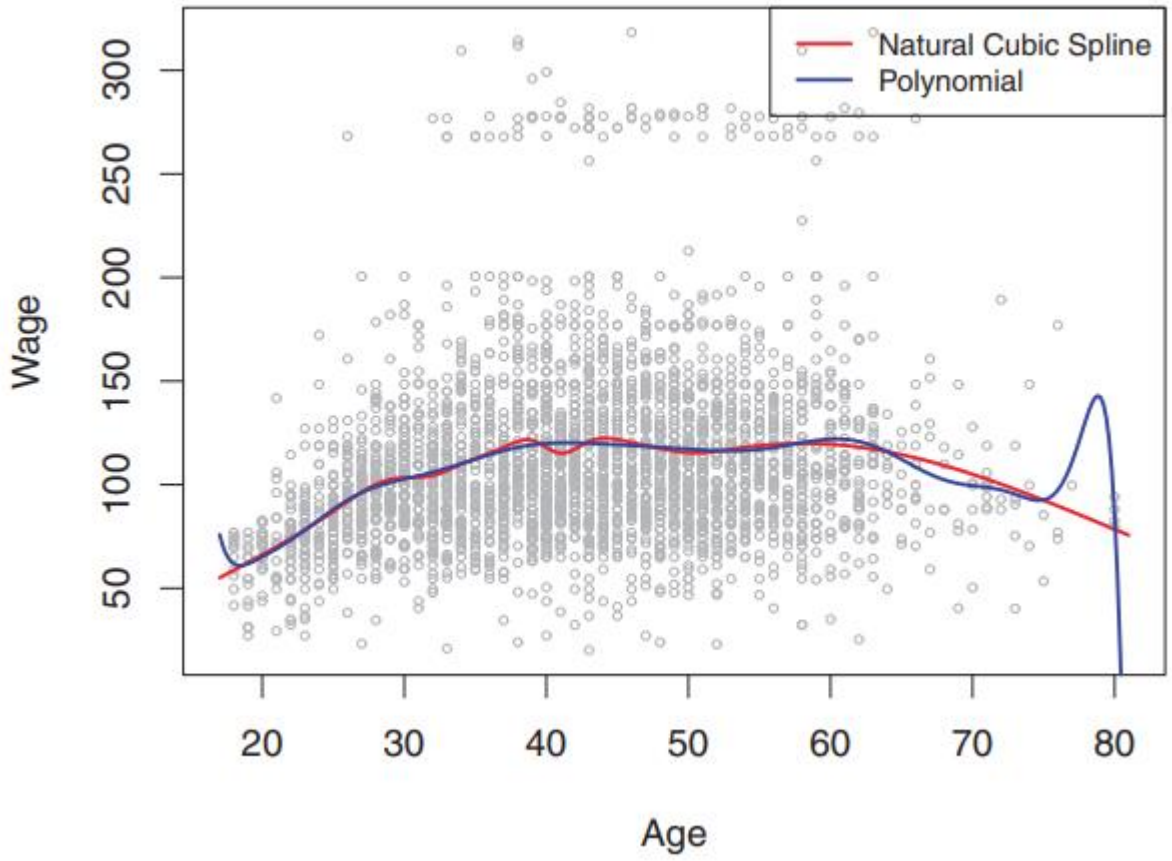
$$b_{k+1}(x) = d_k(x) - d_{K-1}(x)$$

- Model

$$y = \beta_0 + \beta_1 X + \beta_2 b_2(X) + \cdots + \beta_{K-1} b_{K-1}(X) + \epsilon$$

7.4.4 Choosing the Number and Locations of the Knots

7.4.5 Comparison to Polynomial Regression



7.5 Smoothing Splines

- 7.5.1 An Overview of Smoothing Splines
- 7.5.2 Choosing the Smoothing Parameter λ

7.5.1 An Overview of Smoothing Splines

- Fitting
  - Find $g(x)$ that minimizes

$$\text{RSS} = \sum_{i=1}^{n} \big(y_i - g(x_i)\big)^2$$

- Smoothing spline
  - Find g(x) that minimizes "Loss+Penalty"

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

Avoiding Knot Selection Problem

- Using the maximal set of knots with all inputs $x_1, \ldots, x_n$
- Overfitting is controlled by regularization $\int g''(t)^2 dt$

