MSBA 7027 Machine Learning
Nonlinear Methods in Regression

Zhengli Wang

Faculty of Business and Economics
The University of Hong Kong
2023

Outline

- Introduction
- Piecewise Polynomials (Regression Splines)
- Nonparametric Methods
  - Smoothing Splines
  - Nonparametric Logistic Regression
- Generalized Additive Models

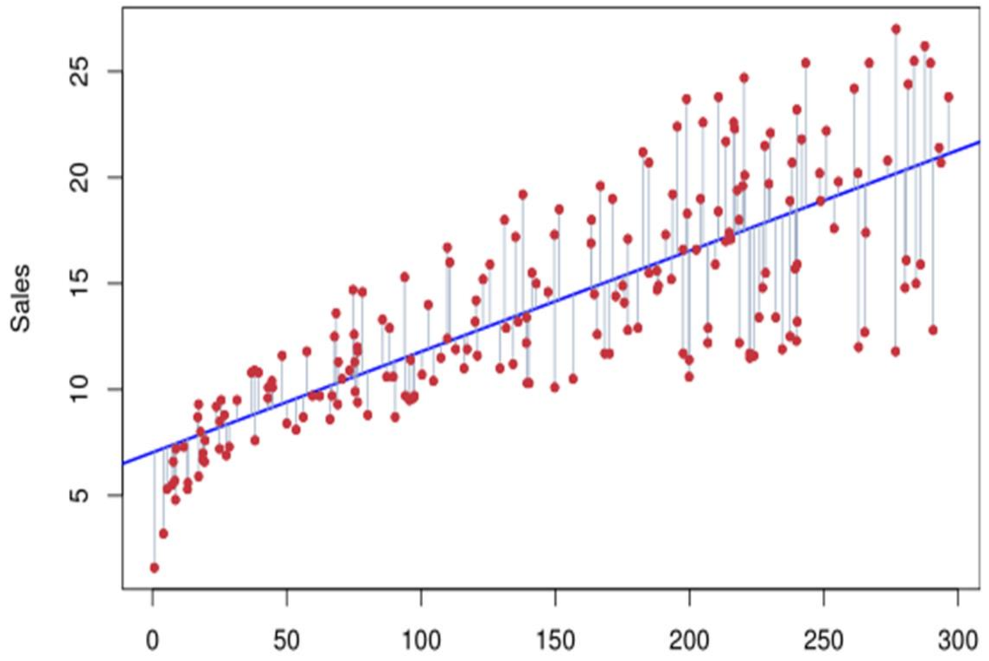
Reading

Chapter 7 of “An Introduction to Statistical Learning”

Outline

- Introduction
- Piecewise Polynomials (Regression Splines)
- Nonparametric Methods
  - Smoothing Splines
  - Nonparametric Logistic Regression
- Generalized Additive Models

