PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY *of* WASHINGTON

Introduction to Machine Learning

MLEARN 510A – Lesson 8

Recap of Lesson 7

- Improving Linear Models – Prediction Accuracy and Model Interpretability
- Subset Selection
- Shrinkage Methods
- Ridge Regression
- Lasso Regression
- Comparison of Shrinkage Methods

Course Outline

1. Introduction to Statistical Learning
2. Linear Regression
3. Classification
4. Model Building, Part 1
5. Model Building, Part 2
6. Resampling Methods
7. Linear Model Selection and Regularization
8. **Moving Beyond Linearity**
9. Bayesian Analysis
10. Dimensionality Reduction

Outline of Lesson 8

- Shrinkage Methods
- Polynomial Regression
- Step Functions
- Basis Functions
- Regression Splines
- Local Regression
- Generalized Additive Models

Shrinkage Methods

- Fit a model containing all *p* predictors using a technique that *constrains* or *regularizes* or *shrinks* the coefficient estimates
- The two best-known techniques for shrinking the regression coefficients
- Ridge Regression
- Lasso Regression

Regularization

- Shrinkage methods come within the realm of Regularization

- ***Regularization*** *is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error*

- How does regularization help?
  - Encourages a more parsimonious description of the model
  - Prevents the weights/learned parameters from becoming too large
  - Smaller weights generate a simpler model and help avoid overfitting

Regularization as Constrained Optimization

- Minimize some loss function while limiting the model complexity

  minimize Loss(Data|Model)

  such that complexity(Model) <= t

- The regularized objective function is written as

  minimize Loss(Data|Model) + λcomplexity(Model)

- Our training optimization algorithm is now a function of two terms:
  - **Loss term:** measures how well the model fits the data
  - **Regularization term:** measures model complexity

- λ – Controls strength of regularization

Penalty Terms of Ridge and Lasso

- Ridge Regression minimizes

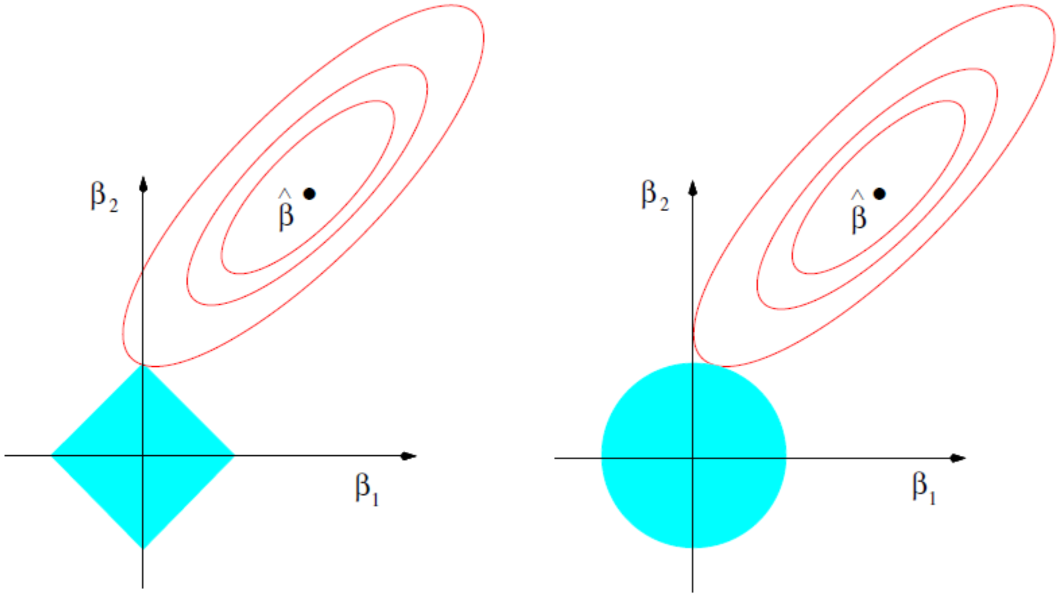
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2,$$

- The LASSO minimizes

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|.$$

How Does the Choice of Regularization Affect the Final Solution?

- Plot shows contours of error (red) and constraint (blue) functions for Lasso (left) and Ridge (right) regularization
- Find the first point where the ellipses hit the constraint region
- Unlike the disk, the diamond has corners; if the solution occurs at a corner, then it has one parameter βj equal to zero i.e., Lasso promotes sparsity by setting some weights to zero



Ridge vs. Lasso Regularization

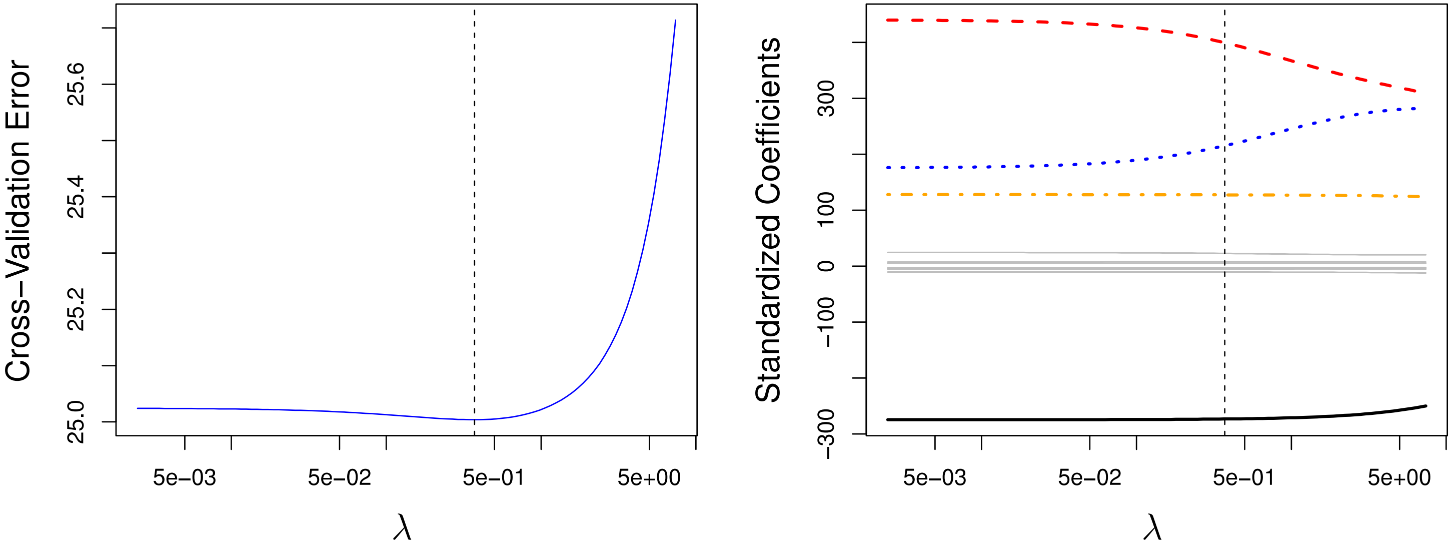
- The table below captures the differences between the two forms of regularization

Ridge	Lasso
L2 penalizes the sum of squares of weights	L1 penalizes the sum of absolute value of weights
L2 does not have a sparse solution	L1 has a sparse solution
L2 has no feature selection and is not robust to outliers	L1 has in-built feature selection and is robust to outliers
L2 gives better performance when the output is a function of all input variables	L1 models find it hard to learn complex patterns

Selecting λ

- How to pick a value for λ?

