STAT 432: Basics of Statistical Learning

Regression Splines

Shiwei Lan, Ph.D. <shiwei@illinois.edu>

http://shiwei.stat.illinois.edu/stat432.html

March 15, 2019

University of Illinois at Urbana-Champaign

Outline

- From Linear to Nonlinear: Histogram Regression
- Basis Functions
- Piecewise Polynomials
- B-Splines and Natural Cubic Splines
- Smoothing Splines

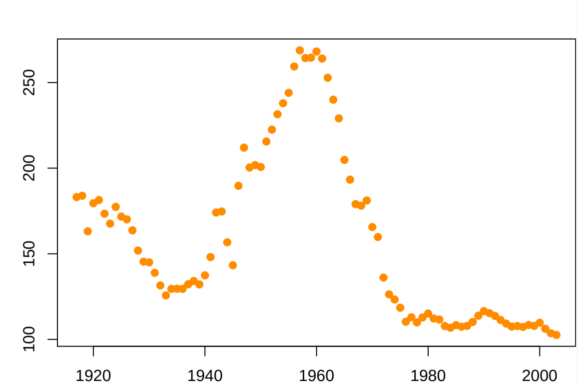
Linear vs. Nonlinear models

- Up to now, we mostly focused on linear models. Why?
  - Convenient and easy to fit
  - Easy to interpret
  - An approximation to the true underlying function $f(x)$
  - Tend not to overfit (when p is small)
- However, nothing is really perfectly linear in practice
- How to relax this restriction and fit more flexible models?

Linear vs. Nonlinear models

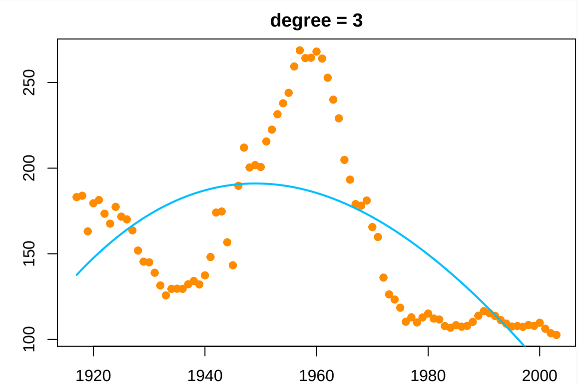
- In this lecture, we will focus on the case that only one variable is involved ($p = 1$)
- We are interested in approximating the regression function $f(x)$
- One idea is to include higher order terms, or nonlinear transformations.
- For example, x^2, x^3, $\log(x)$, $\sqrt{x}$, etc.

Birthrate Data



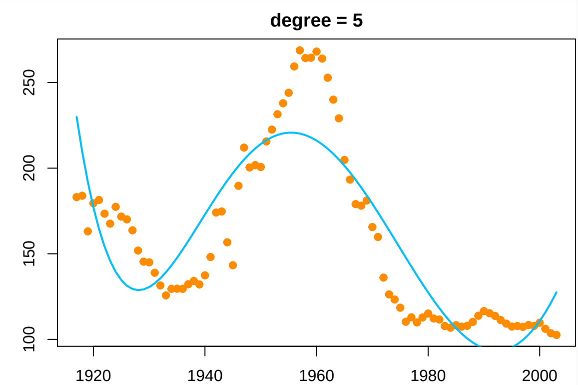
U.S. birthrates 1917 - 2003

Birthrate Data



U.S. birthrates 1917 - 2003

Birthrate Data



U.S. birthrates 1917 - 2003

Linear vs. Nonlinear models

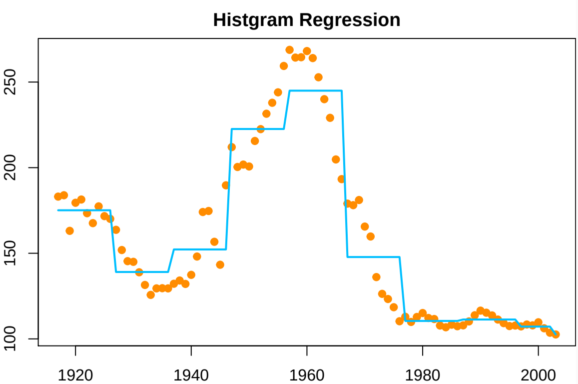
- Another idea is to estimate the regression function locally.
- A model we learned earlier was the kNN, and one essential idea is to model $f(x)$ at a local region.
- In this lecture, we will try another approach, while also motivated from a local estimation
- We will first illustrate a simple model called the histogram regression

