COMP 680
Statistics for Computing and Data Science
Week 13: Beyong Linear Models

Su Chen, Assistant Teaching Professor,
Rice D2K Lab

Outline

The Truth is probably NOT Linear

- But often the linearity assumption is good enough
  - “all models are wrong, some are useful.”

The Truth is probably NOT Linear

- But often the linearity assumption is good enough
  - "all models are wrong, some are useful."
- When linearity is clearly not enough:
  - polynomial regression
  - spline models
  - generalized additive models (GAM)
  - local regression
  - fully non-parametric models

Outline

Motivation

- High dimensional data
  - but OLS solution is **not available for** $p > n$

Motivation

- High dimensional data
  - but OLS solution is **not available for** $p > n$
- Prediction accuracy:
  - when p is large, need to control the variance
  - trade bias with variance to decrease MSE

Motivation

- High dimensional data
  - but OLS solution is **not available for** $p > n$
- Prediction accuracy:
  - when p is large, need to control the variance
  - trade bias with variance to decrease MSE
- Model Interpretability:
  - penalize large models and large slopes
  - automatically perform variable selection

Shrinkage Estimate

- We fit a model involving all p covariates
- But the estimated slopes are shrunken towards 0 relative to OLS.
- This shrinkage is known as regularization
  - penalize “large” β by shrinking them - reduce variance
  - shrink some β to exactly 0 - variable selection

Ridge Regression

- Recall that OLS estimates $\hat{\beta}^{OLS}$ using the values that minimize

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

Ridge Regression

- Recall that OLS estimates $\hat{\beta}^{\boldsymbol{OLS}}$ using the values that minimize

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

- In contrast, the ridge regression coefficient estimates $\hat{\beta}^{\boldsymbol{R}}$ are the values that minimize

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \text{RSS} + \lambda \|\boldsymbol{\beta}\|_{L2}^2,$$

where $\lambda \geq 0$ is a tuning parameter , to be determined separately

Ridge Regression

- As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small.

Ridge Regression

- As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small.
- However, the second term, $\lambda\|\boldsymbol{\beta}\|_{L2}^2$, called a shrinkage penalty , is small when $\beta_1, \cdots, \beta_p$ are close to zero, and so it has the effect of shrinking the estimates of β_j towards zero.

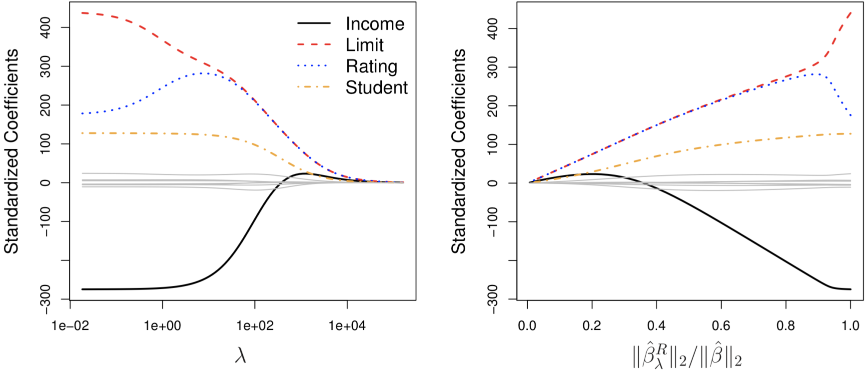
Ridge Regression

- As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small.
- However, the second term, $\lambda\|\boldsymbol{\beta}\|_{L2}^2$, called a shrinkage penalty , is small when $\beta_1, \cdots, \beta_p$ are close to zero, and so it has the effect of shrinking the estimates of β_j towards zero.
- The tuning parameter λ serves to control the relative impact of these two terms on the regression coefficient estimates.

Ridge Regression

- As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small.
- However, the second term, $\lambda\|\boldsymbol{\beta}\|_{L2}^2$, called a shrinkage penalty , is small when $\beta_1, \cdots, \beta_p$ are close to zero, and so it has the effect of shrinking the estimates of β_j towards zero.
- The tuning parameter λ serves to control the relative impact of these two terms on the regression coefficient estimates.
- Selecting a good value for λ is critical!