- Select a grid of potential values, use cross validation to estimate the error rate on test data (for each value of λ) and select the value that gives the least error rate



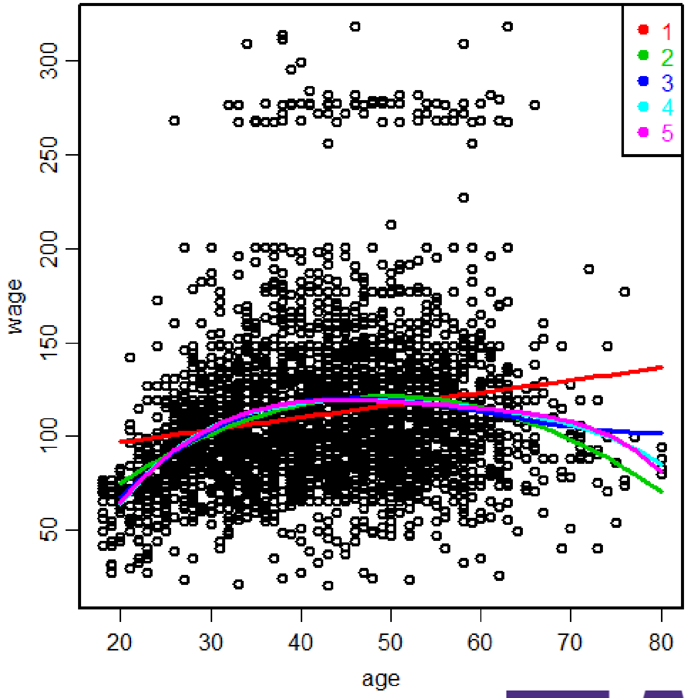
So Far ...

- Supervised Learning
  - A single response y
  - Multiple predictors $x_1, x_2, \dots, x_p$
- Linear Models
- Regularized Models
- **The assumption of linearity:** $\boldsymbol{f(x_1, x_2, \dots, x_p)}$

Non-Linear Models

What can be done when linearity is not good enough?

- ➢ Polynomial regression
- ➢ Step functions
- ➢ Regression splines
- ➢ Smoothing splines
- ➢ Local regression
- ➢ Generalized additive models (GAM)



Polynomial Regression

$$\begin{array}{ll}
\text{linear function} & : f(x) = \beta_0 + \beta_1 x \\
\text{quadratic function} & : f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 \\
\text{cubic function} & : f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 \\
\vdots & \qquad\qquad \vdots \\
\text{degree-}d \text{ polynomial:} & f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_d x^d
\end{array}$$

It's just the standard linear model

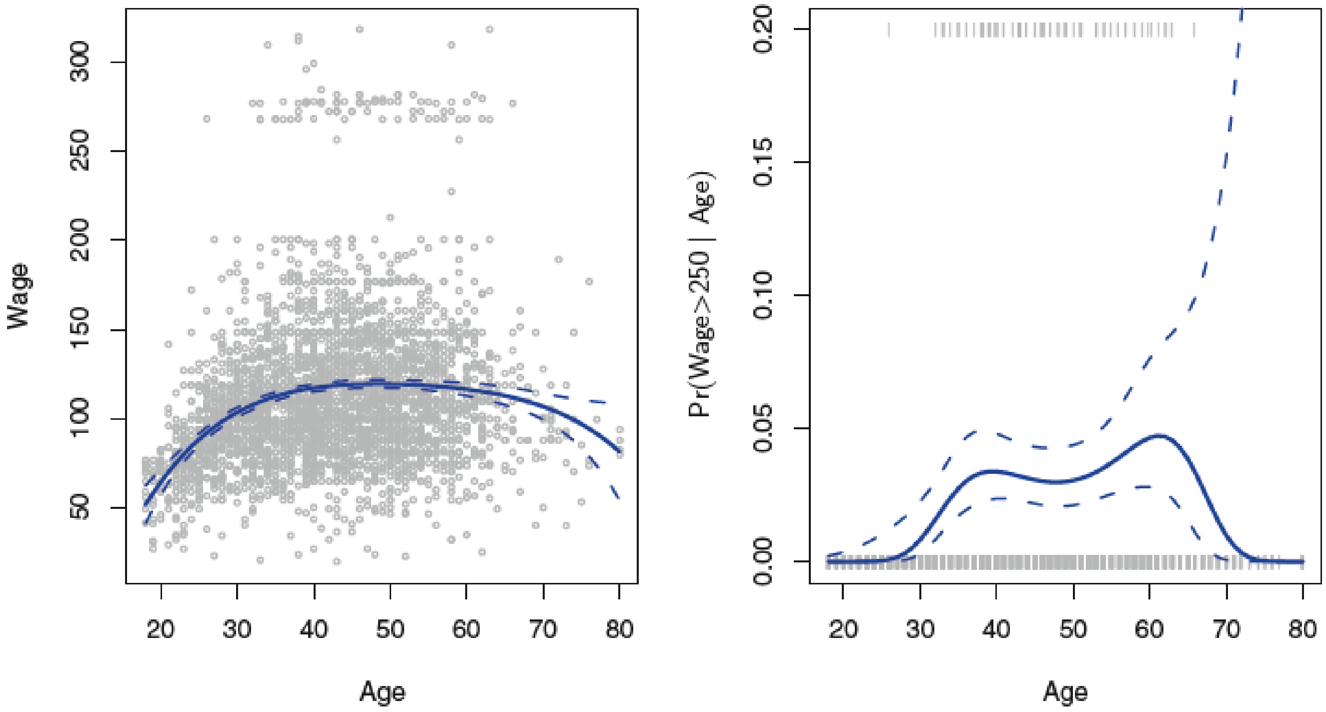
$$f(x_1, \ldots, x_d) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_d x_d$$

where

$$x_1 = x,\ x_2 = x^2,\ \ldots,\ x_d = x^d$$

Polynomial Regression

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



Polynomial Regression

In the left-hand panel of the figure, the solid blue curve is given by

$$\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \hat{\beta}_3 x^3 + \hat{\beta}_4 x^4$$

and the pair of dotted blue curves indicate an estimated 95% confidence interval given by

$$\hat{f}(x) \pm 2 \cdot se\{\hat{f}(x)\}$$

In the right-hand panel, the solid blue curve is given by

$$\hat{\pi}(y > 250 \,|\, x) = \exp\{\hat{f}(x)\}/[1 + \exp\{\hat{f}(x)\}] = \mathrm{sigm}\{\hat{f}(x)\}$$