Linear Regression

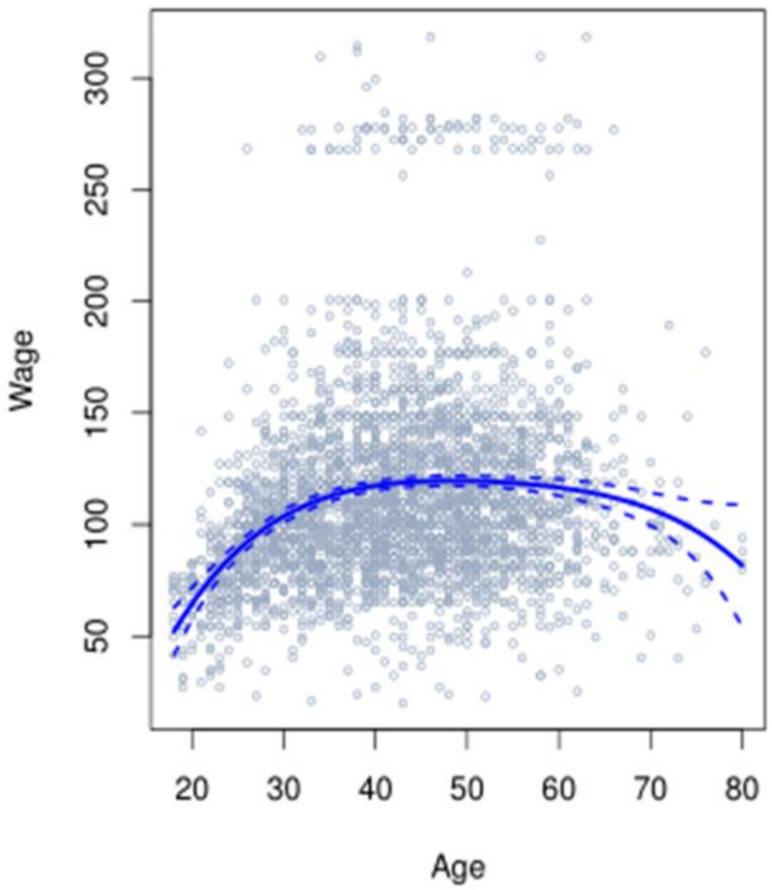


TV ($ spent on advertising)

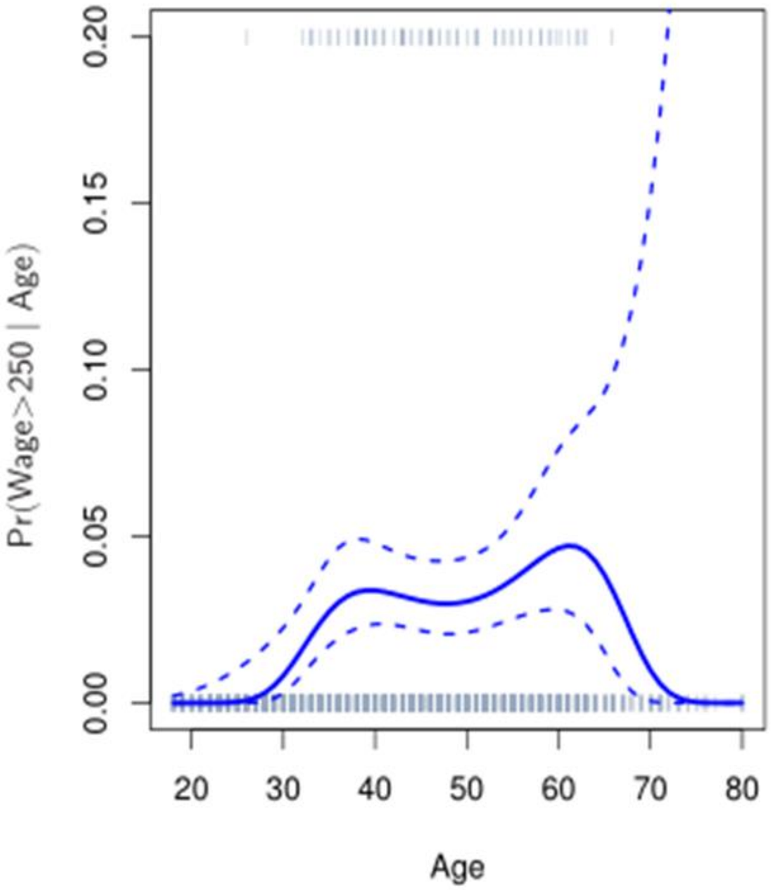
In many scenarios, linear relation may be a poor one to describe relationship between two quantities, e.g.

- Wage’s relationship with age
- Heart disease likelihood’s relationship with blood pressure

Nonlinear example 1: Wage



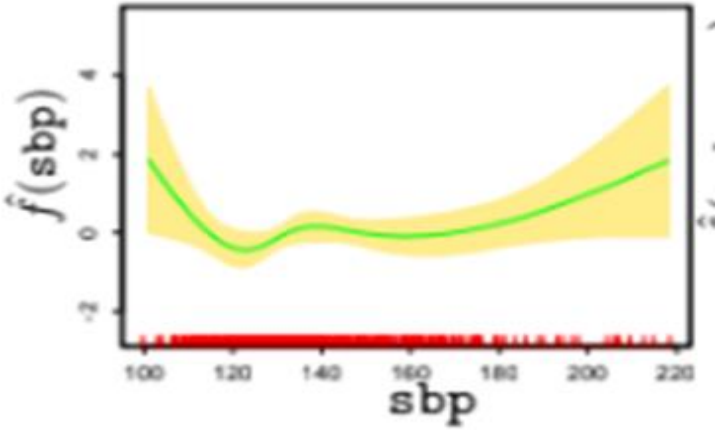
(reg. w/ non-linear models)



(log-reg. w/ non-linear models)

Intuition?

