Module 1: Nonparametric Preliminaries

Selecting Smoothing Parameters

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 11th, 2013

Smoothing Parameter

- In both ridge and lasso regression, we saw that the parameter λ controlled the solution
  - Often, can straightforwardly equate with effective degrees of freedom

Linear smoothers
$\nu_\lambda = tr(L_\lambda)$
"hat matrix"

- Which λ (→ estimator) should we choose???

Want good predictions

Two Goals

- ① **Model Selection:** estimating the performance of models in order to select the best one
  - E.g., choosing λ
- ② **Model Assessment:** having chosen a final model, estimate its prediction error (generalization error) on new data
- Ideally, divide data into 3 parts

① TRAIN | VALIDATION | TEST

	TRAIN	VALIDATION	TEST
Perhaps	Fit models 50%	est. perf. for model selection 25%	assess generalization error for the selected model 25%

Focus on Model Selection

- Which estimator/smoothing parameter should we choose?

TRAIN | VALIDATION

- Recall metrics for assessing the performance of an estimator…

Measuring Predictive Performance

- Assume estimate $\hat{f}_n(\cdot)$ based on training data $y_1, \ldots, y_n$

fixed

- The ***generalization error*** provides a measure of predictive performance

$$GE(\hat{f}_n) = E_{\underline{Y,X}}\left[L(Y, \hat{f}_n(X))\right]$$

fixed

Measuring Predictive Performance

- Assume L_2 loss

$Y = f(X) + \epsilon$ ★ $E[\epsilon] = 0$ $var(\epsilon) = \sigma^2$

- Averaging over repeat training sets $\mathbf{Y}_n = Y_1, \ldots, Y_n$ we get the ***predictive risk*** at x^*

$$E_{Y^*, \mathbf{Y}_n}\left[(Y^* - \hat{f}_n(x^*))^2\right] = E_{Y^*, Y_n}\left[(Y^* - f(x^*) + f(x^*) - \hat{f}_n(x^*))^2\right]$$

test training

fcn of training data Y_n

$$= E_{Y^*}\left[(Y^* - f(x^*))^2\right] + \underbrace{E_{Y_n}\left[(\hat{f}_n(x^*) - f(x^*))^2\right]}_{MSE(\hat{f}_n(x^*))} + 2\,\cancel{E_{Y^*}\left[Y^* - f(x^*)\right]}\, E_{Y_n}\left[\hat{f}_n(x^*) - f(x^*)\right]$$

$\to 0$

$$\boxed{= \sigma^2 + MSE(\hat{f}_n(x^*))}$$ ✓

"irreducible error" "risk"

- Recall $MSE[\hat{f}_n(x)] = \text{bias}(\hat{f}_n(x))^2 + \text{var}(\hat{f}_n(x))$

Measuring Predictive Performance

- Finally, let's average over covariates x
  - ***Integrated MSE*** $\int MSE(\hat{f}_n(x))\, p(x)\, dx$ summary over all inputs
  - ***Average MSE*** $\frac{1}{n}\sum_{i=1}^{n} MSE(\hat{f}_n(x_i))$ Monte Carlo est. $x_i \sim p$
- Note: $\boxed{\textbf{avg. pred. risk} = \sigma^2 + \textbf{avg. MSE}}$

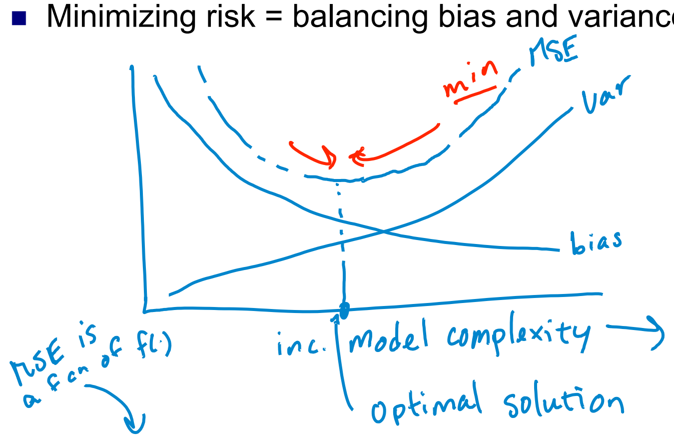
$$\left[\frac{1}{n}\sum_{i=1}^{n} E_{Y_n, Y_n^*}\left[(Y_i^* - \hat{f}(x_i))^2\right]\right.$$

training, new obs. $Y_n^* = Y_1^*, \ldots, Y_n^*$

Bias-Variance Tradeoff

recall polynomial reg. example

- Minimizing risk = balancing bias and variance



- Note: *f(x) is unknown, so cannot actually compute MSE*

Focus on Model Selection

- Which estimator/smoothing parameter should we choose?