Ridge Solution Path



Example from ISLR

Details of Previous Figure

- In the left-hand panel, each curve corresponds to the ridge regression coefficient estimate for one of the ten variables, plotted as a function of λ.

Details of Previous Figure

- In the left-hand panel, each curve corresponds to the ridge regression coefficient estimate for one of the ten variables, plotted as a function of λ.
- The right-hand panel displays the same ridge coefficient estimates as the left-hand panel, but instead of displaying λ on the x-axis, we now display $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$, where $\hat{\beta}$ denotes the vector of least squares coefficient estimates.

Details of Previous Figure

- In the left-hand panel, each curve corresponds to the ridge regression coefficient estimate for one of the ten variables, plotted as a function of λ.
- The right-hand panel displays the same ridge coefficient estimates as the left-hand panel, but instead of displaying λ on the x-axis, we now display $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$, where $\hat{\beta}$ denotes the vector of least squares coefficient estimates.
- The notation $\|\beta\|_2$ denotes the L^2 norm of a vector, and is defined as $\|\beta\|_2 = \sqrt{\sum_{j=1}^p \beta_j^2}$.

Lasso Regression

- Disadvantage of Ridge regression
  - shrink slopes towards 0 but not exactly 0
  - final model still include all p covariates
  - does not perform variable selection

Lasso Regression

- Disadvantage of Ridge regression
    - shrink slopes towards 0 but not exactly 0
    - final model still include all p covariates
    - does not perform variable selection
- The Lasso coefficients, $\hat{\beta}_\lambda^L$, minimize the quantity

$$\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\|\boldsymbol{\beta}\|_{L1}.$$

Lasso Regression

- Disadvantage of Ridge regression
    - shrink slopes towards 0 but not exactly 0
    - final model still include all p covariates
    - does not perform variable selection
- The Lasso coefficients, $\hat{\beta}_\lambda^L$, minimize the quantity

$$\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\|\boldsymbol{\beta}\|_{L1}.$$

- **Lasso uses an L1 penalty while Ridge uses an L2 penalty.**

Lasso Regression

- As with ridge regression, the Lasso also shrinks the coefficient estimates towards zero.

Lasso Regression

- As with ridge regression, the Lasso also shrinks the coefficient estimates towards zero.
- However, the L1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.

Lasso Regression

- As with ridge regression, the Lasso also shrinks the coefficient estimates towards zero.
- However, the L1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.
- Therefore, Lasso performs variable selection automatically when estimating the model

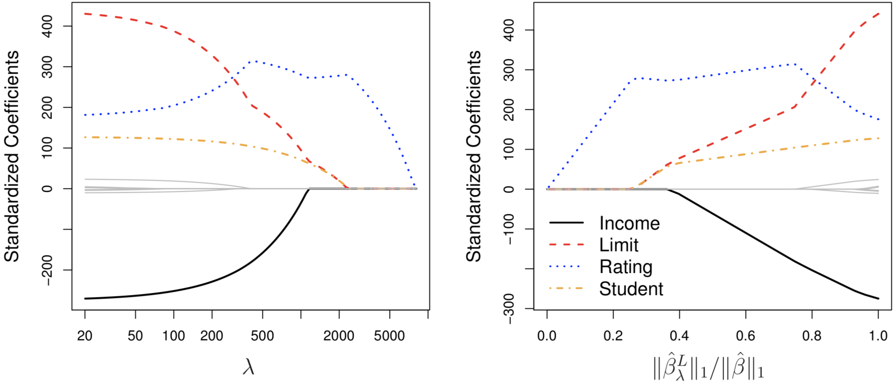
Lasso Regression

- As with ridge regression, the Lasso also shrinks the coefficient estimates towards zero.
- However, the L1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.
- Therefore, Lasso performs variable selection automatically when estimating the model
    - yields sparse models — that is, models that involve only a subset of the variables

Lasso Regression

- As with ridge regression, the Lasso also shrinks the coefficient estimates towards zero.
- However, the L1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.
- Therefore, Lasso performs variable selection automatically when estimating the model
    - yields sparse models — that is, models that involve only a subset of the variables
- Same as in ridge regression, selecting a good value of λ for the lasso is critical!