Histogram Regression

- Suppose we observe a set of observations $\{x_i, y_i\}_{i=1}^n$, note that x_i are univariate.
- Then we can choose several "knots" on the range of x_i. This can be by either an educated decision or based on quantiles.
- Based on these knots, we can isolate the interval between two adjacent knots.
- Suppose the interval that contains a given testing point x is $\phi(x)$, then the prediction at this point is

$$\widehat{f}(x) = \frac{\sum_{i=1}^n Y_i \ I\{X_i \in \phi(x)\}}{\sum_{i=1}^n I\{X_i \in \phi(x)\}}$$

Birthrate Data



U.S. birthrates 1917 - 2003

Histogram Regression

- The histogram regression is still not flexible enough
- However, based on this idea of splitting into intervals and fit curves within interval, we will introduce a new concept called splines.
- First, we introduce the idea of basis functions

Basis Functions

Linear vs. Nonlinear models

- Additive model: stepping outside the linear model, lets assume that our model has the form

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

- We can consider different types of $h_m(x)$
  - $h_m(x) = x$: the original linear model (univariate)
  - $h_m(x) = x^2, x^3, \ldots$: polynomials
  - $h_m(x) = I(a_m \leq x < b_m)$: step function / histogram regression

Linear vs. Nonlinear models

- This is essential a type of feature engineering
- The approach is straight forward: We create nonlinear functions of x as features, and then fit a linear regression on this set of new features.
- These features are called the basis functions
- Hence, for this lecture, we will focus on how to construct basis functions

Piecewise Polynomials

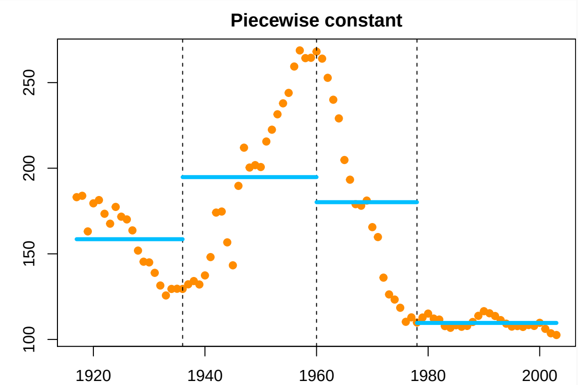
Piecewise Constant

- For example, consider the piecewise constant on four regions:

$$h_1(x) = \mathbf{1}\{x < \xi_1\}, \qquad h_2(x) = \mathbf{1}\{\xi_1 \leq x < \xi_2\},$$
$$h_3(x) = \mathbf{1}\{\xi_2 \leq x < \xi_3\}, \qquad h_4(x) = \mathbf{1}\{\xi_3 \leq x\}.$$

- ξ_1, ξ_2 and ξ_3 are called knots. For the birthrate data, we use three knots: 1936, 1960, 1978
- Because we are fitting a constant on each region, its just the mean of observations in that interval.

