Natural Cubic Spline

- A natural cubic spline is linear at the boundary
  - Reduces variance near the boundaries.

- A natural cubic spline with K (interior) knots has _______ df

- What is a natural cubic spline with 0 knots?

Natural Cubic Spline

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

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \cdots + \beta_{K+1} b_{K+1}(x_i) + \varepsilon_i$$

- Basis (Optional): 1, x, $g(x, \xi_0, \xi_K, \xi_{K+1})$, $g(x, \xi_1, \xi_K, \xi_{K+1})$, …, $g(x, \xi_{K-1}, \xi_K, \xi_{K+1})$ i.e. $K + 2$ df
    - $g()$: very complex, but known and can be written out explicitly
    - R can auto. Implement

- R func: ns()

Natural Cubic Spline

- K: #knots

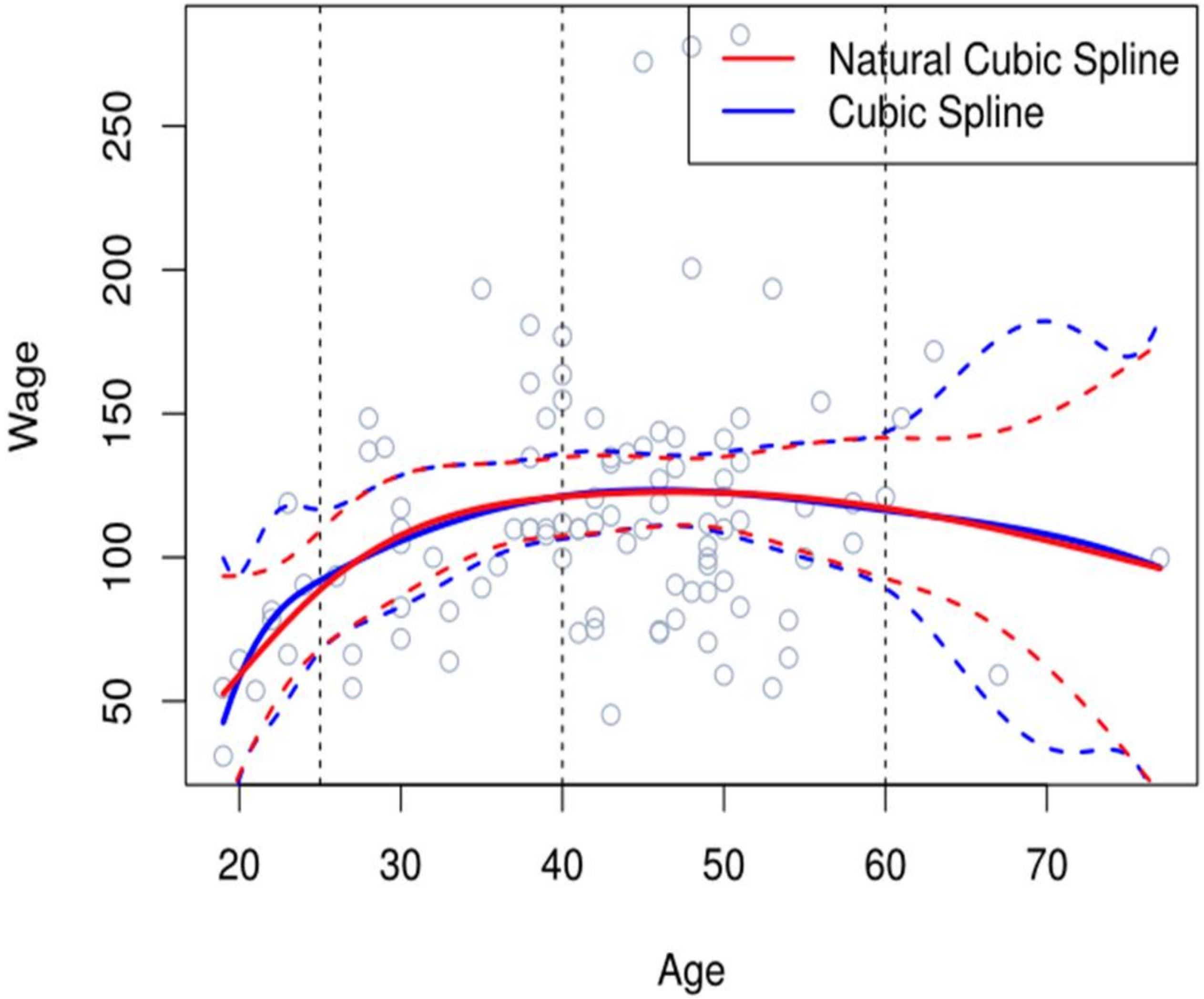
Intercept NOT counted as 1 df

	Our df (Real df)	df in R
Cubic Spline	4 + K	3 + K
Natural Cubic Spline	2 + K	1 + K