Lasso Solution Path



Example from ISLR

The Variable Selection Property of the Lasso

Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero?

The Variable Selection Property of the Lasso

Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero?

$$\hat{\beta}^{\boldsymbol{R}} = \arg\min \left[\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right]$$

this is equivalent to an optimization with constrain:

$$\text{minimize}_{\beta} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \qquad \text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq s$$

i.e. there is a 1-1 correspondence of λ and s that produce the same solution!

The Variable Selection Property of the Lasso

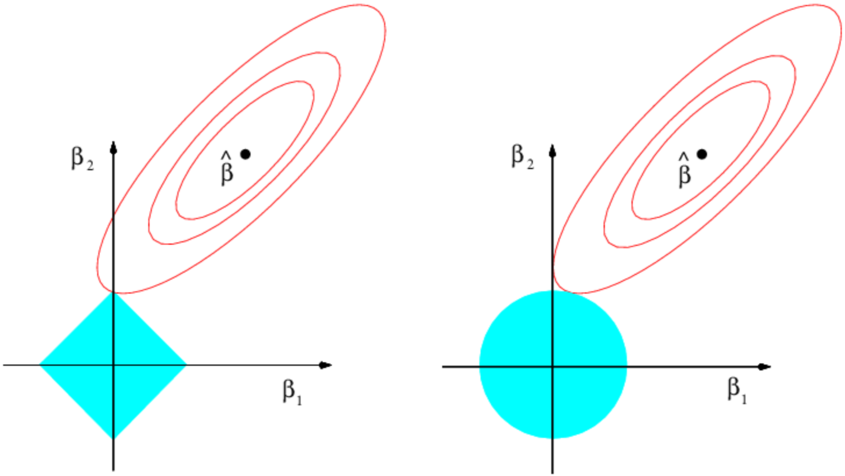
Use two covariates as example for visualization purpose:

$$\text{minimize}_\beta \sum_{i=1}^{n} (y_i - \beta_1 x_{i1} - \beta_2 x_{i2})^2$$

Ridge penalty: $\lambda(\beta_1^2 + \beta_2^2) \implies$ min RSS subject to $\beta_1^2 + \beta_2^2 \leq s$

Lasso penalty: $\lambda(|\beta_1| + |\beta_2|) \implies$ min RSS subject to $|\beta_1| + |\beta_2| \leq s$

The Intuition



What about Inference

- Notice the Ridge and Lasso solutions are no longer MLE
  - no more sampling distributions of $\hat{\beta}^{R}$ or $\hat{\beta}^{L}$
  - no more p-values and CI ???

What about Inference

- Notice the Ridge and Lasso solutions are no longer MLE
  - no more sampling distributions of $\hat{\beta}^{R}$ or $\hat{\beta}^{L}$
  - no more p-values and CI ???
- Lasso selected variables have nothing to do with statistical significance!
  - in practice of course there is still some agreement
  - cares more about prediction than what is the “true” model

What about Inference

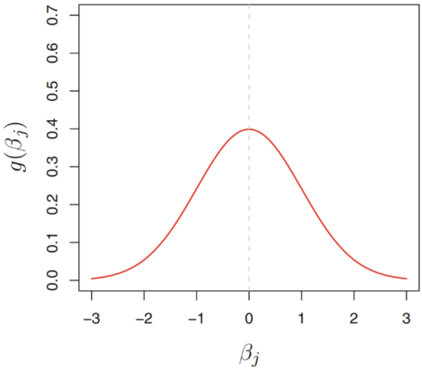
- Notice the Ridge and Lasso solutions are no longer MLE
  - no more sampling distributions of $\hat{\beta}^R$ or $\hat{\beta}^L$
  - no more p-values and CI ???
- Lasso selected variables have nothing to do with statistical significance!
  - in practice of course there is still some agreement
  - cares more about prediction than what is the "true" model
- Post selection inference is an active area of research:
  - can we recover true signals if we have infinity amount of data
  - what if number of variables increases with sample size