Smoothing spline

- Solution: a natural cubic spline with knots at $x_1, \ldots, x_n$!
  - See: *T. Hastie, R. Tibshirani and J Friedman*, The Elements of Statistical Learning Data Mining, Inference, and Prediction
  - Model

$$y = \beta_0 + \beta_1 X + \beta_2 b_2(X) + \cdots + \beta_{n-1} b_{n-1}(X) + \epsilon$$

where

$$b_{k+1}(x) = d_k(x) - d_{n-1}(x)$$

$$d_k(x) = \frac{(x - x_k)_+^3 - (x - x_n)_+^3}{x_n - x_k}$$

- Let

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

$$B = \begin{bmatrix} b_0(x_1) & \cdots & b_{n-1}(x_1) \\ \vdots & \ddots & \vdots \\ b_0(x_n) & \cdots & b_{n-1}(x_n) \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_{n-1} \end{bmatrix}$$

- Loss + Penalty

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

where $\Omega_{ij} = \int b_i''(t) b_j''(t) dt$

- Solution

$$\hat{\beta} = (B^T B + \lambda \Omega)^{-1} B^T y$$

7.5.2 Choosing the Smoothing Parameter λ

- Cross validation

7.6 Local Regression

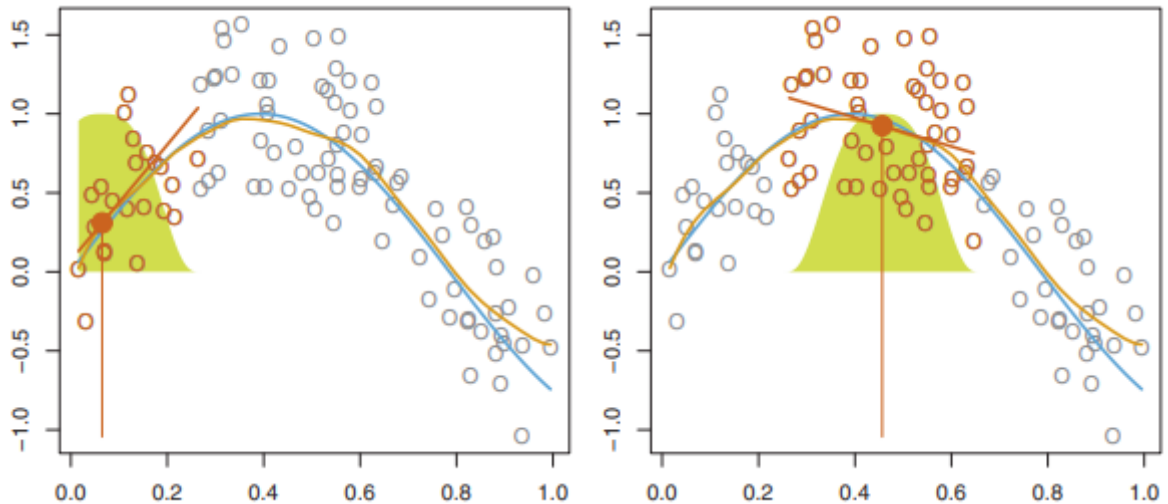
- Local regression
  - Computing the fit at each target point x_0 using nearby training observations
- Memory-based procedure
  - we need all the training data each time we wish to compute a prediction

Algorithm

- Goal: find estimate at x_0
  - 1. Gather k points whose x_i are closest to x_0
  - 2. Assign a weight $K_{i0} = K(x_i, x_0)$ to each x_i in this neighborhood
    - the point furthest from x_0 has weight zero
    - the closest has the highest weight
    - All but these k nearest neighbors get weight zero.
  - 3. Fit a weighted least squares regression
    - find $\hat{\beta}_0$ and $\hat{\beta}_0$ 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 $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$


1.5
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0

7.7 Generalized Additive Models

- Generalized additive models (GAMs)
  - a general framework for generalized additive model extending a standard linear model

7.7.1 GAMs for Regression Problems

- Model

$$\mathrm{Y} = \beta_0 + f_1(X_1) + \cdots + f_p\left(X_p\right) + \epsilon$$

7.7.2 GAMs for Classification Problems

- Model

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