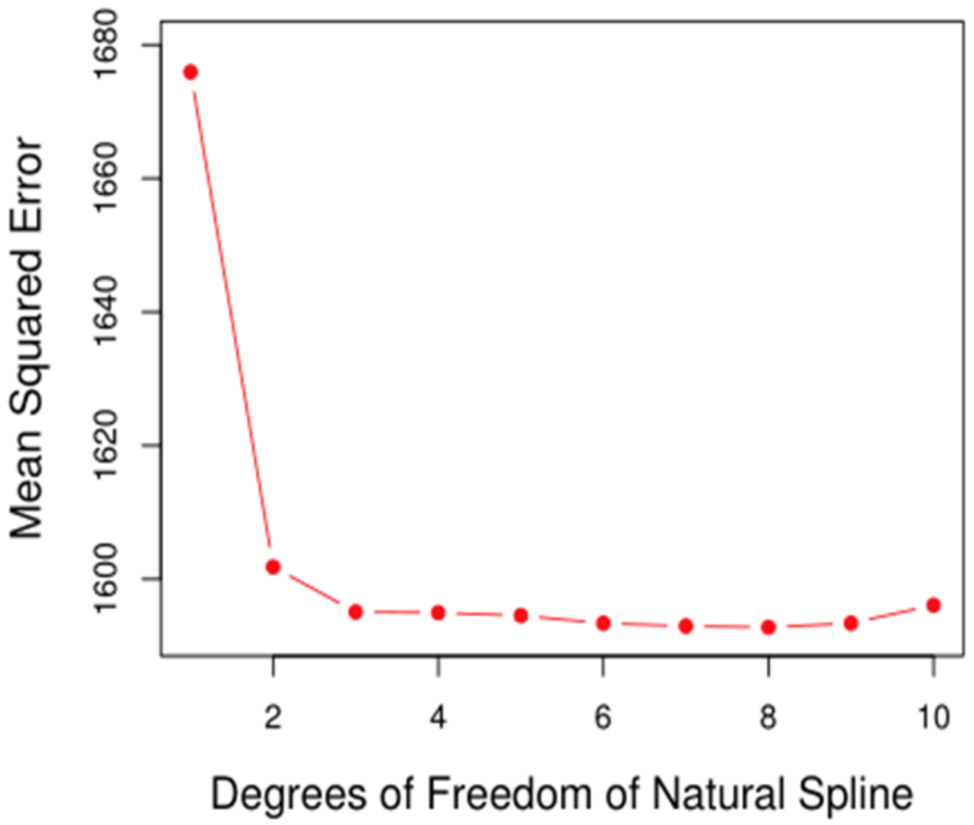
Natural Cubic Spline

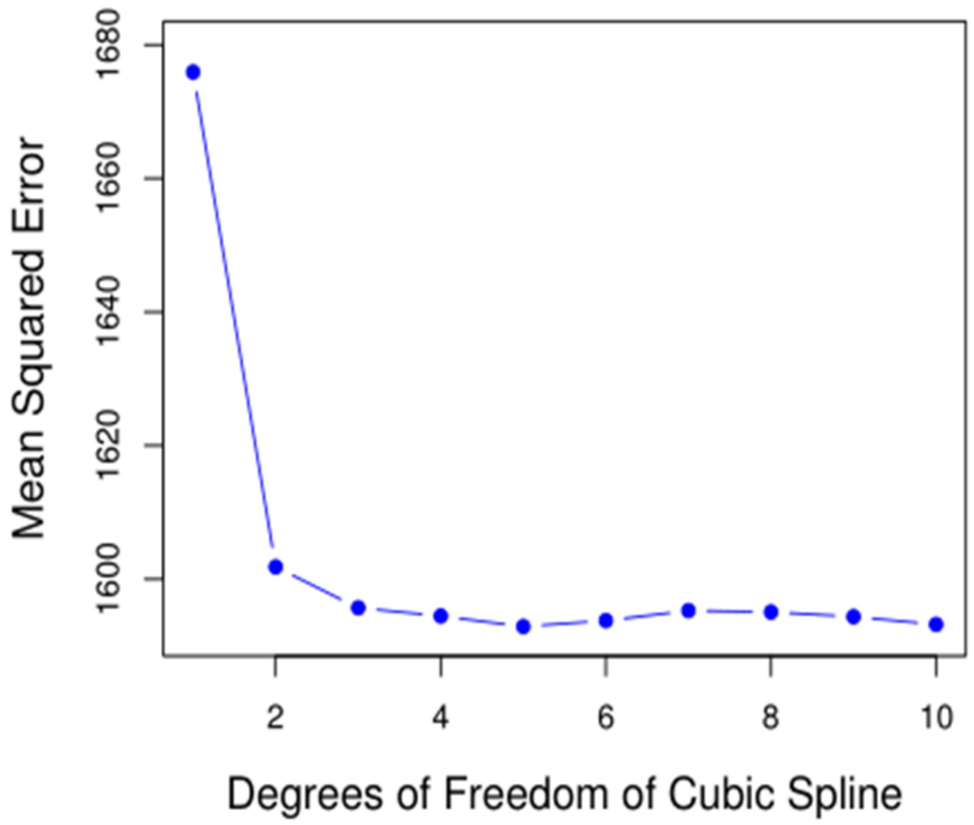
- Look at some code examples

Natural Cubic Spline Illustration



CV-error vs df



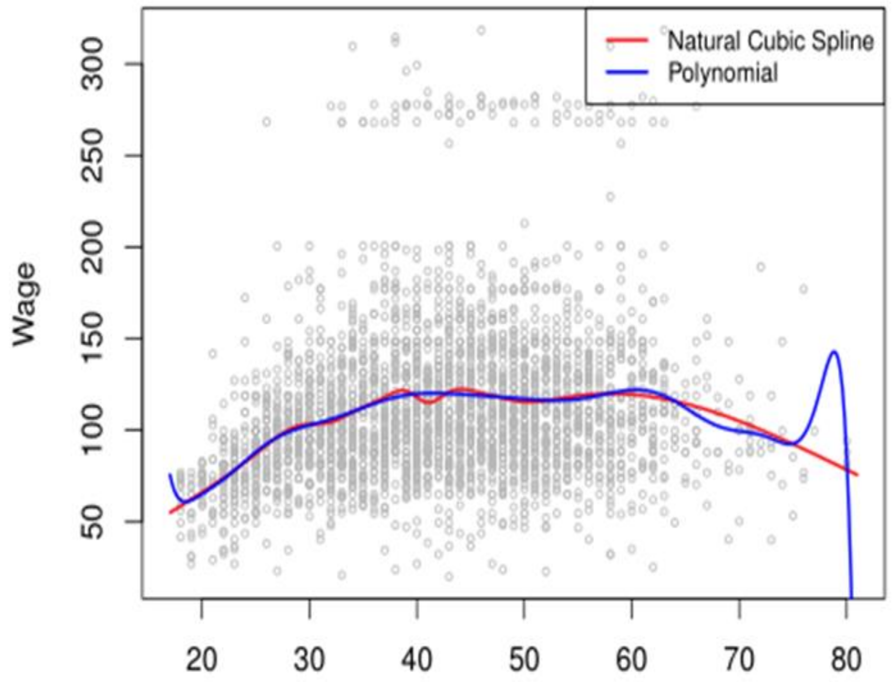


Note: df in R
How many knots?
Which model will you select?

Comparison to Polynomial Regression

- Regression splines often outperforms polynomial regression.
  - Polynomial must use a high degree to produce flexible fits
  - Spline introduces flexibility by increasing #knots but keeping the degree fixed
- As a result, spline is more stable
- Also for splines, can place more knots over rapid-changing regions / fewer knots on stable regions

Natural cubic spline vs Polynomial regression (both have df = 15, on the wage dataset)



Outline

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

Outline

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

Smoothing Spline

- Regression splines: Specifying a set of knots and producing a sequence of basis functions.
- Smoothing spline: A different approach.

$$min_f \quad RSS = \sum_{i=1}^{N} (y_i - f(x_i))^2$$

- If we minimize the above and do not place constraints on f
  - We can interpolate all points and make RSS zero → Overfitting!
  - Add constraints s.t. *f* is NOT too volatile
- Goal: Find function *f* such that RSS is small and *f* is "NOT too wiggly".