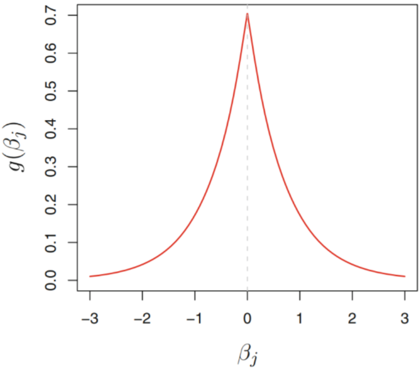
Connection to Bayesian Methods

Solutions are posterior mode with corresponding prior on β:

- Ridge prior: Gaussian
  $$g(\beta_j) = \frac{1}{\sqrt{2\pi\tau}} \exp\left(\frac{\beta_j^2}{2\tau^2}\right)$$

- Lasso prior: Laplace
  $$g(\beta_j) = \frac{1}{2b} \exp\left(\frac{|\beta_j|}{b}\right)$$





In Practice

- Standardize covariates!
    - why?
    - most software does automatically

In Practice

- Standardize covariates!
  - why?
  - most software does automatically
- How to choose λ?
  - hyper-parameter tuning in ML
  - choose a grid of λ values, and compute validation/cross-validation error for each value of λ- choose the smallest validation error

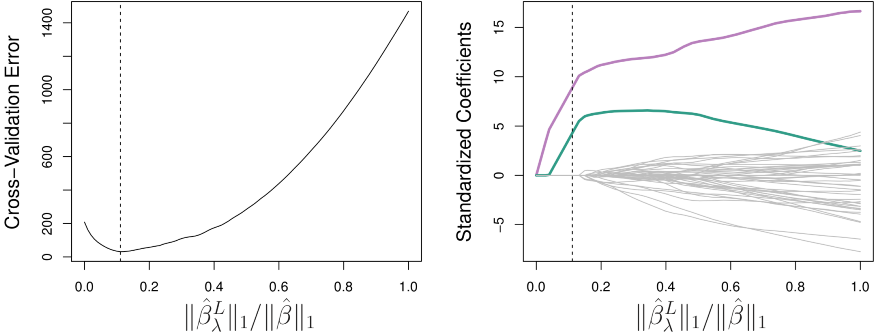
In Practice

- Standardize covariates!
  - why?
  - most software does automatically
- How to choose λ?
  - hyper-parameter tuning in ML
  - choose a grid of λ values, and compute validation/cross-validation error for each value of λ- choose the smallest validation error
- Ridge or Lasso?
  - neither will universally dominate the other
  - Lasso wins when true regression function is sparse!

In Practice

- Standardize covariates!
  - why?
  - most software does automatically
- How to choose λ?
  - hyper-parameter tuning in ML
  - choose a grid of λ values, and compute validation/cross-validation error for each value of λ- choose the smallest validation error
- Ridge or Lasso?
  - neither will universally dominate the other
  - Lasso wins when true regression function is sparse!
- Can be applied to any model fitting using optimization:
  - GLM, spline models, GAM...
  - all parametric ML models...

Lasso with CV



Example from ISLR

Outline

Polynomial Regression

- Introduce nonlinearity by adding polynomial terms

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

Polynomial Regression

- Introduce nonlinearity by adding polynomial terms

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

- Degree of the polynomial controls the flexibility of the model
  - degree of freedom $\approx$ number of parameters

Polynomial Regression

- Introduce nonlinearity by adding polynomial terms

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

- Degree of the polynomial controls the flexibility of the model
  - degree of freedom $\approx$ number of parameters
- How to select the degree of polynomial?
  - Stat approach - ANOVA test to compare nested models
  - ML approach - treat as a tuning parameter

Piecewise Polynomials

- Instead of a single polynomial in X over its whole domain, we can rather use different polynomials in regions defined by knots