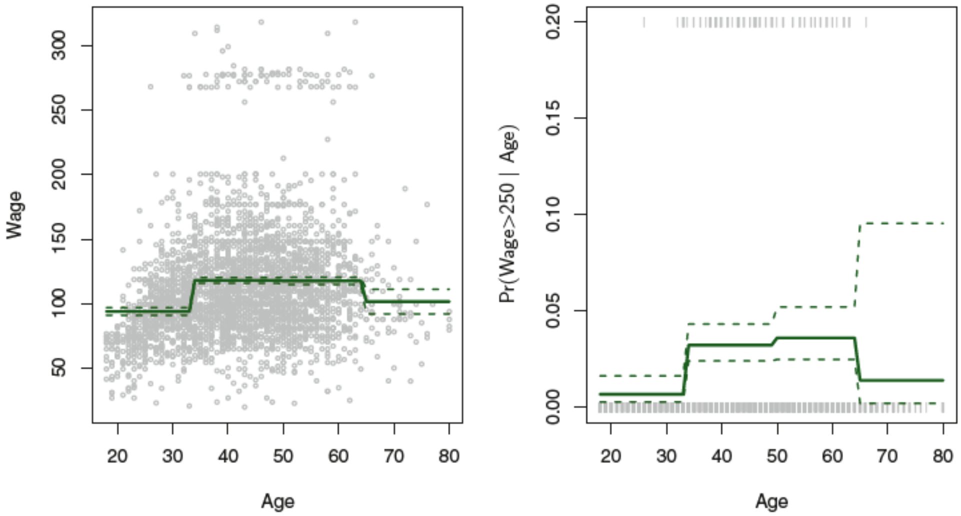
and the pair of dotted blue curves indicate an estimated 95% confidence interval given by

$$\mathrm{sigm}[\hat{f}(x) \pm 2 \cdot se\{\hat{f}(x)\}]$$

Step Functions

- Impose local, rather than global structure
- Cut the variable into distinct regions:

$$
\begin{aligned}
C_0(X) &= I(X < c_1),\\
C_1(X) &= I(c_1 \leq X < c_2),\\
C_2(X) &= I(c_2 \leq X < c_3),\\
&\vdots\\
C_{K-1}(X) &= I(c_{K-1} \leq X < c_K),\\
C_K(X) &= I(c_K \leq X),
\end{aligned}
$$

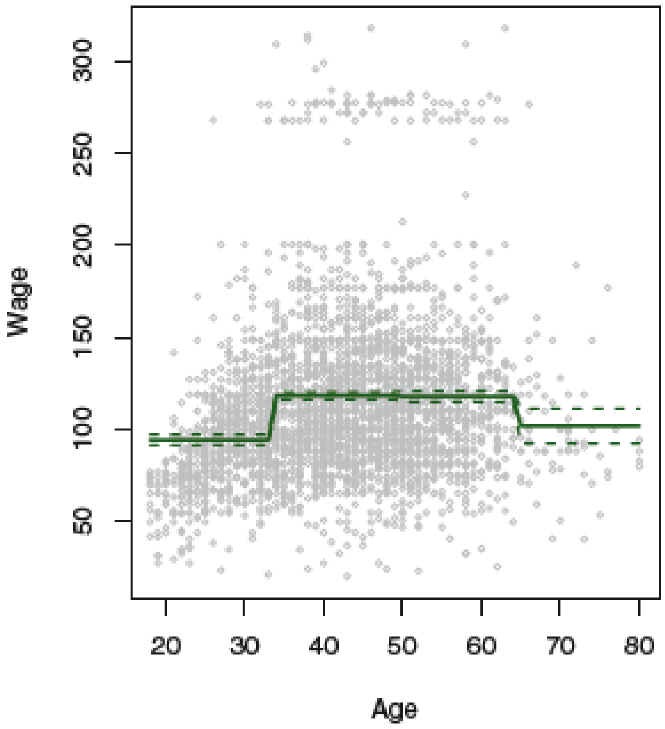


- Then use least squares

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

Step Functions: Details

- Easy to work with; effectively, they convert the data into series of categorical variables:
  - Age<=35
  - Age>35 & Age<=65
  - Age>65



- In downstream analysis you can create interaction variables, etc.

- However, choice of the knots (breaks) can be problematic

Basis Functions

- Basis function approach attempts to create basis functions that can be applied to X: $b_1(X), \dots, b_K(X)$, then fit a linear model