Smoothing Spline

- Among all functions *f* with two continuous derivatives, find one that minimizes the penalized residual sum of squares.

$$RSS(f,\lambda) = \sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt$$

- "Loss + Penalty" (Similar to Lasso and Ridge)
- Loss: $\sum_{i=1}^{n}\left(y_i - f(x_i)\right)^2$
- Penalty: $\int f''(t)^2 dt$ penalizes the variability in *f*.

What does $f'(t), f''(t)$ represent?
Why penalize f'' (instead of f or f')?

Smoothing Spline

- The role of λ

$$min_f \; RSS(f, \lambda) = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt$$

What happens if $\lambda \to 0$? $\lambda \to \infty$?

Bias-variance tradeoff

- λ small vs λ large

Smoothing Spline

- Solution to the minimization problem

$$min_f \; RSS(f,\lambda) = \sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda \int (f''(t))^2 dt$$

Result

The solution f is a **natural cubic spline** with knots at all training observations x_i, $i = 1,2,\ldots,N$

Effective Degree of freedom

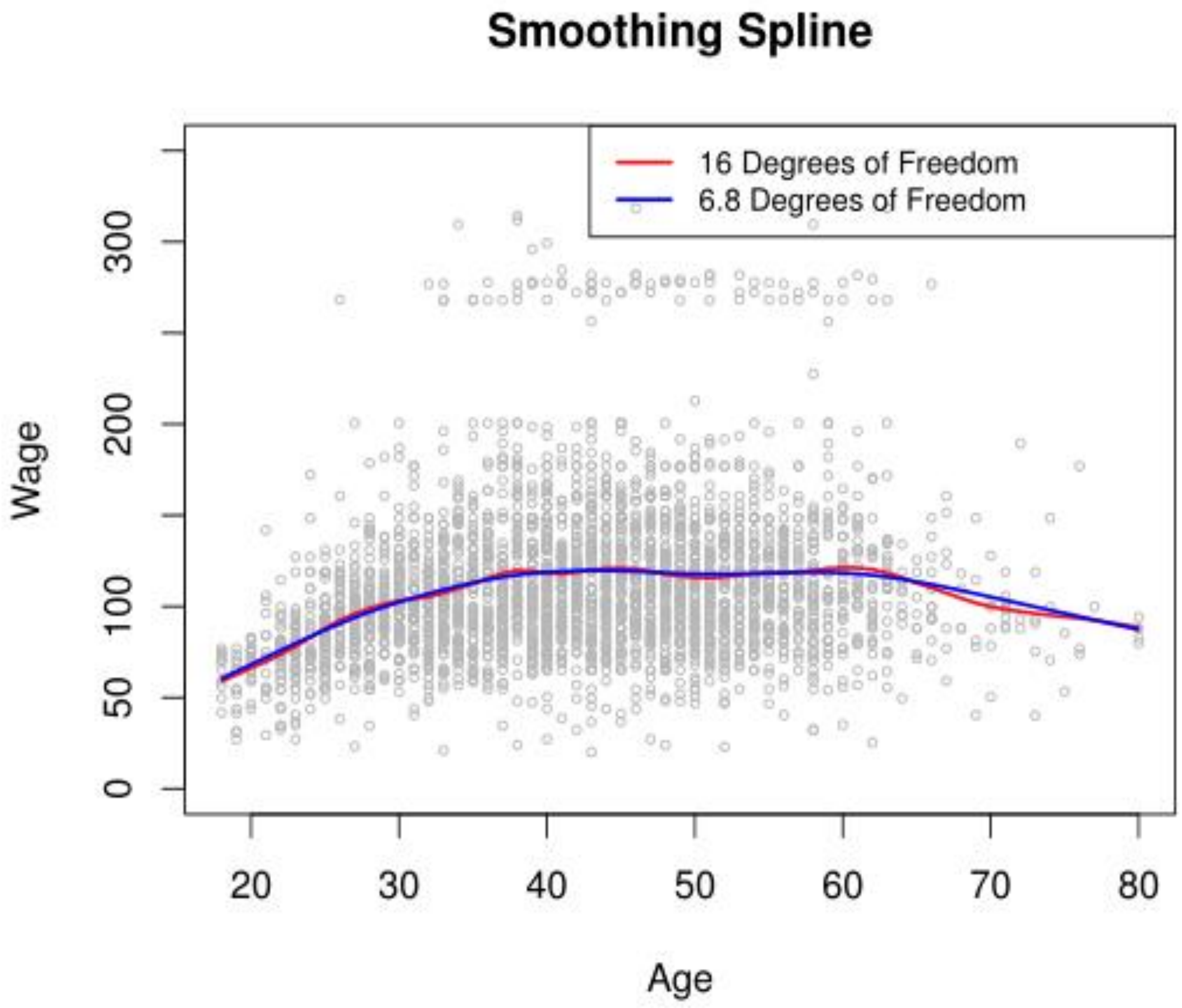
- Abstract notion, just think of it as a mapping from λ
- $\lambda \to 0$, $df_\lambda \to N$ (# obs, can interpolate with a func with this df)
- $\lambda \to \infty$, $df_\lambda \to 2$ (linear model)
- The higher df_λ is, the more flexible the smoothing spline is.
- How to select df_λ?