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

Piecewise Polynomials

- Instead of a single polynomial in X over its whole domain, we can rather use different polynomials in regions defined by knots

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

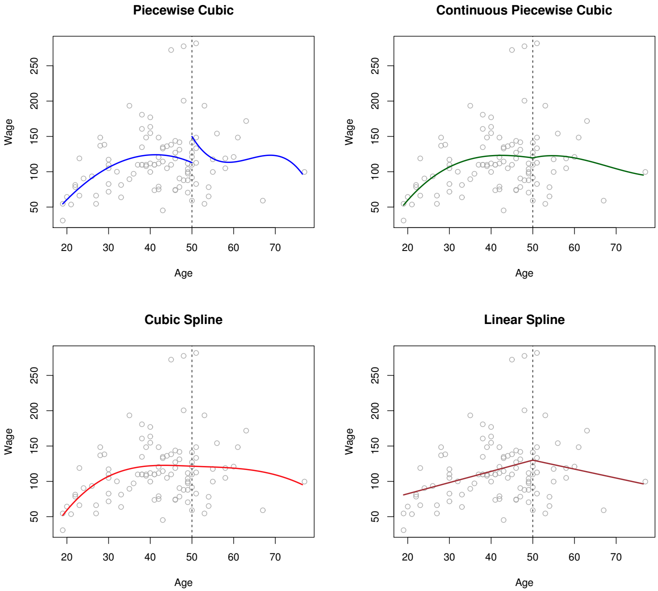
- Better to add constraints to the polynomials, e.g. continuity.

Piecewise Polynomials

- Instead of a single polynomial in X over its whole domain, we can rather use different polynomials in regions defined by knots

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

- Better to add constraints to the polynomials, e.g. continuity.
- Splines have the “maximum” amount of continuity
  - piece together local polynomials smoothly


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

Cubic Splines

- A cubic spline with knots at ξ_k, $k = 1, 2, \cdots, K$ is a piecewise cubic polynomial with continuous derivatives up to order 2 at each knot.

Cubic Splines

- A cubic spline with knots at ξ_k, $k = 1, 2, \cdots, K$ is a piecewise cubic polynomial with continuous derivatives up to order 2 at each knot.
- We can represent this model with truncated power basis functions:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i$$

where the b_k are basis functions

Cubic Spline Basis

- The b_k are basis functions

$$
\begin{aligned}
b_1(x_i) &= x_i \\
b_2(x_i) &= x_i^2 \\
b_3(x_i) &= x_i^3 \\
b_{k+3}(x_i) &= (x_i - \xi_k)_+^3, \quad k = 1, 2, \cdots, K
\end{aligned}
$$