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

Note that basis functions are fixed and known

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

- For example, polynomial regression uses basis functions

$$b_j(x_i) = x_i^j, j = 1, \ldots, K$$

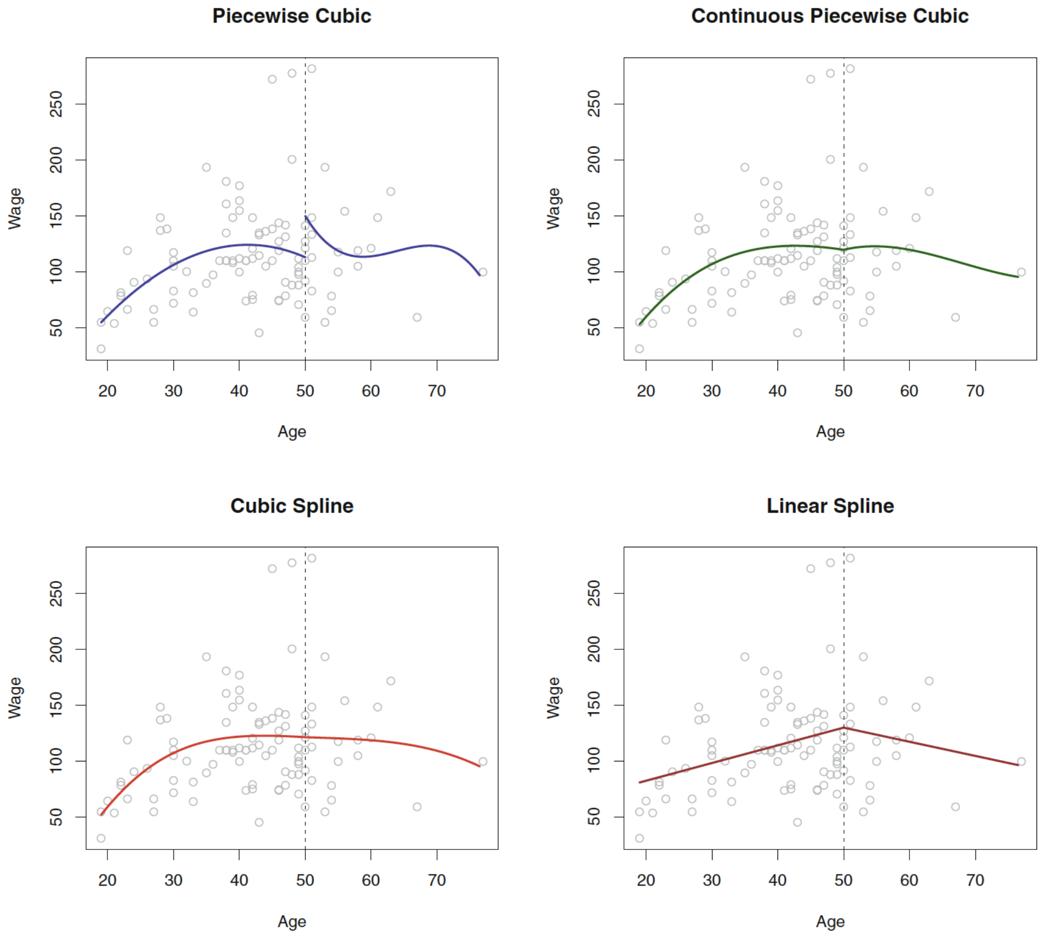
Piecewise Polynomials

- Hybrid of step function approach and polynomial function approach
- Divide the range of values of covariates into sub-intervals same as step function approach
- The points where the coefficients change are called *knots*
- Use a polynomial function on each sub-interval
- For example, a piecewise cubic polynomial with a single knot at a point c takes the form

$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \epsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \epsilon_i & \text{if } x_i \geq c. \end{cases}$$

- Advantage: capture local variation; the degree of polynomial is generally low
- Disadvantage: discontinuity at knots

Piecewise Polynomials



Piecewise Cubic

Continuous Piecewise Cubic

Cubic Spline

Linear Spline

Top Left: No constraint

Top Right: Continuity at Age=50

Bottom left: Continuity, 1st and 2nd derivative the same at Age=50

Bottom right: Continuity, at Age=50

Regression Splines

- A degree-d spline can be represented by the linear model

$$f(x) = \beta_0 + \beta_1 b_1(x) + \cdots + \beta_d b_d(x) + \beta_{d+1} b_{d+1}(x) + \cdots + \beta_{d+K} b_{d+K}(x)$$

Where

$$b_1(x) = x$$
$$b_2(x) = x^2$$
$$\cdots\cdot$$
$$b_d(x) = x^{\mathrm{d}}$$

$$b_{d+k}(x) = \begin{cases} (x - \xi_k)^d \; if \; x > \xi_k \\ 0 \; Otherwise \end{cases}$$

Are basis functions

What is a Spline?

- A ‘spline’ is a function that is constructed piece-wise from polynomial functions

- The term comes from the tool used by shipbuilders and drafters to construct smooth shapes having desired properties

- Drafters have long made use of a bendable strip fixed in position at a number of points that relaxes to form a smooth curve passing through those points

- The malleability of the spline material combined with the constraint of the control points would cause the strip to take the shape that minimized the energy required for bending it between the fixed points

Linear Splines

- A linear spline with knots at ξ_k, k = 1, …, K is a piecewise linear polynomial continuous at each knot

- We can represent this linear spline as:

$$f(x) = \beta_0 + \beta_1 b_1(x) + \beta_2 b_2(x) + \cdots + \beta_{1+K} b_{1+K}(x)$$

Where

$$b_1(x) = x$$

$$b_{1+k}(x) = \begin{cases} (x - \xi_k)\ if\ x > \xi_k \\ 0\ Otherwise \end{cases}$$

Quadratic Splines

- A quadratic spline with knots at ξ_k, k = 1, …, K can be modeled as:

$$f(x) = \beta_0 + \beta_1 b_1(x) + \beta_2 b_2(x) + \cdots + \beta_{2+K} b_{2+K}(x)$$

Where

$$b_1(x) = x$$

$$b_2(x) = x^2$$

$$b_{2+k}(x) = \begin{cases} (x - \xi_k)^2 \; if \; x > \xi_k \\ 0 \; Otherwise \end{cases}$$

Cubic Splines

- A cubic spline with knots at ξ_k, k = 1, …, K is a piecewise cubic polynomial with continuous derivatives up to order 2 at each knot
- It can be modeled as:

$$f(x) = \beta_0 + \beta_1 b_1(x) + \beta_2 b_2(x) + \cdots + \beta_{3+K} b_{3+K}(x)$$

Where

$$b_1(x) = x$$
$$b_2(x) = x^2$$
$$b_3(x) = x^3$$

$$b_{3+k}(x) = \begin{cases} (x - \xi_k)^3 \; if \; x > \xi_k \\ 0 \; Otherwise \end{cases}$$