Smoothing spline fitted to Wage dataset

Red curve: specify 16 df
Blue curve: best λ from CV, 6.8 df

Little difference between the two curves, except red one (16 df) wigglier

Which curve will you choose?



Smoothing spline for Regression

- Look at some code examples

Outline

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

Nonparametric Logistic Regression

- Model

$$log \frac{P(G=1|X)}{P(G=0|X)} = f(X)$$

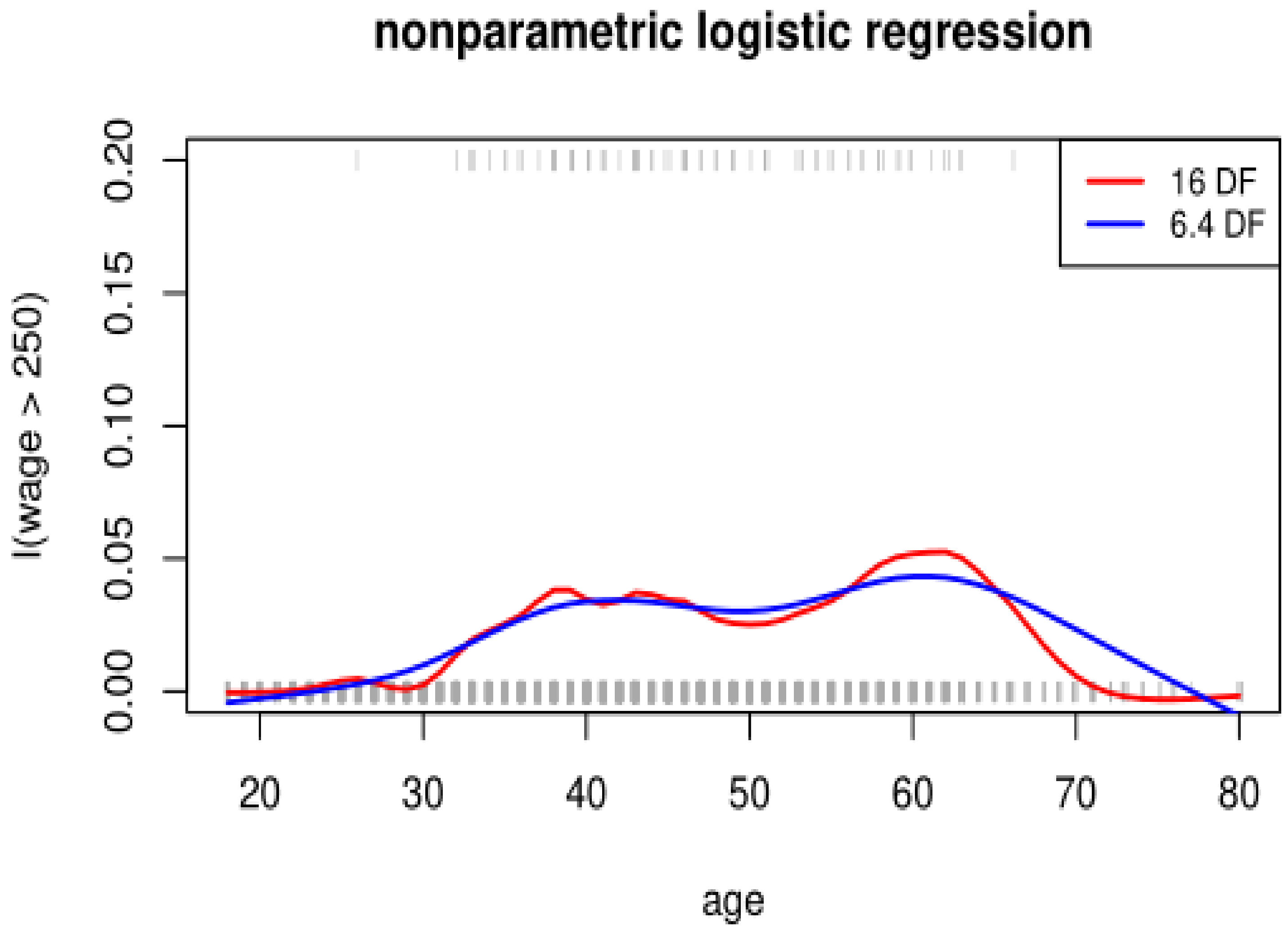
- Consider the penalized log-likelihood criterion