Birthrate Data



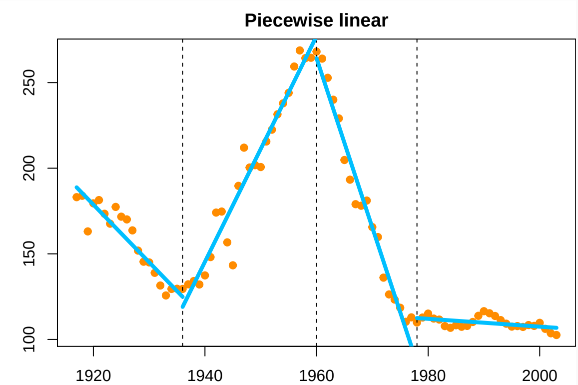
Piecewise Linear

- We can also fit a linear function at each region by considering four additional basis functions:

$$h_5(x) = x\mathbf{1}\{x < \xi_1\}, \qquad h_6(x) = x\mathbf{1}\{\xi_1 \le x < \xi_2\},$$
$$h_7(x) = x\mathbf{1}\{\xi_2 \le x < \xi_3\}, \qquad h_8(x) = x\mathbf{1}\{\xi_3 \le x\}.$$

- We can of course increase the degree, but a clear drawback is that the function is not continuous.
- How to force continuity?

Birthrate Data



Force Continuity

- If we set the constraint that

$$f(\xi_1^-) = f(\xi_1^+)$$

  to force continuity.
- Note that our function f has 8 basis, 4 for linear and 4 for slopes.
- This implies

$$\beta_1 + \xi_1\beta_5 = \beta_2 + \xi_1\beta_6$$

- However, solving a constrained linear model seems to be difficult. Is there a easier way to construct the basis?

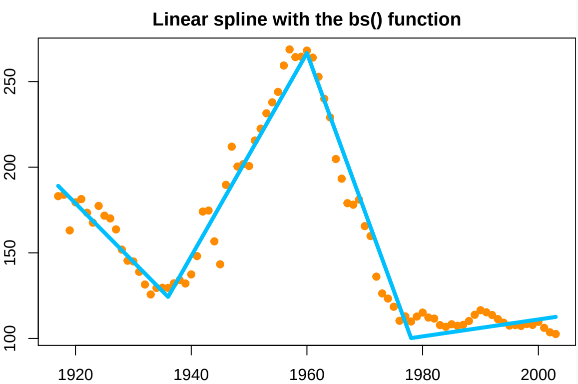
Force Continuity

- The trick is to incorporate the constrains into the basis functions:

$$h_1(x) = 1,\ h_2(x) = x,\ h_3(x) = (x - \xi_1)_+,$$
$$h_4(x) = (x - \xi_2)_+,\ h_5(x) = (x - \xi_3)_+,$$

  where $(\cdot)^+$ denotes the positive part.
- How many parameters in the original basis? — 8
- How many constrains? — 3
- Hence, essentially we only need 5 parameters.

Birthrate Data



Linear Spline

- The final model is

$$f(x) = \sum_{m=1}^{5} \beta_m h_m(x)$$

- We can then check that any linear combination of these five functions lead to
  - Continuous everywhere
  - Linear everywhere except the knots
  - Has a different slope for each region
- This can be easily done using R function bs in the package splines.

Polynomial Spline

- In general, we may need to consider
  - The number of degrees in each region
  - The number of knots
  - The locations of the knots
- Selecting the knots can be difficult
- See our R Lab examples

Cubic Splines

- A common choice is cubic splines, which uses cubic functions within each region. However, continuity of the first and second order at the knots is forced.
- For each knot ξ, we need the following 4 basis functions:

$$h_1(x) = 1,\ h_2(x) = x,\ h_3(x) = x^2,\ h_4(x) = (x-\xi)^3.$$

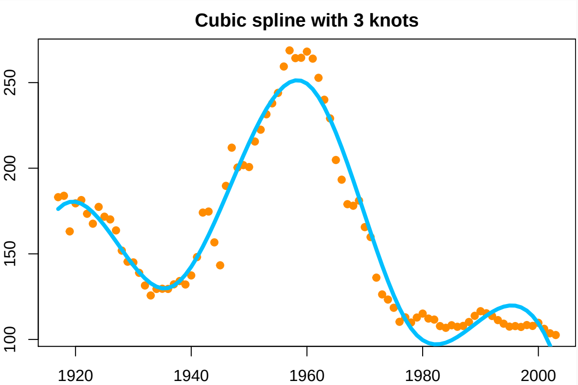
- However, the first three basis are shared by all knots.
- Cubic spline function with K knots:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{k=1}^{K} b_k (x - \xi_k)_+^3$$