TRAIN | VALIDATION

n m

index λ

size n ↑ data

- We saw that minimizing (average) prediction error can be equated with minimizing (average) MSE

est. from training data using λ

- With a validation set, we can estimate the prediction error

$$\frac{1}{m}\sum_{i=1}^{m}\left(y_i^* - \hat{f}_n^{\lambda}(x_i^*)\right)^2$$

obs. in validation set size m

Data Scarce Approximations

- Often, we do not have enough data to form suitably sized training and validation sets
  - What is a good training/test split? Sensitivity?
  - Typically want to use as much data for training as possible
- Rely on other approximations

~~Approx 1: Training Data Only~~

DON'T USE THIS

- **Goal:** Minimize average MSE

unknown

$$\min_{\lambda} E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^{\lambda}(x_i))^2\right]$$

- **Solution:** Use training error

$$\min \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_n^{\lambda}(x_i))^2 \qquad \text{training error} = \frac{RSS}{n}$$

BAD biased downwards + leads to overfitting (undersmoothing)

Data was used twice: → est. fcn ← tried to min $\frac{RSS}{n}$, so underest. risk; → est. risk

Approx 2: Cross Validation

- **Goal:** Minimize average MSE

$$\min_{\lambda} E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^{\lambda}(x_i))^2\right]$$

- **Solution:** Mimic heldout data using *training* data

- Leave-one-out (LOO) cross validation (CV) algorithm:
  - Estimate fit using all but i^{th} data point $\hat{f}_{-i}^{\lambda}$ ← no obs. y_i
  - Predict i^{th} observation
  - Repeat for all i

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_{-i}^{\lambda}(x_i))^2$$

  - Repeat for all values of λ

Approx 2: Cross Validation

- Reasoning

$$CV \approx E[(y_i - \hat{f}^{\lambda}_{-i}(x_i))^2] = E[(y_i - f(x_i) + f(x_i) - \hat{f}^{\lambda}_{-i}(x_i))^2]$$

$$= \sigma^2 + E[(f(x_i) - \hat{f}^{\lambda}_{-i}(x_i))^2]$$

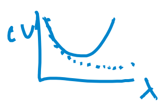
$$\approx \sigma^2 + E[(f(x_i) - \hat{f}^{\lambda}(x_i))^2]$$

- For linear smoothers

$$CV(\lambda) = \frac{1}{n}\sum\left(\frac{y_i - \hat{f}^{\lambda}_n(x_i)}{1 - L_{ii}}\right)^2$$

only do fit once (per λ)

ith diag element of hat matrix



- Warning: Curves can be very flat…Don't just choose and use without thinking. Some rules of thumb (see Elements of Statistical Learning)

Approx 2: Cross Validation

- K-fold cross validation typically $K = 5, 10$

TRAIN	TRAIN	VALID-ATION	TRAIN	TRAIN
$\frac{1}{K}$	$\frac{1}{K}$	$\frac{1}{K}$	$\frac{1}{K}$	$\frac{1}{K}$

randomly assigning obs. to each group

- Algorithm
  1. Fit model using data with k^{th} fraction removed
  2. Using fitted model, compute

$$\mathrm{CV}_k^{(\lambda)} = \frac{1}{n_k}\sum_{i \in J(k)} (y_i - \hat{f}^{\lambda}_{-k}(x_i))$$

w/o kth block

indices for kth block

  3. Store

$$\mathrm{CV}^{(\lambda)} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{CV}_k$$

  4. Repeat for each value of λ using same split of the data

Approx 3: Generalized CV

- Recall LOO ordinary CV for linear smoothers

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^\lambda(x_i)}{1 - L_{ii}}\right)^2$$