Nonlinear example 2: Heart Disease



Intuition?

Recall MSBA7002

- Linear Regression

$$\mathbb{E}(Y|X)) = f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

Linear model may be a poor one to describe relationship between quantities

Why are linear models still frequently used?

Over large range, may be better to use nonlinear models

Recall MSBA7002

- Linear Regression

$$\mathbb{E}(Y|X)) = f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

However, if we just go from linear to nonlinear methods and are NOT careful: easily overfit

- Nonlinear methods are too powerful
- Need to have some sort of control

Bias-variance tradeoff: linear vs non-linear

This Course: helps you develop a systematic framework on how to appropriately build nonlinear models

Move Beyond Linearity: Derived Input Features

- Have seen models linear in the input features, both for regression and classification.

$$f(X) = \sum_{j=1}^{P} \beta_j X_j$$

- The aim is to find which X_i's are important to predict well.
- To move beyond linearity, essentially do the followings
  - Augment / replace the vector of inputs *X* with additional variables (transformations of *X)*
  - Then use linear models in this new space of these variables (derived input features)

$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X)$$

  - E.g. Plot y = 1 + 2 $\sqrt{X}$ against X vs against $\sqrt{X}$

Move Beyond Linearity

- Denote by $h_m(X)$: $R^P \rightarrow R$ the m[th] transformation
  - $h_m(X) = X_m$
  - $h_m(X) = X_j^2 \; or \; h_m(X) = X_j X_k$
  - $h_m(X) = \log(X_j), \sqrt{X_j}$
  - $h_m(X) = I(L_m \leq X_j < U_m)$
- Polynomial regression
- Step functions
- Piecewise polynomial: spline
- Smoothing splines for regression

Polynomial Regression

- y_{i} is expressed as a polynomial of x_i with degree d

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

where ε_i is the error term

- Fit least square regression with predictors $x_i, x_i^2, \cdots, x_i^d$ (What does *d*=*1* mean? *d*=2?)
- For large *d*, a polynomial regression produces an extremely non-linear curve.
  - A degree-*d* polynomial can have as many as *d-1* turning points.
  - Polynomial regression with a large d suffers from over-fitting.
  - Large d produces overly flexible curve and leads to some very strange shapes
  - Generally speaking, it is unusual to use *d* greater than 3 or 4.

Polynomial Regression

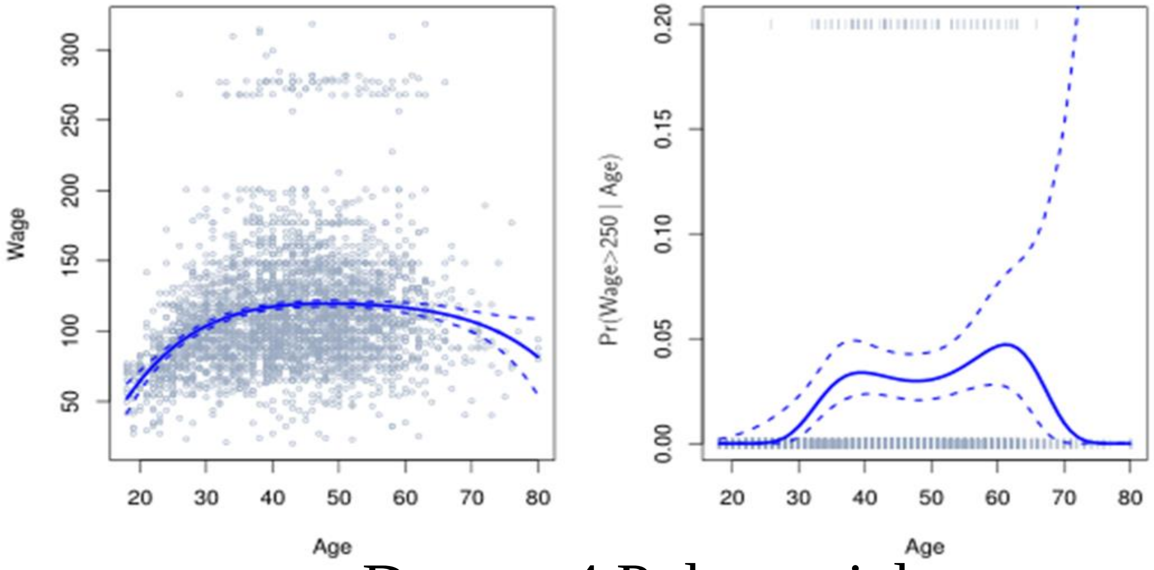
Look at some Code Example

Polynomial Regression

- Polynomial logistic regression

$$log\frac{P(G=1|x)}{P(G=0|x)} = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_d x_i^d$$