- The degrees of freedom for a cubic spline:

$(\#\,\text{regions}) \times (4\,\text{per region}) - (\#\,\text{knots}) \times (3\,\text{constraints per knot})$

Birthrate Data



B-Splines and Natural Cubic Spline

B-Spline basis

- The previous definitions are known as regression splines
- An alternative (computationally more efficient) way of defining the spline basis is proposed by de Boor (1978)
- Each basis function is nonzero over at most (degree + 1) consecutive intervals
- The resulting design matrix is banded

Construct the B-Spline basis

- Create augmented knot sequence τ:

$$\tau_1 = \cdots = \tau_M = \xi_0$$
$$\tau_{M+j} = \xi_j, \quad j = 1, \ldots K$$
$$\tau_{M+K+1} = \cdots = \tau_{2M+K+1} = \xi_{K+1}$$

where ξ_0 and ξ_{K+1} are the left and right boundary points.

Construct the B-Spline basis

- Denote $B_{i,m}(x)$ the ith B-spline basis function of order m for the knot sequence τ, $m \leq M$. We recursively calculate them as follows:

$$B_{i,1}(x) = \begin{cases} 1 & \text{if} \quad \tau_i \leq x < \tau_{i+1} \\ 0 & \text{o.w.} \end{cases}$$

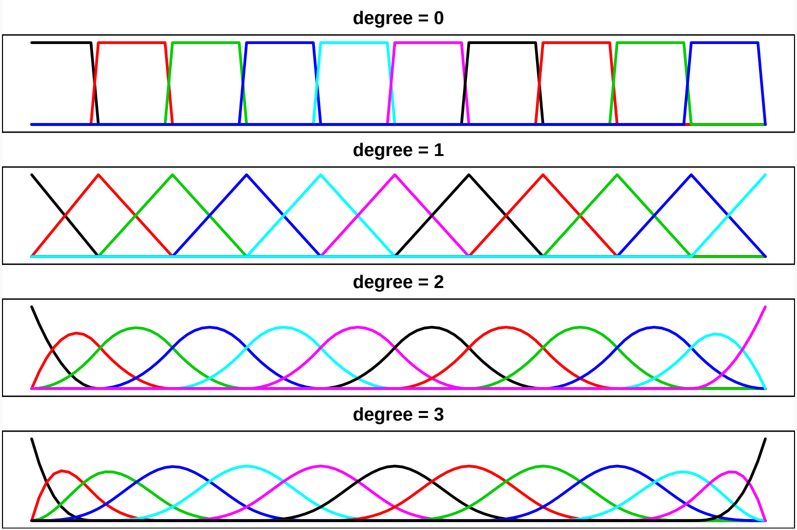
$$B_{i,m}(x) = \frac{x - \tau_i}{\tau_{i+m-1} - \tau_i} B_{i,m-1}(x) + \frac{\tau_{i+m} - x}{\tau_{i+m} - \tau_{i+1}} B_{i+1,m-1}(x)$$

Generating B-Spline basis in R

```
> library(splines)
> bs(x, df = NULL, knots = NULL, degree = 3, intercept = FALSE)
```

- df : degrees of freedom (the total number of basis)
- knots : specify knots. By default, these will be the quantiles of x
- degree : degree of piecewise polynomial, default 3 (cubic splines)
- intercept : if TRUE, an intercept is included, default FALSE
- Return a matrix of dimension $n\times$ df