- Instead of L_{ii}^λ, use $\frac{1}{n}\sum_{i=1}^{n} L_{ii}^\lambda = \frac{1}{n} tr(L^\lambda) = \frac{\nu_\lambda}{n}$

$$GCV(\lambda) = \frac{1}{n}\sum\left(\frac{y_i - \hat{f}_n^\lambda(x_i)}{1 - \frac{\nu_\lambda}{n}}\right)^2$$

- Often very close to OCV solution

Approx 3: Generalized CV

$$GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^\lambda(x_i)}{1 - \frac{\nu_\lambda}{n}}\right)^2$$

- One motivation: Invariance to orthonormal transformations

$y, X \rightarrow Qy, QX \qquad Q^TQ = QQ^T = I$

$\min (y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta$

$\Updownarrow$

$\min (Qy - QX\beta)^T(Qy - QX\beta) + \lambda\beta^T\beta$

but not the same OCV score

If L^λ is linear smoother for original data, $L_Q^\lambda = QL^\lambda Q^T$ is for trans. data

$tr(L_Q^\lambda) = tr(QL^\lambda Q^T) = tr(L^\lambda)$

trace trick

$\boxed{\text{GCV will be the same}}$

Approx 3: Generalized CV

$$GCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{f}_n^\lambda(x_i)}{1-\frac{\nu_\lambda}{n}}\right)^2$$

- Using $(1-x)^{-2} \approx 1 + 2x$

$$GCV(\lambda) \approx \frac{1}{n}\sum(y_i - \hat{f}_n^\lambda(x_i))^2 + \frac{2\nu_\lambda \sigma^2}{n} \leftarrow \frac{1}{n}\sum(y_i - \hat{f}_n^\lambda(x_i))^2$$

$\simeq$ Mallow's C_p stat

(not exactly the right $\hat{\sigma}^2$...)

Approx 4: Mallows C_p Statistic

- **Goal:** Minimize average MSE

for linear smoothers

$$\min_\lambda E\left[\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}_n^\lambda(x_i))^2\right]$$

- **Solution:** Approximate directly

$$\text{avg. MSE} = \frac{1}{n}E\left[(f - \hat{f}_n^\lambda)^T(f - \hat{f}_n^\lambda)\right] = \frac{1}{n}E[(\underbrace{Y-\epsilon}_{f} - \underbrace{L^\lambda Y}_{\hat{f}_n^\lambda})^T(Y-\epsilon-L^\lambda Y)]$$

$$= \frac{1}{n}E[(Y-L^\lambda Y)^T(Y-L^\lambda Y)] - \sigma^2 + \frac{2}{n}\nu_\lambda\sigma^2$$

uses $E[\epsilon^T L^\lambda \epsilon] = E[tr(\epsilon^T L^\lambda \epsilon)] = E[tr(L^\lambda \epsilon\epsilon^T)]$

$= tr(L^\lambda I \sigma^2) = \sigma^2 \nu_\lambda$

Approx 4: Mallows C_p Statistic

$$\text{avg. MSE} = \frac{1}{n}E\left[(Y - L^\lambda Y)^T(Y - L^\lambda Y)\right] - \sigma^2 + \frac{2}{n}\nu_\lambda\sigma^2$$

- Estimate as

$$\widehat{\text{avg. MSE}} = \frac{1}{n}RSS^\lambda - \frac{1}{n}(n-2\nu_\lambda)\hat{\sigma}^2_{max}$$

using a maximal model

$$\Updownarrow$$

min Mallow's Cp

$$\frac{RSS^\lambda}{\hat{\sigma}^2_{max}} - (n - 2\nu_\lambda)$$

- Note: Arises from considering L_2 loss. Log-likelihood loss leads to AIC. For BIC, consider Bayesian model selection

Bayesian Model Selection

- Assume some *M* possible models
  - Model M_m *m=1,…,M* has parameters θ_m and prior $p(\theta_m \mid M_m)$
  - Prior over models $p(M_m)$
- Model posterior

training data

$$p(M_m \mid Z) \propto p(M_m)p(Z \mid M_m)$$
$$\propto p(M_m)\int p(Z \mid \theta_m, M_m)p(\theta_m \mid M_m)d\theta_m$$

