Module 2: Splines and Kernel Methods

B-Splines, Penalized Regression Splines

STAT/BIOSTAT 527, University of Washington

Emily Fox

April 16th, 2013

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

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

- 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   mult. x later in course
- 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

Cubic Spline Basis and Fit

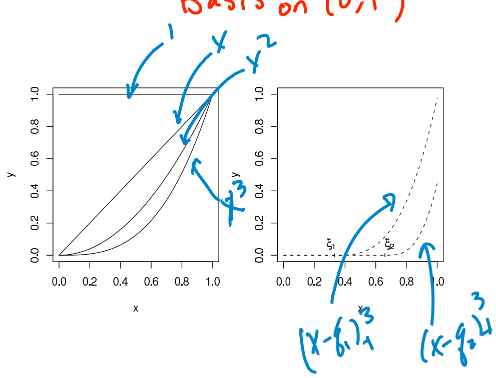
- Cubic spline function with *K* knots:

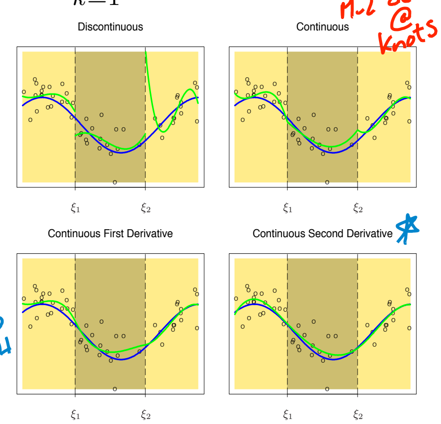
truncated power basis

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

M=4
M-1 deg
M-2 der @ knots

basis on (0,1)





B-Splines

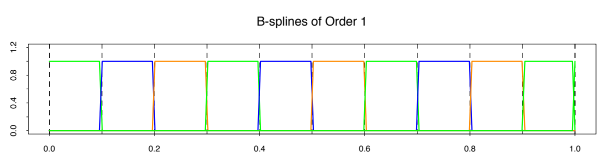
- Alternative basis for representing polynomial splines
- Computationally attractive…Non-zero over limited range
- As before:
  - Knots $\xi_1 < \cdots < \xi_K$
  - Domain (a,b)
  - Number of basis functions = $M+K$ ← deg. of polys + 1
- Step 1: Add knots $\xi_0 = a$ $\xi_{K+1} = b$
- Step 2: Define auxiliary knots τ_j needed to construct basis

choice is arb.

$$\tau_1 \leq \tau_2 \leq \cdots \leq \tau_M \leq \xi_0$$
$$\tau_{j+M} = \xi_j$$
$$\xi_{K+1} \leq \tau_{K+M+1} \leq \cdots \leq \tau_{K+2M}$$

B-Splines

- For 1st order B-spline



From Hastie, Tibshirani, Friedman book

$$B_j^1(x) = \begin{cases} 1 & \tau_j \le x \le \tau_{j+1} \\ 0 & \text{ow} \end{cases}$$

Haar basis function

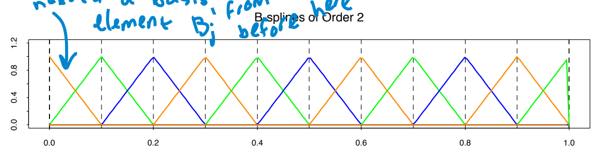
Can form any piecewise constant fcn

B-Splines

- For 2^{nd} order B-spline → piecewise linear fcn + cont. @ knots

needed a basis element B_j^1 from before here

bad



From Hastie, Tibshirani, Friedman book

- Modify 1^{st} order basis:

$$B_j^2(x) = \frac{x-\tau_j}{\tau_{j+1}-\tau_j} B_j^1(x) + \frac{\tau_{j+2}-x}{\tau_{j+2}-\tau_{j+1}} B_{j+1}^1(x)$$

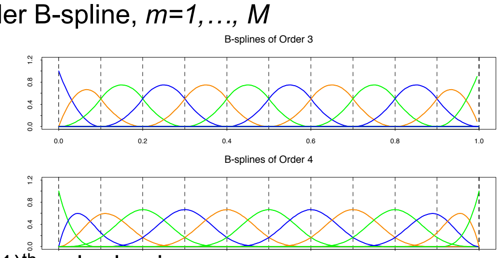
pos. slope

neg. slope

- Convention: If divide by 0, set basis element to 0 if $\tau_j = \tau_{j+1}$

B-Splines

- For m^{th} order B-spline, *m=1,…, M*



From Hastie, Tibshirani, Friedman book

- Modify $(m-1)^{th}$ order basis:

$$B_j^m(x) = \frac{x-\tau_j}{\tau_{j+m-1}-\tau_j} B_j^{m-1} + \frac{\tau_{j+m}-x}{\tau_{j+m}-\tau_{j+1}} B_{j+1}^{m-1}$$