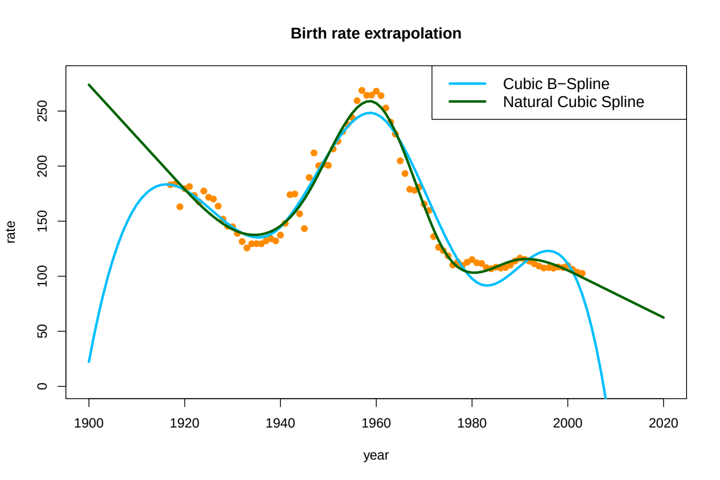
B-Spline Basis



Natural Cubic Splines

- Polynomials fit to data tends to be erratic near the boundaries, and extrapolation can be dangerous
- Natural cubic splines (NCS) forces the second and third derivatives to be zero at the boundaries, i.e., $\min(x)$ and $\max(x)$
- Hence, the fitted model is linear beyond the two extreme knots $(-\infty, \xi_1]$ and $[\xi_K, \infty)$
- The constraint frees up 4 degrees of freedom (two for each end). The degrees of freedom of NCS is just the number of knots K.

Extrapolating beyond the boundaries



Constructing Natural Cubic Splines

- Starting with a basis for cubic splines, and derive the reduced bases by imposing the boundary constraints, we obtain the basis functions

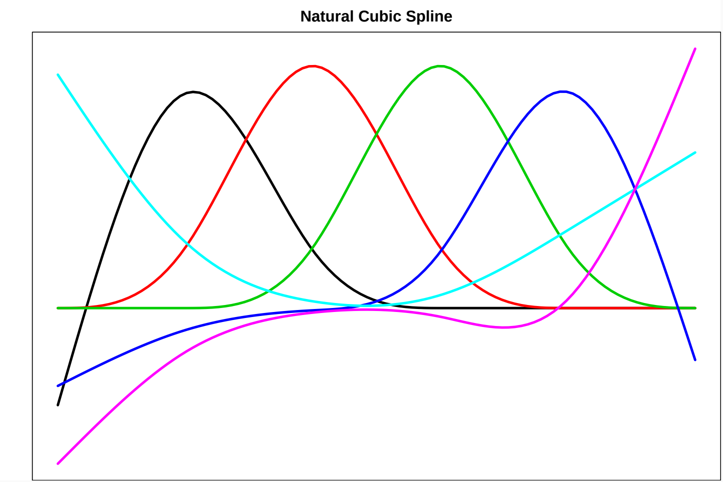
$$N_1(x) = 1,\ N_2(x) = x,\ N_{k+2}(x) = d_k(x) - d_{K-1}(x)$$

  where

$$d_k(x) = \frac{(x - \xi_k)_+^3 - (x - \xi_K)_+^3}{\xi_K - \xi_k}, \quad k = 1, \dots, K-2$$

- We can check that each of the basis functions $N_k(x)$'s has zero second and third derivatives for $x \leq \xi_1$ and $x \geq \xi_K$

NCS Basis



Generating Natural Cubic Spline basis in R

```
> library(splines)
> ns(x, df = NULL, knots = NULL, intercept = FALSE)
```

- df : degrees of freedom (the total number of basis)
- knots : specify knots. By default, these will be the quantiles of x
- intercept : if TRUE , an intercept is included, default FALSE
- Return a matrix of dimension $n\times$ df

Smoothing Splines

Smoothing Splines

- B-splines and NCS are both methods that construct a $n \times M$ basis matrix $\mathbf{F}$, and then model the outcome using a linear regression on $\mathbf{F}$.
- Inevitably, we need to select the order of the spline, the number of knots (AIC, BIC, CV) and even the location of knots (difficult)
- Is there a method that we can select the number and location of knots automatically?