- Compare models:

post. odds

$$\frac{p(M_m \mid Z)}{p(M_\ell \mid Z)} = \frac{p(M_m)}{p(M_\ell)}\boxed{\frac{p(Z \mid M_m)}{p(Z \mid M_\ell)}} \overset{m}{\underset{\ell}{\gtrless}} 1$$

often, uniform prior

Bayes factor

Bayesian Model Selection

- For Bayes factor, approximate Laplace + ... ν_m ← # of free params

$$\log p(Z \mid M_m) \approx \log p(Z \mid \hat{\theta}_m, M_m) - \frac{\nu_m}{2}\log n + O(1)$$

← ML est.

- If loss is $-2\log p(Z \mid \hat{\theta}_m, M_m)$, then equivalent to BIC
  - Minimizing BIC = maximizing approximated posterior
- However, in addition to being able to select the best model, in Bayesian framework we also get the relative merit of each

$$\approx \frac{e^{-\frac{1}{2}\mathrm{BIC}_m}}{\sum_{\ell=1}^{M} e^{-\frac{1}{2}\mathrm{BIC}_\ell}}$$

- BIC is asymptotically consistent, but AIC is not
- For finite samples, BIC tends to choose too simple models

Module 2: Splines and Kernel Methods

Spline Model Overview, Regression Splines, Smoothing Splines

STAT/BIOSTAT 527, University of Washington

Emily Fox

April 11th, 2013

Moving Beyond Linearity

- So far we have assumed standard linear models

$$\min_\beta \|y - X\beta\|_2^2 \quad \longleftarrow f(x) = \beta^T x$$

- In the case of many predictors relative to number of observations, we considered penalized regression to avoid overfitting

$$\min_\beta \|y - X\beta\|_2^2 + \lambda \|\beta\|$$

- Often a convenient form, and necessary to assume simple structure to avoid overfitting in data-scarce regimes, but linear assumption rarely holds in practice

Moving Beyond Linearity

- Consider generic functional forms (univariate *x* for now)

$$\min_f \|y - f(x)\|_2^2$$

  - If constrained to linear forms → LS soln
  - If arbitrary → interpolator… overfitting
- As before, penalize complexity. Here, in terms of roughness.

$$\min_f \|y - f(x)\|_2^2 + \lambda \int f''(x)^2 \, dx$$

  - If λ → 0, interpolator
  - If λ → ∞, LS soln (line) … no 2^{nd} der.
- Remarkable result: Explicit, finite-dimensional minimizer

TBD natural cubic spline w/ knots at data pts

"smoothing spline"

Backtrack a bit…

- Instead of just considering input variables *x* (potentially mult.), augment/replace with transformations = "input features"
- ***Linear basis expansions*** maintain linear form in terms of these transformations

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

trans.

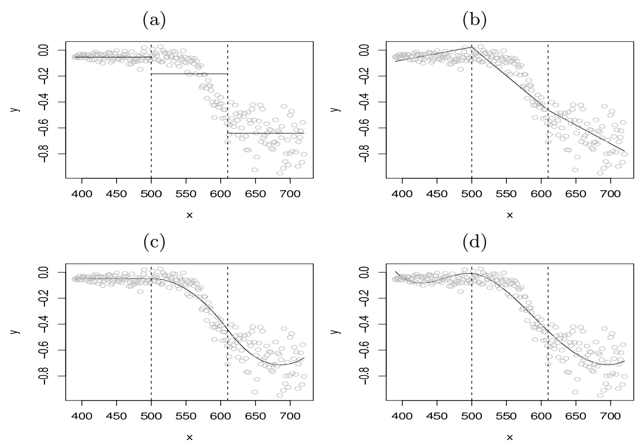
- What transformations should we use?
  - $h_m(x) = x_m$ → linear model
  - $h_m(x) = x_j^2, \quad h_m(x) = x_j x_k$ → polynomial reg.
  - $h_m(x) = I(L_m \leq x_k \leq U_m)$ → piecewise constant
  - …