Look at some Code Example



Degree-4 Polynomial

Step functions

- Main idea: Break the range of X into bins and fit a different constant in each bin.
- Create cut points $\xi_1, \xi_{2,} \ldots, \xi_k$, in the range of X and then construct $K + 1$ new variables,

$$h_0(X) = I(X < \xi_1)$$
$$h_1(X) = I(\xi_1 \leq X < \xi_2)$$
$$h_2(X) = I(\xi_2 \leq X < \xi_3)$$
$$h_3(X) = I(\xi_3 \leq X < \xi_4)$$
$$\vdots$$
$$h_{k-1}(X) = I(\xi_{k-1} \leq X < \xi_k)$$
$$h_k(X) = I(\xi_k \leq X)$$

Note: 1. if K sufficiently large, step funcs can approximate any func, why?
2. In some problem instances, obvious change of behavior after certain point (Most important thing: determine the right cutoff points)

Step functions

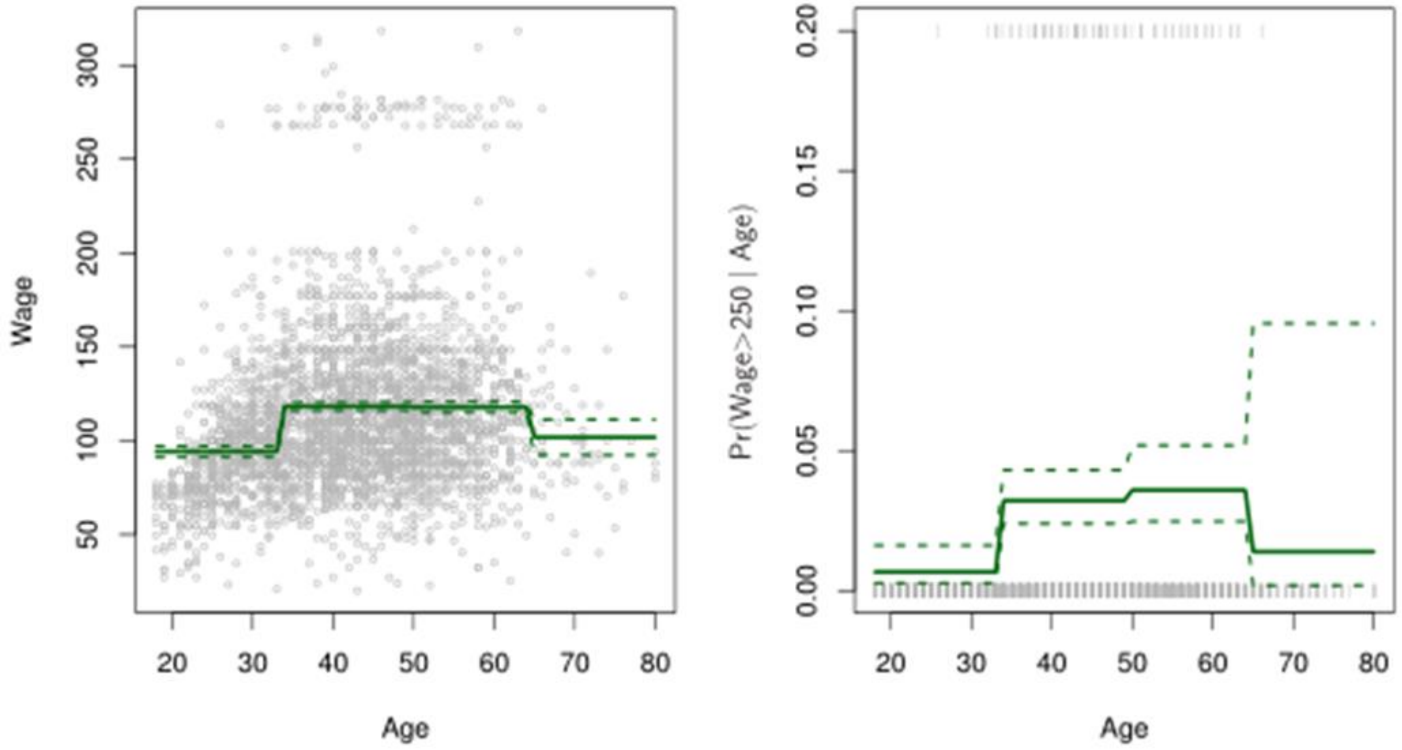
- Fit regression function

$$y_i = \beta_0 + \beta_1 h_1(x_i) + \beta_2 h_2(x_i) + \beta_3 h_3(x_i) + \cdots + \beta_k h_k(x_i)$$

- Interpretation
  - β_0 : Mean value of Y for $X < \xi_1$
  - $\beta_0 + \beta_j$: Mean value of Y for $\xi_j \le X \le \xi_{j+1}$
  - β_j : Difference between mean value of Y for $\xi_j \le X \le \xi_{j+1}$ and mean value of Y and for $X < \xi_1$
- Logistic regression with step functions

$$log\frac{P(G = 1|x)}{P(G = 0|x)} = \beta_0 + \beta_1 h_1(x_i) + \beta_2 h_2(x_i) + \beta_3 h_3(x_i) + \cdots + \beta_k h_k(x_i)$$

Look at some Code Example: step func



No natural breakpoints in the predictors → piecewise constant functions miss the action.

Basis functions

- Polynomial regression and piecewise-constant regression models are **special cases** of a **basis function approach**.

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_k b_k(x_i) + \varepsilon_i$$

- Properties of basis funcs
  - Fixed and known in advance