Smoothing Splines

- Smoothing Splines: Let's start with an easy but "horrible" solution, by putting knots at all the observed data points $(x_1, \dots x_n)$:

$$\mathbf{y}_{n\times 1} = \mathbf{F}_{n\times n}\boldsymbol{\beta}_{n\times 1}$$

  Instead of selecting knots, let's use ridge-type shrinkage

$$\text{minimize}_{\boldsymbol{\beta}} \ \|\mathbf{y} - \mathbf{F}\boldsymbol{\beta}\|^2 + \lambda\boldsymbol{\beta}^\mathsf{T}\Omega\boldsymbol{\beta}$$

  where Ω will be defined later and λ can be chosen by CV or GCV.
- In fact, the solution can be derived from a different aspect

Roughness Penalty Approach

- Let $W^2([a,b])$ be a second-order Sobolev space on $[a,b]$, equipped with L_2 norm:

$$\left\{g : g, g' \text{ are absolutely continuous and } \int_a^b \left[g''(x)\right]^2 dx < \infty\right\}$$

- $W^2[a,b]$ is an infinitely-dimensional function space
- Global polynomial functions and cubic spline (including NCS) functions are in $W^2[a,b]$.
- Find the best function in $W_2[a,b]$ to approximate f

Roughness Penalty Approach

- Suppose, instead of using splines to approximate the function f, we do a penalized residual sum of squares

$$\text{RSS}(g, \lambda) = \frac{1}{n} \sum_{i=1}^{n} \big(y_i - g(x_i)\big)^2 + \lambda \int_a^b [g''(x)]^2 dx$$

- The first term measures the closeness between the fitted model, $g(x_i)$, and the observed data y_i.
- The second term penalizes the roughness/curvature (second derivative) of the fitted function
- $\int_a^b [g''(x)]^2 dx$ is called the roughness penalty

Roughness Penalty Approach

- λ is the smoothing parameter that controls the bias-variance trade-off
- $\lambda = 0$: interpolate the data, over-fitting
- $\lambda = \infty$: $g'' \equiv 0 \Longrightarrow$ linear least-squares regression
- It turns out that the solution to the penalized residual sum of squares has to be a NCS
- This avoids the knot selection problem?

Roughness Penalty Approach

Theorem

Let the two bounds $a = \min_i x_i$ and $b = \max_i x_i$. Then, for any λ, the solution $\widehat{g}$ for the penalized residual sum of squares approach,

$$\widehat{g} = \underset{g \in W^2([a,b])}{\arg\min} \; \textit{RSS}(g, \lambda)$$

can be represented by a set of NCS basis with knots at the n observed data points $x_1, \ldots, x_n$

Proof*

Intuition: Let g be any function in $W^2[a, b]$ and $\widetilde{g}$ be a function represented by NCS basis with

$$g(x_i) = \widetilde{g}(x_i), \quad i = 1, \ldots, n.$$

Note: We can always find such $\widetilde{g}$ since the NCS consists of n basis. Then, **IF** we can show

$$\int \left[g''(x)\right]^2 dx \geq \int \left[\widetilde{g}''(x)\right]^2 dx,$$

Then the NCS “representation” of g has a smaller penalty, hence, we will always prefer the NCS solution.

Proof*

Hence, its left to show (with some abbreviations) that

$$\int g''^2 dx \geq \int \widetilde{g}''^2 dx.$$

We define the difference of the two solutions:

$$h(x) = g(x) - \widetilde{g}(x)$$

So $h(x_i) = 0$ for $i = 1, \ldots, n$, by the definition of $\widetilde{g}(x)$. Then

$$\begin{aligned} \int g''^2 dx &= \int \left[\widetilde{g}'' + h''\right]^2 dx \\ &= \underbrace{\int \widetilde{g}''^2 dx}_{\text{NCS Penalty}} + \underbrace{\int h''^2 dx}_{\geq 0} + \underbrace{2 \int \widetilde{g}'' h'' dx}_{?} \end{aligned}$$