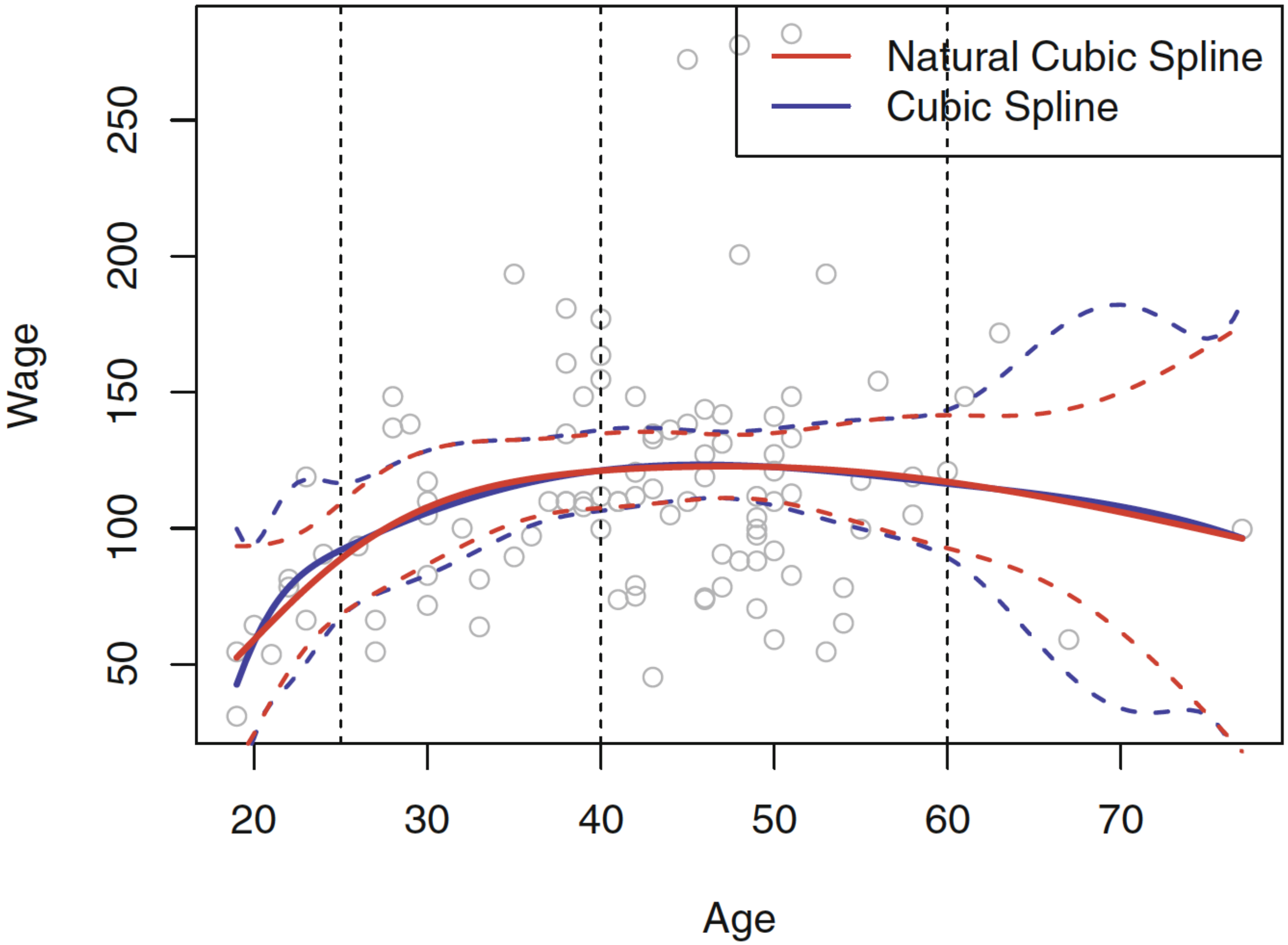
Cubic Splines: Degrees of Freedom

- The degree of freedom of a cubic spline with K knots is:
  - $4 \times (K + 1) - 3K = K + 4$
- Because:
  - There are K+1 functions
  - Each function has 4 parameters
  - Each knot has 3 constraints:
    - Continuity
    - Continuity of 1st derivative
    - Continuity of 2nd derivative

Natural Splines

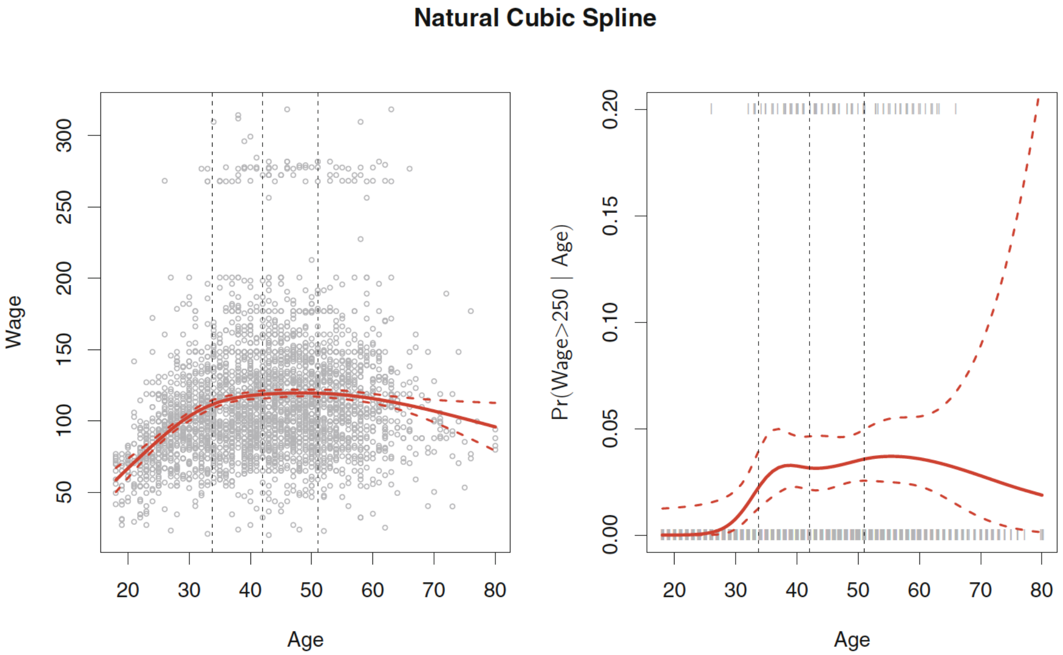
- Spline can have high variance at the outer range of the predictors
- A natural spline is a regression spline with additional boundary constraints
- Function is required to be linear at the boundary (in the region where *X* is smaller than the smallest knot, or larger than the largest knot)
- This additional constraint means that natural splines generally produce more stable estimates at the boundaries

Natural Splines



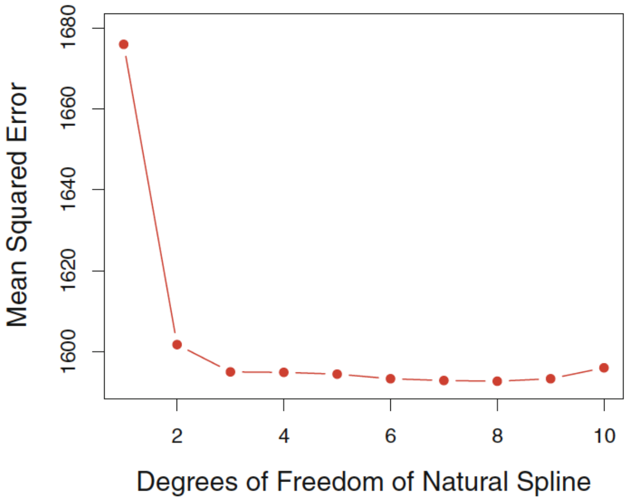
Choosing the Location of Knots

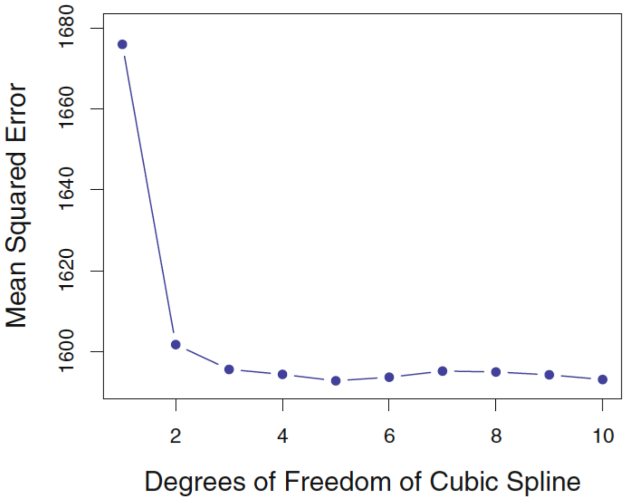
- Where should we place the knots?
  - More knots should be placed where the function $f(x)$ might vary more rapidly
  - Fewer knots should be placed where the function $f(x)$ might seem more stable
  - More common solution is to place knots over quantiles of the data



How Many Knots?