    - Polynomial reg: $b_j(x_i) = x_i^j$
    - Step func: $b_j(x_i) = I(c_j \leq x < c_{j+1})$
  - Linearly independent

- Essentially, standard linear models with predictors $b_1(x_i), b_2(x_i), b_3(x_i), \ldots$
- Can use all the inference tools for linear models
  - E.g. standard error for coeff. est., ANOVA

Basis functions

- Polynomial regression and piecewise-constant regression models are **special cases** of a **basis function approach**.

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_k b_k(x_i) + \varepsilon_i$$

- Most commonly used basis funcs
  - Regression splines
  - Smoothing splines

Parametric: need to choose cutoff points

Non-parametric: no need to choose cutoff points

- Other basis funcs
  - Fourier basis (Trigonometric funcs)
  - Wavelets

NOT covered in this course;
Splines are useful enough for our purpose

Outline

- Introduction
- Piecewise Polynomials (Regression Splines)
- Nonparametric Methods
  - Smoothing Splines
  - Nonparametric Logistic Regression
- Generalized Additive Models

Piecewise Polynomial Regression

- Main idea: fitting separate polynomials over different regions of X, instead of fitting a polynomial over the entire range.
- The points where the coefficients change are called knots.
- A piecewise cubic polynomial with a single knot at ξ takes the form

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \varepsilon_i \; if \; x_i < \xi \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \varepsilon_i \; if \; x_i \geq \xi \end{cases}$$

- Idea: Fit two separate polynomials on two separate regions
  - 1^{st} poly has coeff. $\beta_{01}, \beta_{11}, \beta_{21}, \beta_{31}$; 2^{nd} poly has coeff. $\beta_{02}, \beta_{12}, \beta_{22}, \beta_{32}$
  - Each poly can be fitted using OLS