- B-spline bases are non-zero over domain spanned by at most M+1 knots
- Only subset are needed for basis of order *M* with knots ξ

$$\{B_i^m \mid i = M-m+1, \dots, M+K\}$$

For m=M → M+K basis fcns

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

Cubic B-Splines as Linear Smoothers

- Cubic B-spline with K knots has basis expansion:

$$f(x) = \sum_{j=1}^{K+4} B_j^4(x)\beta_j$$

- Simply a linear model

$$B = \begin{bmatrix} B_1^4(x_1) & \cdots & B_{K+4}^4(x_1) \\ \vdots & \ddots & \vdots \\ B_1^4(x_n) & \cdots & B_{K+4}^4(x_n) \end{bmatrix} \qquad \gamma = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{K+4} \end{bmatrix}$$

- Computational gain: $\hat{\gamma} = (B^T B)^{-1} B^T y$

$n \times (K+M)$ matrix B has many 0's

$\rightarrow$ fewer multiplies (sparse inv.)

Return to Smoothing Splines

- Objective:

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

Smoothness penalty

- 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

of obs.

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

natural cubic spline basis

- Rewrite objective:

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

n x n matrix $[N]_{ij} = N_j(x_i)$

$[\Omega_N]_{jk} = \int N_j''(t)N_k''(t)dt$

- 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"

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

Smoothing Splines

Previously,

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

n+4 ← order M spline (for M=4 ⇒ cubic poly)

Now,

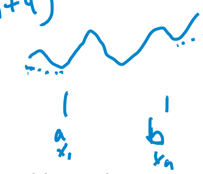
- Using B-spline basis instead: $f(x) = \sum_{j=1} B_j(x)\beta_j$

- Solution: $\hat{\beta} = (B^T B + \lambda \Omega_B)^{-1} B^T y$

(n+4) x (n+4)

n x (n+4)

lower 4 banded → comp. eff.



- Penalty implicitly leads to natural splines
  - Objective gives infinite weight to non-zero derivatives beyond boundary

forces soln to be linear beyond boundary pts → natural splines

Spline Overview (so far)

Smoothing Splines

- Knots at data points x_i
- Natural cubic spline
- O(n) parameters
  - Shrunk towards subspace of smoother functions

Regression Splines

- $K < n$ knots chosen
- M^{th} order spline = piecewise M-1 degree polynomial with M-2 continuous derivatives at knots
- no reg, but many fewer params

- Linear smoothers, for example using natural cubic spline basis:

$L = N(N^T N + \lambda \Omega_N)^{-1} N^T$ vs. $L = N(N^T N)^{-1} N^T$

n×n

penalty/reg.

n×K matrix

K=n

\# params = 4+K-4 ← add'l const.

Penalized Regression Splines

- Alternative approach:
  - Use $K < n$ knots few params relative to # of obs.
  - How to choose K and knot locations? ??
- Option #1:
  - Place knots at n unique observation locations x_i and do stepwise
  - Issue?? 2^n models!
- Option #2:
  - Place many knots for flexibility
  - Penalize parameters associated with knots just like ridge/lasso
- Note: Smoothing splines penalize complexity in terms of roughness. Penalized reg. splines shrink coefficients of knots.

Penalized Regression Splines

- General spline model $f(x) = \sum_{j=1}^{J} h_j(x)\beta_j$ some spline basis
- Definition: A ***penalized regression spline*** is $\underline{\hat{\beta}^T h(x)}$ with

$$\hat{\beta} = \min_{\beta} \sum_{i=1}^{n} (y_i - \beta^T h(x_i))^2 + \lambda \beta^T D \beta$$

penalty matrix

- Form of resulting spline depends on choice of
  - Basis $\{h_j(x)\}$
  - Penalty matrix D
  - Penalty strength λ
- Still need to K and associated locations…RoT (Ruppert et al 2003):

$$K = \min(\frac{1}{4} \times \ \# \text{ unique } x_i, 35) \qquad \xi_k \text{ at } \frac{k+1}{K+2} th \text{ points of } x_i$$

PRS Example #1

$$\sum_{i=1}^{n}(y_i - \beta^T h(x_i))^2 + \lambda\beta^T D\beta$$

- cubic B-spline basis + penalty

$h_j = B_j^4$

$$\lambda \int \left(\sum_{j=1}^{K+4} B_j^4(x)'' \beta_j \right)^2 dx \qquad \beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{K+4} \end{bmatrix}$$

- For this penalty, the matrix *D* is given by

$$D_{jk} = \int B_j^4(x)'' B_k^4(x)'' dx$$

- Leads to "O'Sullivan Splines"

when $K=n$ + @ unique x_i, exactly equivalent to smoothing spline

©Emily Fox 2013 17

PRS Example #2

$$\sum_{i=1}^{n}(y_i - \beta^T h(x_i))^2 + \lambda\beta^T D\beta$$

- B-spline basis + penalty

$$\lambda \sum_{j=1}^{J-1} (\beta_{j+1} - \beta_j)^2$$

- For this penalty, the matrix *D* is given by

$$D = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots \\ -1 & 2 & -1 & 0 & \cdots \\ 0 & -1 & 2 & -1 & 0 & \cdots \\ \vdots & & \vdots \end{bmatrix}$$

- Leads to

"P-Splines"

penalizes large changes in coeff. of adj. basis fcns.