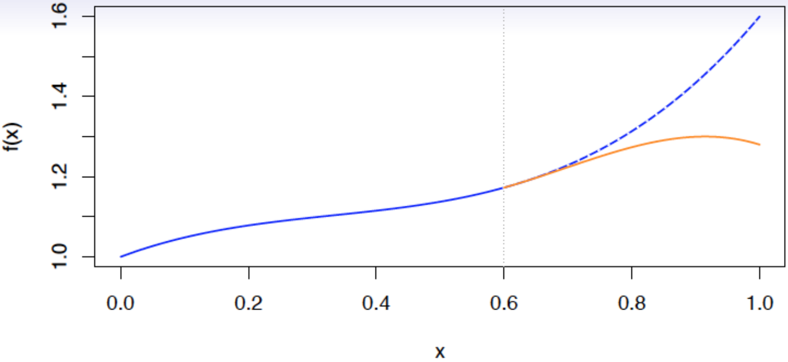
Cubic Spline Basis

- The b_k are basis functions

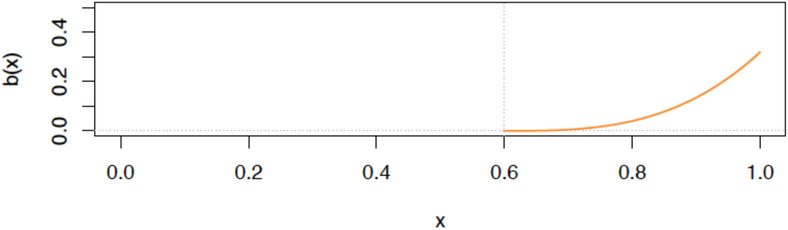
$$
\begin{aligned}
b_1(x_i) &= x_i \\
b_2(x_i) &= x_i^2 \\
b_3(x_i) &= x_i^3 \\
b_{k+3}(x_i) &= (x_i - \xi_k)_+^3, \quad k = 1, 2, \cdots, K
\end{aligned}
$$

- where

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



f(x)
1.0
1.2
1.4
1.6
0.0
0.2
0.4
0.6
0.8
1.0
x



b(x)
0.0
0.2
0.4
0.0
0.2
0.4
0.6
0.8
1.0
x

More Constrains

- Natural Cubic Splines
  - extrapolates linearly beyond the boundary knots
  - cubic spline with K knots $= K + 4$ df
  - natural spline with K knots $= K$ df

More Constrains

- Natural Cubic Splines
  - extrapolates linearly beyond the boundary knots
  - cubic spline with K knots $= K + 4$ df
  - natural spline with K knots $= K$ df
- How many knots and where?
  - choose number of knots = choose flexibility
  - some ad-hoc: fixed intervals, percentiles of X, etc.

More Constrains

- Natural Cubic Splines
  - extrapolates linearly beyond the boundary knots
  - cubic spline with K knots $= K + 4$ df
  - natural spline with K knots $= K$ df
- How many knots and where?
  - choose number of knots $=$ choose flexibility
  - some ad-hoc: fixed intervals, percentiles of X, etc.
- Smoothing splines
  - choose knot at each data point x_i
  - add smoothing penalty to control df

Smoothing Splines

- Consider a regression model $y_i = g(x_i) + \epsilon_i$ where we solve for:

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

Smoothing Splines

- Consider a regression model $y_i = g(x_i) + \epsilon_i$ where we solve for:

$$\hat{g} = \underset{g \in \mathbb{S}}{\arg\min} \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt \right)$$

- Minimize RSS tries to make $g(x)$ match the data at each x_i
  - notice g is restricted to be smooth

Smoothing Splines

- Consider a regression model $y_i = g(x_i) + \epsilon_i$ where we solve for:

$$\hat{g} = \underset{g \in \mathbb{S}}{\arg\min} \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt \right)$$

- Minimize RSS tries to make $g(x)$ match the data at each x_i
  - notice g is restricted to be smooth
- Add a roughness penalty to control how wiggly $g(x)$ is

Smoothing Splines

- Consider a regression model $y_i = g(x_i) + \epsilon_i$ where we solve for:

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

- Minimize RSS tries to make $g(x)$ match the data at each x_i
  - notice g is restricted to be smooth
- Add a roughness penalty to control how wiggly $g(x)$ is
  - $\lambda \to 0$?

Smoothing Splines

- Consider a regression model $y_i = g(x_i) + \epsilon_i$ where we solve for:

$$\hat{g} = \underset{g \in \mathbb{S}}{\arg\min} \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt \right)$$

- Minimize RSS tries to make $g(x)$ match the data at each x_i
  - notice g is restricted to be smooth
- Add a roughness penalty to control how wiggly $g(x)$ is
  - $\lambda \to 0$?
  - $\lambda \to \infty$?