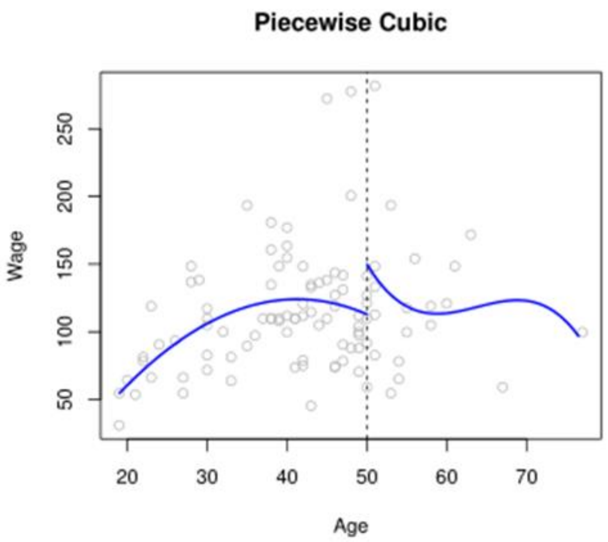
Piecewise Polynomial Regression

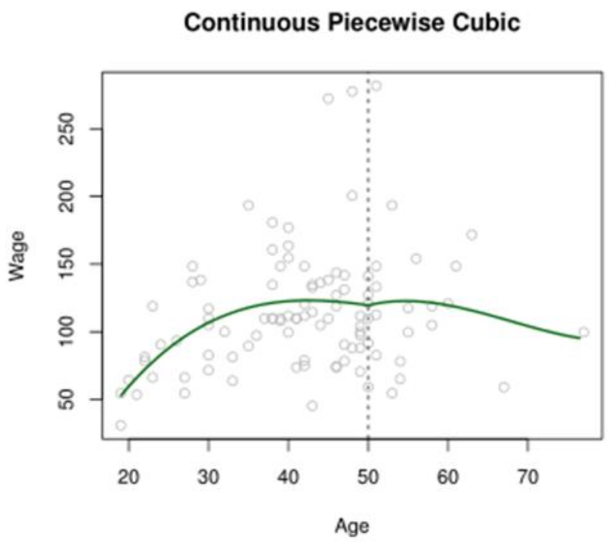
- Main idea: fitting separate polynomials over different regions of X, instead of fitting a polynomial over the entire range.
- The points where the coefficients change are called knots.
- A piecewise cubic polynomial with a single knot at ξ takes the form

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \varepsilon_i \; if \; x_i < \xi \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \varepsilon_i \; if \; x_i \geq \xi \end{cases}$$

- K different knots → What is the # of different polynomials?
- Special cases
  - Polynomial regression = piecewise polynomial regression with 0 knots
  - Step func = piecewise polynomial regression where polynomials are of degree 0 (piecewise constant regression)

Piecewise Cubic Illustration

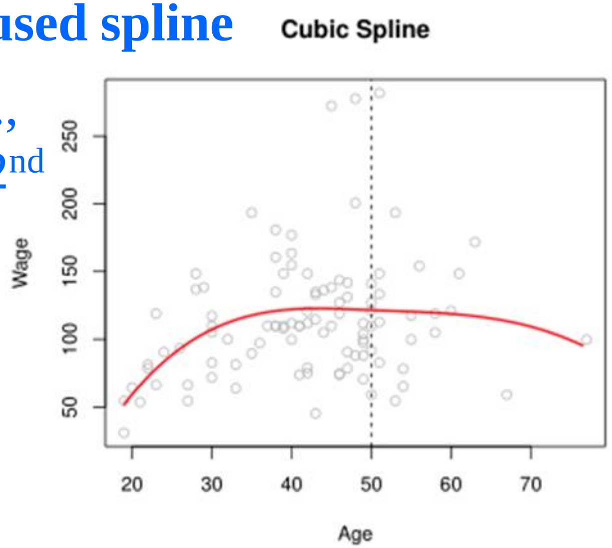




(only constrained to be cont.)

Most widely-used spline

(constrained to be cont., and have cont. 1st and 2nd deri.)



Cubic Spline = Piecewise Cubic + Constraints

- Terminologies
  - Degree d = 3
  - Order M = d+1 = 4
  - #knots K, with placements of knots $\xi_1, \ldots, \xi_K$
- K knots divides the domain of X into $K + 1$ intervals:

$$(-\infty, \xi_1), (\xi_1, \xi_2), \ldots, (\xi_{k-1}, \xi_k), (\xi_k, \infty)$$

- At each knot ξ_j, there is one cubic polynomial on LHS and one on RHS
  - These two polynomials have the same value, 1^{st} and 2^{nd} derivatives at ξ_j.
- Knot-discontinuity NOT visible to the human eye