→ smoothing

≈ integrated squared derivative penalty of O'Sullivan splines

©Emily Fox 2013 18

PRS Example #3

$$\sum_{i=1}^{n}(y_i - \beta^T h(x_i))^2 + \lambda \beta^T D \beta$$

- Cubic spline using truncated power basis K

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

  \+ penalty on truncated power coefficients

$$\lambda \sum_k b_k^2 \iff \lambda \|b\|_2^2$$

- For this penalty, the matrix D is given by

before γ

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_3 \\ b_1 \\ \vdots \\ b_K \end{bmatrix} \qquad D = \begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & \ddots & & & \\ & & & 0 & & \\ & & & & 1 & \\ & & & & & \ddots \\ & & & & & & 1 \end{bmatrix}$$

β_j's

b_k's

A Brief Spline Summary

- ***Smoothing spline*** – contains *n* knots @ x_i
- ***Cubic smoothing spline*** – piecewise cubic
- ***Natural spline*** – linear beyond boundary knots
- ***Regression spline*** – spline with $K < n$ knots chosen
- ***Penalized regression spline*** – imposes penalty (various choices) on coefficients associated with piecewise polynomial
- The # of basis functions depends on
  - # of knots K
  - Degree of polynomial $M-1$
  - A reduced number if a natural spline is considered (add constraints)

Module 2: Splines and Kernel Methods

Intro to Kernels, Local Polynomial Reg., Kernel Density Estimation

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 16th, 2013

Motivating Kernel Methods

- Recall original goal from Lecture 1:
  - We don't actually know the data-generating mechanism
  - Need an estimator $\hat{f}_n(\cdot)$ based on a random sample $Y_1, \ldots, Y_n$, also known as ***training data***
- Proposed a simple model as estimator of $E[\,Y \mid X\,]$

$$\hat{f}(x) = \text{Avg}(y_i \mid x_i \in \underline{\text{Nbhd}}(x))$$

use all obs. y_i in a neighborhood of target x

Choice 1: k Nearest Neighbors

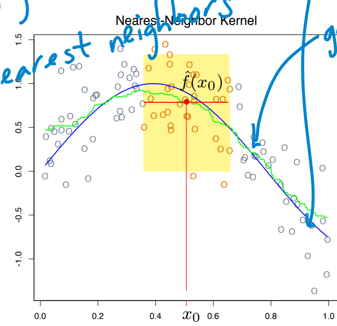
- Define nbhd of each data point x_i by the k nearest neighbors
  - Search for k closest observations and average these

$$\hat{f}(x) = \text{Avg}(y_i \mid x_i \in N_k(x))$$

$\uparrow$ k-nearest neighbors

blue = true

green = est.



From Hastie, Tibshirani, Friedman book

- Discontinuity is unappealing

neighbors are either in or out $\rightarrow$ disc.

Choice #2: Local Averages

- A simpler choice examines a fixed distance h around each x_i
  - Define set: $B_x = \{i : |x_i - x| \leq h\}$
  - # of x_i in set: n_x

$$\hat{f}(x) = \frac{1}{n_x} \sum_{i \in B_x} y_i$$

avg. obs. within distance h

- Results in a linear smoother

$$\hat{f}(x) = \sum_{i=1}^{n} l_i(x)\, y_i \qquad l_i(x) = \begin{cases} \frac{1}{n_x} & \text{if } |x_i - x| \leq h \\ 0 & \text{ow} \end{cases}$$

- For example, with $x_i = \frac{i}{9}$ and $h = \frac{1}{9}$

$$L = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & \cdots \\ 1/3 & 1/3 & 1/3 & 0 & \cdots \\ 0 & 1/3 & 1/3 & 1/3 & \\ \cdots & & & & \end{bmatrix}$$

More General Forms

- Instead of weighting all points equally, slowly add some in and let others gradually die off

- ***Nadaraya-Watson kernel weighted average***

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{\hat{}} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{\hat{}} K_\lambda(x_0, x_i)}$$

$$K_\lambda(x_0, x) = K\left(\frac{|x_0 - x|}{\lambda}\right)$$

kernel

bandwidth

- But what is a ***kernel*** ???

Kernels

- Could spend an entire quarter (or more!) just on kernels
- Will see them again in the Bayesian nonparametrics portion

- For now, the following definition suffices

$K(\cdot)$ is a kernel if

$K(x) \geq 0 \quad \forall x$

$\int K(u)\,du = 1$

$\sigma_K^2 = \int u^2 K(u)\,du < \infty$

$\int u\,K(u)\,du = 0$

Example Kernels

ind. on -1, 1

- *Gaussian* $K(x) = \frac{1}{2\pi} e^{-\frac{x}{2}}$
- *Epanechnikov* $K(x) = \frac{3}{4}(1-x)^2 I(x)$
- *Tricube* $K(x) = \frac{70}{81}(1-|x|^3)^3 I(x)$
- *Boxcar* $K(x) = \frac{1}{2} I(x)$