Smoothing Splines

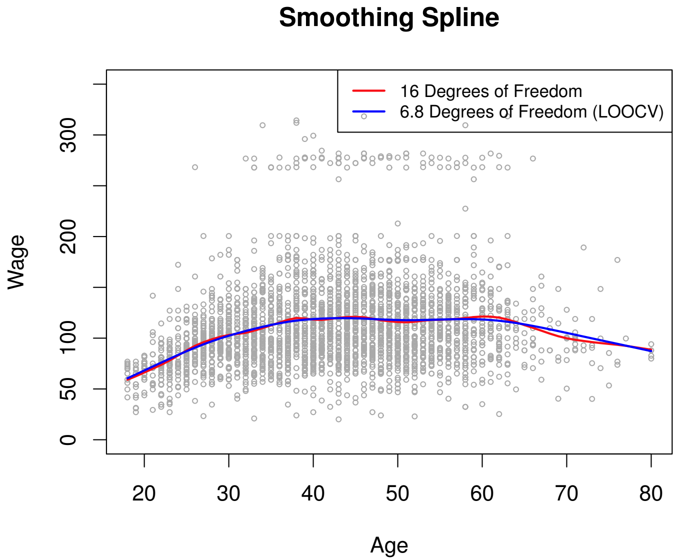
- The solution is a natural cubic spline, with a knot at every unique value of x_i

Smoothing Splines

- The solution is a natural cubic spline, with a knot at every unique value of x_i
- The roughness penalty controls the df!
    - df $<<$ number of knots
    - avoid the knot-selection issue
    - a single λ to be chosen

Smoothing Splines

- The solution is a natural cubic spline, with a knot at every unique value of x_i
- The roughness penalty controls the df!
  - df $<<$ number of knots
  - avoid the knot-selection issue
  - a single λ to be chosen
- Most software can specify df rather than λ
  - ML approach: treat λ as a tuning parameter
  - same regularization idea


Smoothing Spline
16 Degrees of Freedom
6.8 Degrees of Freedom (LOOCV)
Wage
Age
0
50
100
200
300
20
30
40
50
60
70
80

Generalized Additive Models (GAM)

- Allow nonlinearity in GLM but still additive in covariates:

$$g(\mathbb{E}[Y|X]) = \beta_0 + f_1(X_{i1}) + f_2(X_{i2}) + \cdots + f_p(X_{ip})$$

Generalized Additive Models (GAM)

- Allow nonlinearity in GLM but still additive in covariates:

$$g(\mathbb{E}[Y|X]) = \beta_0 + f_1(X_{i1}) + f_2(X_{i2}) + \cdots + f_p(X_{ip})$$

  - f_j can be linear, polynomial, spline...
  - nonlinear terms only for numerical covariates
- Coefficients no longer interpretable
  - fitted function values are (partial plot)

Generalized Additive Models (GAM)

- Allow nonlinearity in GLM but still additive in covariates:

$$g(\mathbb{E}[Y|X]) = \beta_0 + f_1(X_{i1}) + f_2(X_{i2}) + \cdots + f_p(X_{ip})$$

  - f_j can be linear, polynomial, spline...
  - nonlinear terms only for numerical covariates
- Coefficients no longer interpretable
  - fitted function values are (partial plot)
- Relax additive assumption?
  - bivariate smoothers
  - low-order interactions

Outline

1. Beyond Linearity
2. Regularization
3. Splines and GAM
4. Nonparametric Models
5. Code Demo

Kernel Density Estimate

- Nonparametric method to estimate a density function:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x - X_i}{h}\right)$$

  - K is the kernel function: uniform, triangular, Gaussian...
  - h is the bandwidth

Kernel Density Estimate

- Nonparametric method to estimate a density function:

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{x - X_i}{h}\right)$$

  - K is the kernel function: uniform, triangular, Gaussian...
  - h is the bandwidth
- "Smoothed out" histogram
  - converges faster

Kernel Density Estimate

- Nonparametric method to estimate a density function:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x - X_i}{h}\right)$$

  - K is the kernel function: uniform, triangular, Gaussian...
  - h is the bandwidth
- "Smoothed out" histogram
  - converges faster
- In practice, need to choose a kernel and a bandwidth
  - some asymptotic guideline
  - software default choice

Kernel Regression

- Want to estimate the regression function as the conditional mean:

$$f(x) = \mathbb{E}[Y|X = x]$$

with observed data (x_i, y_i) for $i = 1, 2, \cdots n$.

Kernel Regression

- Want to estimate the regression function as the conditional mean:

$$f(x) = \mathbb{E}[Y|X = x]$$

with observed data (x_i, y_i) for $i = 1, 2, \cdots n$.