Cubic Spline = Piecewise Cubic + Constraints

- Terminologies
  - Degree d = 3
  - Order M = d+1 = 4
  - #knots K, with placements of knots $\xi_1, \ldots, \xi_K$

- K knots divides the domain of X into $K + 1$ intervals:

$$(-\infty, \xi_1), (\xi_1, \xi_2), \ldots, (\xi_{k-1}, \xi_k), (\xi_k, \infty)$$

- A cubic spline with K knots has ______ degree of freedom

Cubic Spline Basis

- A cubic spline with K knots can be modeled as:

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

- A truncated power basis function is defined as:

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

- Basis: $1, x, x^2, x^3, h(x, \xi_1), \ldots, h(x, \xi_K)$, i.e. $K + 4$ df

- R func: bs()

Fit the Spline

- Specify #knots (or the #basis functions or df).
- Specify the placement of the knots
- For each training point (x_i, y_i), evaluate the $4+K$ basis functions at the input value x_i and obtain

$$h(x_i) = \left(h_1(x_i), \ldots, h_{4+K}(x_i)\right)^T$$

- Fit a linear model with derived inputs

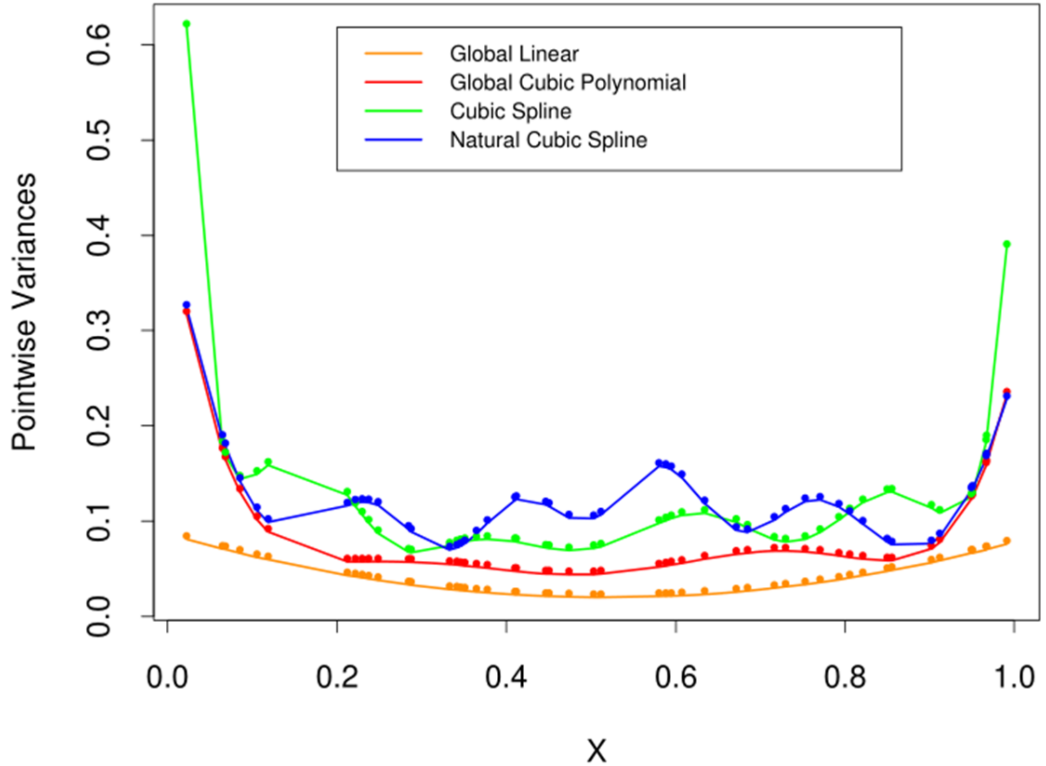
Cubic Spline

- Look at some code examples

Boundary Effect – Motivation to Natural Cubic Spline

- Splines can have high variance at the outer range of predictions
  - i.e. when X is very small or very large

Boundary Effect – Motivation to Natural Cubic Spline



Global Linear (Orange), df = 2: smallest variance

Global Cubic Poly (Red), df = 4: Larger variance

Cubic Spline (Green), df = 6: Variance explodes at the boundary

Natural Cubic Spline (Blue), df = 6: Variance controlled at the boundary