Proof*

W.L.O.G., assume that x_i's are ordered. The the cross-term is

$$
\begin{aligned}
\int \widetilde{g}'' h'' dx &= \widetilde{g}'' h' \Big|_a^b - \int_a^b h' \widetilde{g}^{(3)} dx \qquad \big(\text{integration by parts}\big) \\
&= 0 - \int_a^b h' \widetilde{g}^{(3)} dx \qquad \big(\widetilde{g}''(a) = \widetilde{g}''(b) = 0\big) \\
&= -\sum_{i=1}^{n-1} \widetilde{g}^{(3)}(x_j^+) \int_{x_j}^{x_{j+1}} h' dx \qquad \big(\widetilde{g}^{(3)} \text{constant piecewise}\big) \\
&= -\sum_{i=1}^{n-1} \widetilde{g}^{(3)}(x_j^+) \big(h(x_{j+1}) - h(x_j)\big) \\
&= 0 \quad \big(h(x_j) = 0\big)
\end{aligned}
$$

The Smoothing Spline

- Hence the optimal solution in $W^2[a, b]$ has a finite sample representation using NCS basis:

$$\widehat{g}(x) = \sum_{j=1}^{n} \beta_j N_j(x),$$

- N_j's are a set of natural cubic spline basis functions with knots at each of the unique x_i values

The Smoothing Spline

- We can then rewrite the objective function in the penalized RSS approach as

$$\sum_{i=1}^{n} \big(y_i - g(x_i)\big)^2 = (\mathbf{y} - \mathbf{F}\boldsymbol{\beta})^\mathsf{T}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta})$$

where $\mathbf{F}$ is an $n \times n$ matrix with $\mathbf{F}_{ij} = N_j(x_i)$

The Smoothing Spline

- The penalty function in that approach becomes

$$\int_a^b g''(x)^2 dx = \int \Big(\sum_j \beta_i N_i''(x)\Big)^2 dx$$
$$= \sum_{j,k} \beta_j \beta_k \int N_j''(x) N_k''(x) dx$$
$$= \boldsymbol{\beta}^\mathsf{T} \Omega \boldsymbol{\beta}$$

where Ω is an $n \times n$ matrix with $\Omega_{jk} = \int N_j''(x) N_k''(x) dx$.

The Smoothing Spline

- Hence our goal is to find $\boldsymbol{\beta}$ that minimizes

$$\mathsf{RSS}(\boldsymbol{\beta}, \lambda) = \|\mathbf{y} - \mathbf{F}\boldsymbol{\beta}\|^2 + \lambda\boldsymbol{\beta}^{\mathsf{T}}\Omega\boldsymbol{\beta}$$

- This is a ridge penalized function and the solution is

$$\begin{aligned}\widehat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} \mathsf{RSS}(\boldsymbol{\beta}, \lambda) \\ &= (\mathbf{F}^{\mathsf{T}}\mathbf{F} + \lambda\Omega)^{-1}\mathbf{F}^{\mathsf{T}}\mathbf{y}\end{aligned}$$

- This method is called the smoothing spline.

Remarks

- The smoothing spline version of the "hat" matrix is called the smoother matrix

$$\begin{aligned}\widehat{f} &= \mathbf{F}(\mathbf{F}^{\mathsf{T}}\mathbf{F} + \lambda\Omega)^{-1}\mathbf{F}^{\mathsf{T}}\mathbf{y} \\ &= \mathbf{S}_{\lambda}\mathbf{y}\end{aligned}$$

- This method also obeys the bias-variance trade-off.
- What happens when $\lambda \to 0$ or $\lambda \to \infty$?

Remarks

- The degrees of freedom of a smoothing spline is

$$\text{df} = \text{Trace}(\mathbf{S}_\lambda)$$

  which ranges between 0 and n.
- Tune λ using GCV:

$$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \widehat{g}(x_i)}{1 - \frac{1}{n}\text{Trace}(\mathbf{S}_\lambda)}\right)^2$$

Smoothing Splines in R

```
> library(splines)
> smooth.spline(x, y = NULL, w = NULL, df, cv = FALSE)
```

- cv : FALSE uses GCV, TRUE uses Leave-one-out CV
- df : degrees of freedom between 1 and n, let GCV decide it automatically
- w : can be used if x has replicates