- How many knots should we use?
  - Try different number of knots to see which produces a better fit
  - Cross-validation





Comparison With Polynomial Regression

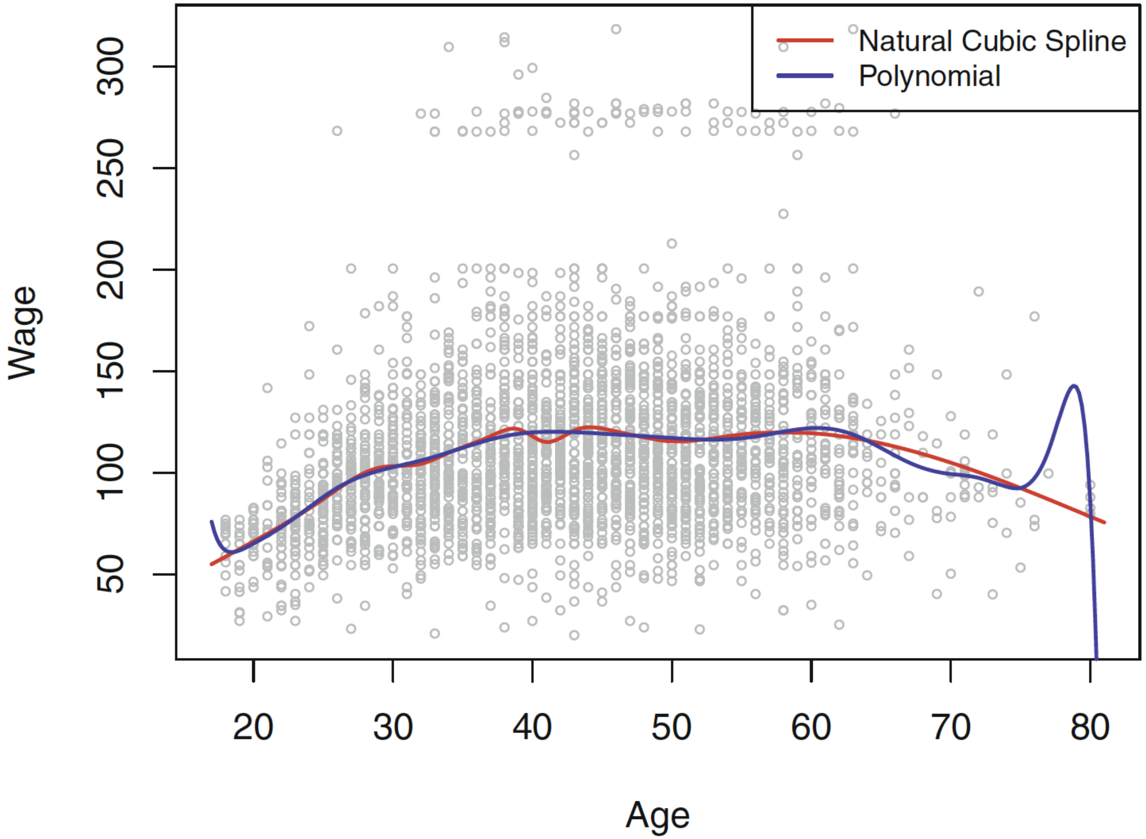
Polynomial Regression

- ➢ Require high orders of polynomials to produce flexible fits
- ➢ Poor boundary behavior

Regression splines

- ➢ Typically give better results
- ➢ Flexible fits are provided by increasing number of knots
- ➢ Possible to introduce more knots at highly flexible regions
- ➢ Poor boundary behavior for cubic splines, better for natural splines

Comparison With Polynomial Regression



Smoothing Splines

- In finding a smooth curve that fits the data, we typically want to find *g(x)* that minimizes:

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

- Subject to some ‘smoothness’ constraints (without it, the curve would go through all the points)
- The most common constraint is:

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

- Where $\lambda \int g''(t)^2 dt$ is the roughness penalty term

- The function *g* that minimizes (1) is known as a

  smoothing spline

The function g(t)

- The notation $g''(t)$ indicates the second derivative of the function g
- $g'(t)$ measures the slope of a function at t, and $g''(t)$ corresponds to the amount by which the slope is changing
- $g''(t)$ is a measure of its *roughness*: it is large in absolute value if $g(t)$ is very wiggly near t, and it is close to zero otherwise
- If g is very smooth, then $g'(t)$ will be close to constant and $\int g''(t)^2 dt$ will take on a small value
- Conversely, if g is jumpy and variable then $g'(t)$ will vary significantly and $\int g''(t)^2 dt$ will take on a large value

Quiz

- λ controls the amount of the roughness penalty
- How does g(x) look like when $\lambda = 0$?
- How does g(x) look like when $\lambda = \infty$?
- What tradeoff is λ controlling here?

The Tuning Parameter

- λ controls the amount of the roughness penalty
- If $\lambda = 0$ no penalty, degree of freedom = n (likely overfit)
- If $\lambda = \infty$ infinity penalty; $f(x)$ must be linear (degree of freedom = 2)
- What is the degree of freedom when $\lambda > 0$ and is finite?
  - It's called effective degree of freedom, denoted as df_λ

Solving for *g(t)*

- The function *g(t)* that minimizes constrained RSS has a few properties:
  - It is a natural cubic spline with knots at every unique value of $x_1,{}_{,}x_2,\ldots,x_n$ and continuous first and second derivatives at each knot
  - It is linear in the region outside of the extreme knots

- Smoothing splines avoid the knot selection issue, leaving a single λ to be chosen

- It is a shrunken version of a natural cubic spline with knots at the unique values of $x_1,{}_{,}x_2,\ldots,x_n$

Effective Degrees of Freedom

- Effective Degree of Freedom, df_λ, is a measure of the flexibility of the smoothing spline