- Kernel regression with kernel K and bandwidth h:

$$\hat{f}(x) = \sum_{i=1}^{n} \omega_i(x) y_i, \quad \text{where} \quad \omega_i(x) = \frac{K(\frac{x-x_i}{h})}{\sum_{k=1}^{n} K(\frac{x-x_k}{h})}$$

Kernel Regression

- Want to estimate the regression function as the conditional mean:

$$f(x) = \mathbb{E}[Y|X = x]$$

with observed data (x_i, y_i) for $i = 1, 2, \cdots n$.

- Kernel regression with kernel K and bandwidth h:

$$\hat{f}(x) = \sum_{i=1}^{n} \omega_i(x) y_i, \quad \text{where} \quad \omega_i(x) = \frac{K(\frac{x-x_i}{h})}{\sum_{k=1}^{n} K(\frac{x-x_k}{h})}$$

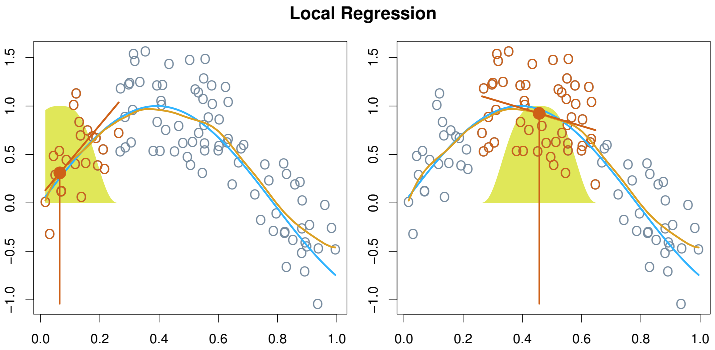
- K and h play the similar role in KDE

Local Regression

- Locally Weighted Scatterplot Smoothing
  - fit linear regression locally by weighted least squares
  - (weighted) nearest neighbor regression as a special case

Local Regression

- Locally Weighted Scatterplot Smoothing
  - fit linear regression locally by weighted least squares
  - (weighted) nearest neighbor regression as a special case



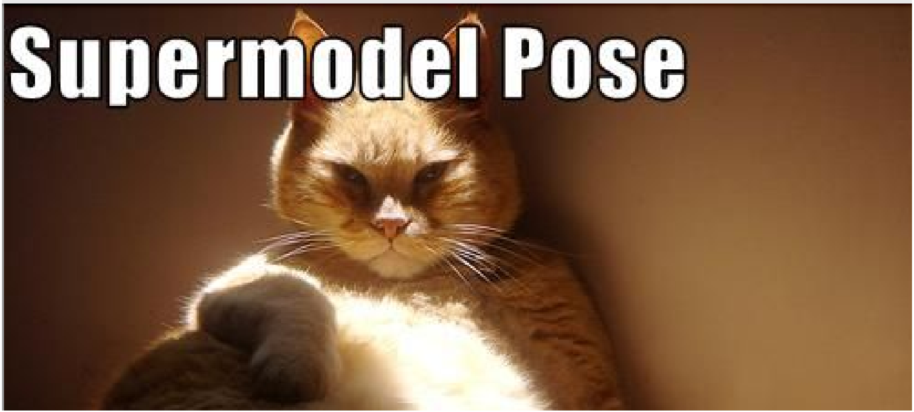
Example from ISLR

Summary

- With GAM in your statistics toolbox, you are able to:
  - model any type of **response variable** in the GLM family:
    - continuous: Gaussian, Gamma, Beta
    - counts: Poisson, Negative Binomial
    - binary: Binomial
    - categorical: Multinomial
  - include both numerical and categorical **predictors**
  - include a **mix of linear and nonlinear** effects
    - how do you decide?
  - include interaction terms to relax additive assumption
    - interpretation is key
  - apply regularization with Ridge or Lasso penalty

Which means...

You are now officially a supermodeler. Bravo!!!



Outline