Piecewise Polynomial Fits

- Again, assume *x* univariate
- Polynomial fits are often good locally, but not globally
  - Adjusting coefficients to fit one region can make the function go wild in other regions
- Consider ***piecewise polynomial*** fits
  - Local behavior can often be well approximated by low-order polynomials

Piecewise Polynomial Fits

LIDAR Data Example



From Wakefield book

Piecewise Constant/Linear Fits

- **Example 1: Piecewise constant, with 3 basis functions**

$h_1(x) = I(x \leq \xi_1)$ "knot"

$h_2(x) = I(\xi_1 \leq x \leq \xi_2)$

$h_3(x) = I(\xi_2 \leq x)$

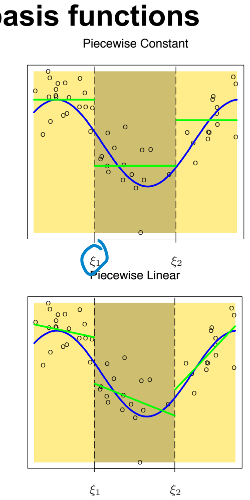
- Resulting model: $f(x) = \sum_{m=1}^{3} \beta_m h_m(x)$

- Fit: Take mean of data in each region

$\hat{\beta}_m = \bar{y}_m$

- **Example 2: Piecewise linear**
- Add three basis functions:

$$h_{m+3} = h_m(x)x \quad m=1,2,3$$



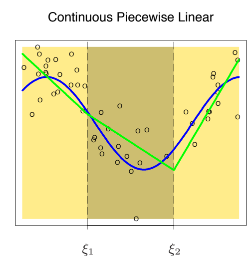
From Hastie, Tibshirani, Friedman book

Regression Splines – Linear

- Resulting piecewise linear model:

$$f(x) = I(x < \xi_1)(\beta_1 + \beta_4 x) + I(\xi_1 \leq x < \xi_2)(\beta_2 + \beta_5 x) + I(\xi_2 \leq x)(\beta_3 + \beta_6 x)$$

  - # of params? 6
- Typically prefer continuity...
  - Enforce $f(\xi_1^-) = f(\xi_1^+)$
    $f(\xi_2^-) = f(\xi_2^+)$
  - Which implies
    $\beta_1 + \beta_4 \xi_1 = \beta_2 + \beta_5 \xi_1$
    $\beta_2 + \beta_5 \xi_2 = \beta_3 + \beta_6 \xi_2$
  - # params?
    $6 - 2 = 4$

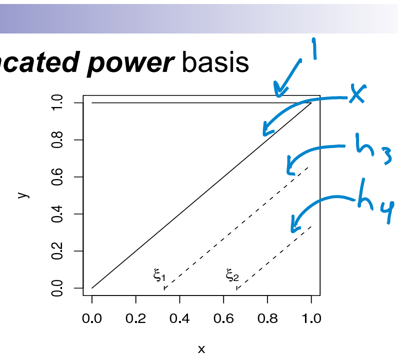


From Hastie, Tibshirani, Friedman book

Regression Splines – Linear

- More directly, we can use the ***truncated power*** basis

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



From Wakefield book

- Resulting model:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 (x - \xi_1)_+ + \beta_3 (x - \xi_2)_+$$

- Continuous at the knots because all prior basis functions are contributing to the fit up to any single *x*

Regression Splines – Cubic

- Naively, extend as quadratic

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 (x-\xi_1)_+ + \beta_4 (x-\xi_1)_+^2 + \beta_5 (x-\xi_2)_+ + \beta_6 (x-\xi_2)_+^2$$

- But, 1st derivate is discontinuous (check this)
- Drop the truncated linear basis:

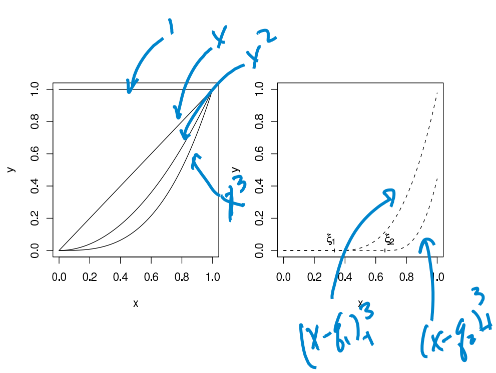
$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + b_1 (x-\xi_1)_+^2 + b_2 (x-\xi_2)_+^2$$