- The vector of fitted values is a linear combination of y and can be written as

$$\hat{\mathbf{g}}_\lambda = \mathbf{S}_\lambda \mathbf{y},$$

Where S_λ is a $n \times n$ matrix

- The **effective degrees of freedom** is then defined to be

$$df_\lambda = trace(S_\lambda)$$

Choice of λ

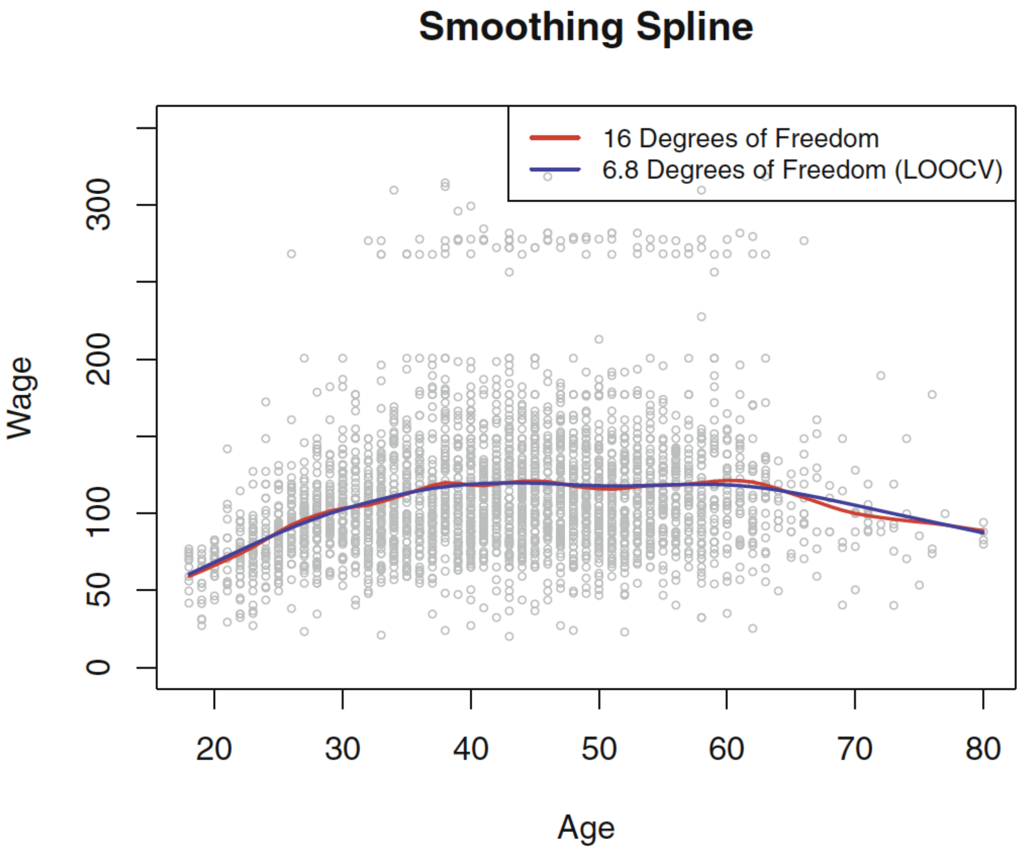
- Cross-validation

- For LOOCV, it can be shown that

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

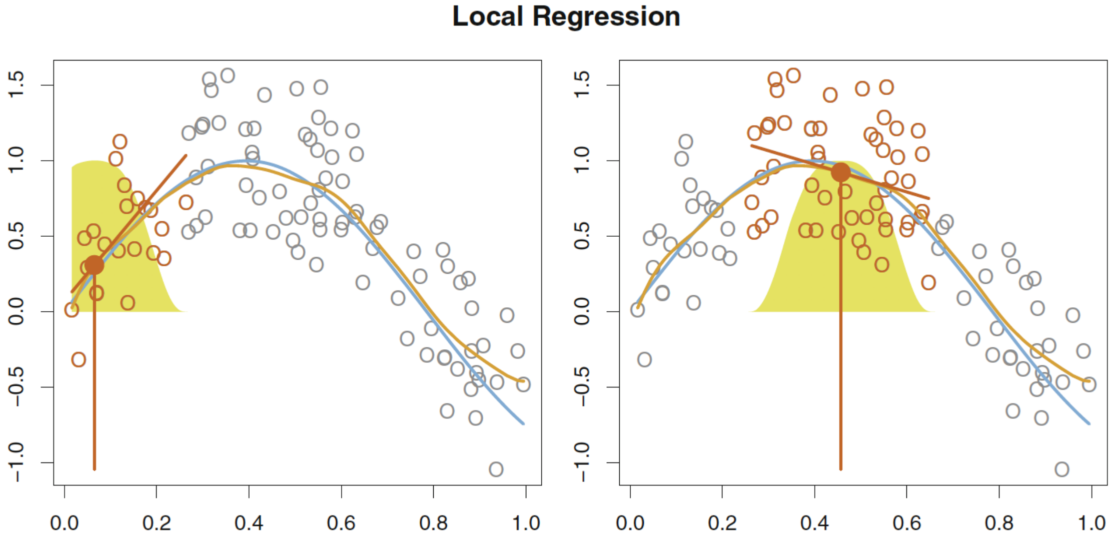
- One fit does it all!

Smoothing Splines: Wage Data Set



Local Regression

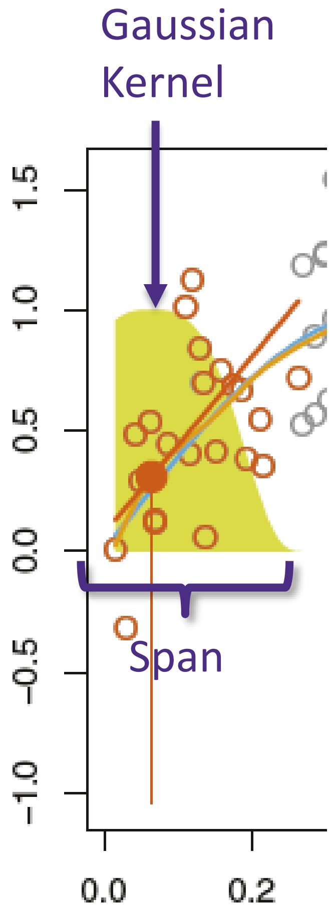
- It involves computing the fit at a target point x_0 using only the nearby training observations (kind of like k-NN)



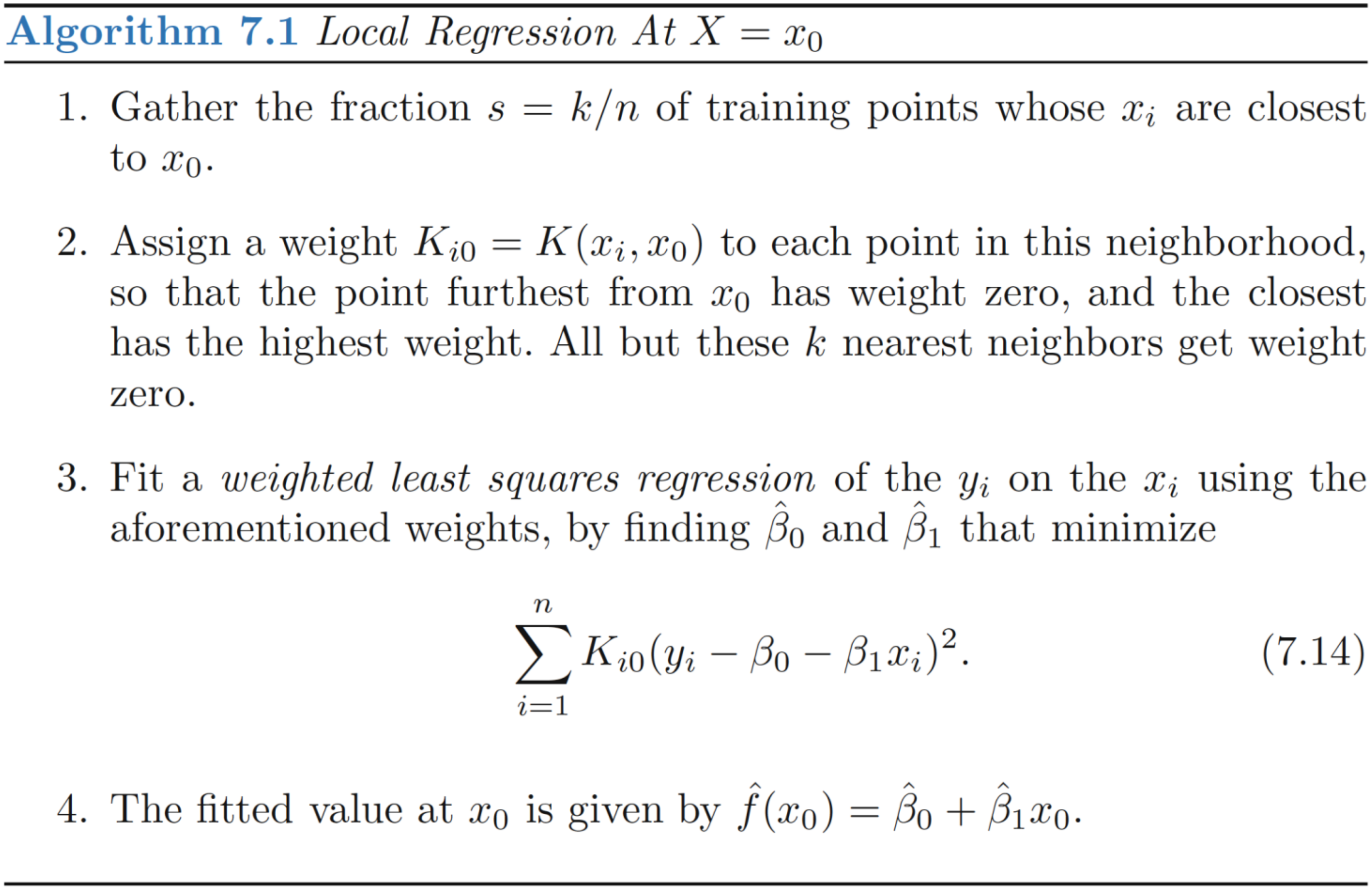
- Previous methods typically maintained a ‘global’ view during function fitting