$$max_f \sum_{i=1}^{N} \left\{ g_i \log\big(p(x_i)\big) + (1-g_i)\log\big(1-p(x_i)\big) - \frac{1}{2}\lambda \int (f''(t))^2 dt \right\}$$

where

$$p(x_i) = \frac{\exp(f(x_i))}{1+\exp(f(x_i))}$$

Illustration



Nonparametric Logistic Regression

- Look at some code examples

Questions

- We have learnt the following methods
    - Polynomial regression
    - Step functions
    - Cubic splines
    - Natural cubic splines
    - Smoothing splines
- Given a new dataset, which method should you use?

Outline

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

Generalized Additive Models: Motivation

- Previously, methods applied to a single predictor (age)
- Generalize to multiple predictors: GAM
  - General framework for allowing nonlinear funcs of each predictors
  - Analogous: simple linear regression → multiple linear regression
  - Assumption: effect of interaction term NOT significant

Generalized Additive Models

Data $(x_i, y_i), i = 1,2, \dots, n$, where $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$

- Multiple Linear Regression

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

- GAMs for Regression Problems

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

Generalized Additive Models

- Example

$$wage = \beta_0 + f_1(year) + \ f_2(age) + education$$

- Or

$$log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + f_1(year) + f_2(age) + education$$

where

$$p(x) = p(wage > 250|year, age, education)$$

Generalized Additive Models

- Example

$$wage = \beta_0 + f_1(year) + \ f_2(age) + education$$

- f_j's can be the methods we've learnt (smoothing splines, natural splines, etc)
- Fit the model using least squares: easy
  - Natural splines:
    - pre-construct the basis funcs
    - Entire model: big regression onto spline basis and dummy variables (for education)
  - Smoothing splines:
    - More complicated, but has efficient algorithm to solve it (Backfitting)

Generalized Additive Models

- Look at some code examples

Generalized Additive Models

- Pros
  - Easy to implement, auto. fit nonlinear models
  - More accurate predictions due to nonlinear fits
  - Model additive: easily interpreted
    - Holding other predictors constant, effect of one predictor on the outcome
- Cons
  - Assumption on Additivity, cannot model interaction effects
- Fully general models will be covered later in the course (e.g. Random Forest)
  - GAM: compromise between linear and fully general models

Method	R func
Polynomial Reg	lm(wage~poly(age,4), data=Wage) glm(I(wage>250)~poly(age,4),data=Wage,family=binomial)
Step Func	lm(wage~cut(age,4), data=Wage) glm(I(wage>250)~poly(age,4),data=Wage,family=binomial)
Cubic Spline	lm(wage~bs(age,knots=c(25,40,60)),data=Wage) lm(wage~bs(age,df=6)),data=Wage)
Natural Cubic Spline	lm(wage~ns(age,df=4),data=Wage)
Smoothing Spline	smooth.spline(age,wage,df = 16) smooth.spline(age,wage,cv = TRUE)
Generalized Additive Model	lm(wage~ns(year,4)+ns(age,5)+education,data = Wage) gam(wage~s(year,4)+s(age,5)+education,data = Wage) (Note: s() is smoothing spline, need to use gam() if additive model has smoothing spline func)

End