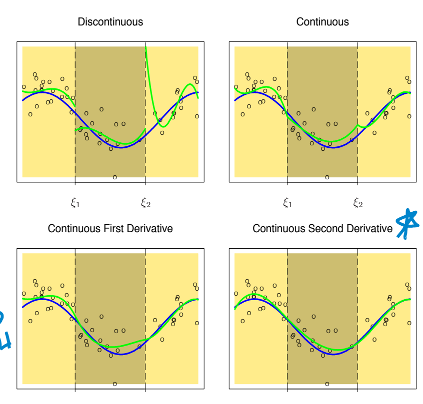
- Has continuous 1st derivative (check), but not 2nd
- Popular to consider ***cubic spline***:

$$\boxed{f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + b_1 (x-\xi_2)_+^3 + b_2 (x-\xi_2)_+^3}$$

- Has continuous 1st and 2nd derivatives
- Typically people stop here … smooth enough

Cubic Spline Basis and Fit





Cubic Splines as Linear Smoothers

- Cubic spline function with K knots:

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

- Simply a linear model $f(x) = E[Y \mid C] = C\gamma$

$$C = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & (x_1-\xi_1)_+^3 & \cdots & (x_1-\xi_K)_+^3 \\ \vdots & & & & & & \\ 1 & x_n & x_n^2 & x_n^3 & (x_n-\xi_1)_+^3 & \cdots & (x_n-\xi_K)_+^3 \end{bmatrix} \qquad \gamma = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ b_1 \\ \vdots \\ b_K \end{bmatrix}$$

- Estimator: $\hat{\gamma} = (C^T C)^{-1} C^T y$

- Linear smoother: $\hat{f} = \underbrace{C(C^T C)^{-1} C^T}_{L} y$

Natural Cubic Splines

- For polynomial regression, fit near boundaries is erratic.
  - Problem is worse for splines: each is fit locally so no global constraint

- ***Natural cubic splines*** enforce linearity beyond boundary knots

- Starting from a cubic spline basis, the natural cubic spline basis is

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

$$d_k(x) = \frac{(x-\xi_k)_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_k}$$

- Derivation HW 3

Regression Splines – Summary

- Definition:

  *An **order-M spline** with knots* $\xi_1 < \xi_2 < \cdots < \xi_K$ *is a piecewise M-1 degree polynomial with M-2 continuous derivatives as the knots*

  *A spline that is linear beyond the boundary knots is called a **natural spline***

- Choices:
  - Order of the spline
  - Number of knots
  - Placement of knots

require some thought

Return to Smoothing Splines

- Objective:

$$\min_f \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(x)^2 dx$$

- Solution:
  - ***Natural cubic spline*** ✓
  - Place knots at every observation location x_i ☆

- Proof: See Green and Silverman (1994, Chapter 2) or Wakefield textbook

- Notes:
  - Would seem to overfit, but penalty term shrinks spline coefficients toward linear fit
  - Will not typically interpolate data, and smoothness is determined by λ

Smoothing Splines

- Model is of the form: $f(x) = \sum_{j=1}^{n} N_j(x)\beta_j$

  n: # of obs.

  N_j: natural cubic spline basis

- Rewrite objective:

$$(y - N\beta)^T(y - N\beta) + \lambda\beta^T\Omega_N\beta$$

$$[N]_{ij} = N_j(x_i) \qquad [\Omega_N]_{jk} = \int N_j''(t)N_k''(t)\,dt$$

- Solution: $\hat{\beta} = (N^TN + \lambda\Omega_N)^{-1}N^Ty$ as in ridge

- Linear smoother:

$$\hat{f} = \underbrace{N(N^TN + \lambda\Omega_N)^{-1}N^T}_{L_\lambda}y$$

"smoothing matrix"

$$V_\lambda = tr(L_\lambda)$$

Splines – Summary

- **Regression splines:**

  Fewer number of knots and no regularization

- **Smoothing splines:**

  Knots at every observation and regularization (smoothness penalty) to avoid interpolators