Local Regression

- To perform Local Regression, we need to specify:
- Weighting function (Kernel):
  - Uniform kernel
  - Triangle kernel
  - Gaussian kernel
- Regression function:
  - Constant
  - Linear
  - Quadratic
- Span (bandwidth):
  - # of points that influence the fit (typically the most important decision)



Local Regression



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

Quiz

- How does the span *s* control the bias-variance tradeoff?
- What if *s* is very large?
- What if *s* is very small?

Generalized Additive Models

- A flexible way to predict response y from multiple predictors $x_1, \dots, x_p$

- A natural way to extend the standard multiple linear model

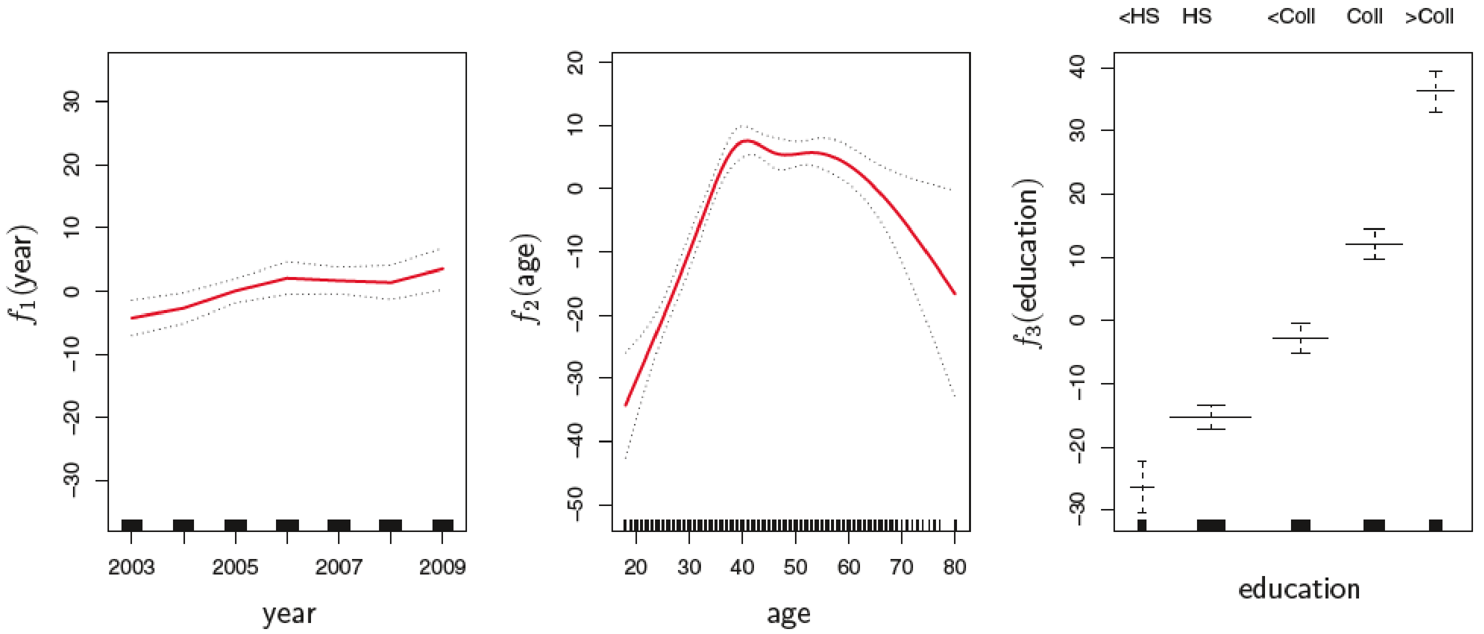
$$f(x_1, \dots, x_p) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

- In order to allow for non-linear relationships, is to write

$$f(x_1, \dots, x_p) = \beta_0 + f_1(x_1) + \cdots + f_p(x_p)$$

Where $f_1(x_1)$,…, $f_p(x_p)$ are (smooth) non-linear functions

GAM: Wage Data Set



$f(x_1, x_2, x_3) = \beta_0 + f_1(x_1) + f_2(x_2) + f_3(x_3)$, where $f_1(x_1)$ and $f_2(x_2)$ are natural splines, and $f_3(x_3)$ is a piecewise linear function

GAM: Pros and Cons

- Can automatically model non-linear relationships that standard linear regression will miss, leading to more accurate predictions

- Model is still interpretable

- The smoothness of the function f_j for the variable x_j can be summarized via degrees of freedom

- Cons: GAMs are restricted to be additive so that some important interactions can be missed

- More flexible alternatives include random forests and boosting

Resources

- *Chapter 5: Elements of Statistical Learning*
- *GAM: The Predictive Modeling Silver Bullet*

Jupyter Notebook

- *Case Study*

ON-BRAND STATEMENT

FOR GENERAL USE

> What defines the students and faculty of the University of Washington? Above all, it’s our belief in possibility and our unshakable optimism. It’s a connection to others, both near and far. It’s a hunger that pushes us to tackle challenges and pursue progress. It’s the conviction that together we can create a world of good. And it’s our determination to Be Boundless. Join the journey at **uw.edu**.