Module 2: Splines and Kernel Methods

B-Splines Recap

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 15th, 2014

Cubic Spline Basis and Fit

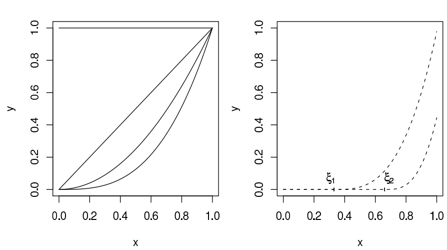
using truncated power basis

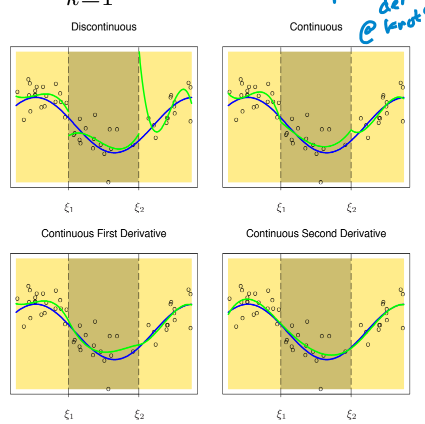
- Cubic spline function with *K* knots:

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

M=4
M-1 deg poly
M-2 cont. der @ knots

basis on (0,1)





B-Splines

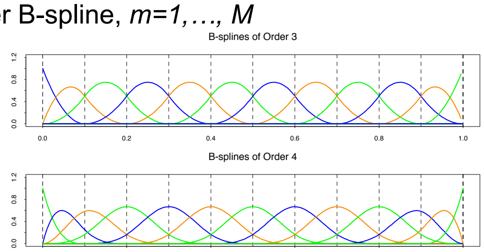
- Alternative basis for representing polynomial splines
- Computationally attractive…Non-zero over limited range
- As before:
  - Knots $\xi_1 < \dots < \xi_K$
  - Domain (a, b)
  - Number of basis functions = $M + K$ deg. of poly. +1
- Step 1: Add knots $\xi_0 = a$ $\xi_{K+1} = b$
- Step 2: Define auxiliary 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 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~~ subsets $\{B_i^m \mid i = M - m + 1, \dots, M + K\}$ are needed for basis of order *m* with knots ξ_j for $m = M \rightarrow M+K$ basis fcns

Cubic Splines as Linear Smoothers

truncated power basis

- 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

$f(x) = E[Y|C] = C\gamma$

$$C_{n\times(4+K)} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & (x_1-\xi_1)_+^3 & \cdots & (x_1-\xi_K)_+^3 \\ \vdots & & & & & & \\ 1 & x_n & x_n^2 & x_n^3 & (x_n-\xi_1)_+^3 & \cdots & (x_n-\xi_K)_+^3 \end{bmatrix} \quad \gamma = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ b_1 \\ \vdots \\ b_K \end{bmatrix}$$

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

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

Cubic B-Splines as Linear Smoother

- 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

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

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

- Computational gain:

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

$\rightarrow$ fewer multiplies (sparse inv.)

Return to Smoothing Splines

- Objective:

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

- 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

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×n matrix

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

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

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

as in ridge

- Linear smoother:

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

"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 knots

Now,

- Using B-spline basis instead: $f(x) = \sum_{j=1}^{n+4} B_j^4(x)\beta_j$ — order M=4 spline (cubic)
- Solution: $\hat{\beta} = (B^T B + \lambda \Omega_B)^{-1} B^T y$

n×(n+4)

(n×4)×(n×4)

lower 4 banded → computational eff.

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

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

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

Module 2: Splines and Kernel Methods

Penalized Regression Splines

STAT/BIOSTAT 527, University of Washington

Emily Fox

April 15th, 2014

Penalized Regression Splines

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

Penalized Regression Splines

- General spline model
- Definition: A ***penalized regression spline*** is $\hat{\beta}^T h(x)$ with
- Form of resulting spline depends on choice of
  - Basis
  - Penalty matrix
  - Penalty strength
- Still need to choose *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
- For this penalty, the matrix *D* is given by
- Leads to

PRS Example #2

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

- B-spline basis + penalty
- For this penalty, the matrix *D* is given by
- Leads to

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

  + penalty on truncated power coefficients

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

A Brief Spline Summary

- ***Smoothing spline*** – contains *n* knots
- ***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
  - Degree of polynomial
  - A reduced number if a natural spline is considered (add constraints)

Reading

- Hastie, Tibshirani, Friedman: 5.1-5.5 (skipping 5.3), Ch. 5 appendix
- Wakefield: 11.1.1-11.2.6

What you should know…

- Regression splines
  - Cubic splines, natural cubic splines, …
  - Interpretation as a linear smoother
  - Degrees of freedom
- Smoothing splines
  - Arising from penalized regression setting with smoothness penalty
  - Cubic spline basis with knots at every data point
- Natural splines
  - Linear beyond boundary points
- B-splines
  - Basis functions with compact support
- Penalized regression splines
  - Choose knots as in regression splines, but penalize associated coefficients

Module 2: Splines and Kernel Methods

Local Polynomial Reg., Kernel Density Estimation

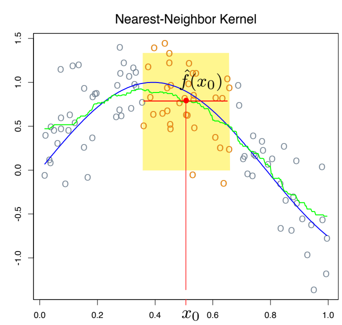
STAT/BIOSTAT 527, University of Washington
Emily Fox
April 15th, 2014

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 $E[\,Y \mid X\,]$

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
- Discontinuity is unappealing



From Hastie, Tibshirani, Friedman book

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

- Results in a linear smoother

- For example, with x_i= and *h*=

$$L =$$

More General Forms

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

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

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

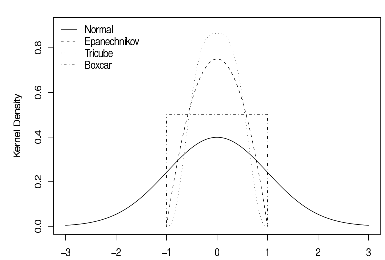
Kernels

- 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

Example Kernels

- *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* $K(x) = \frac{1}{2} I(x)$



Nadaraya-Watson Estimator

- Return to Nadaraya-Watson kernel weighted average

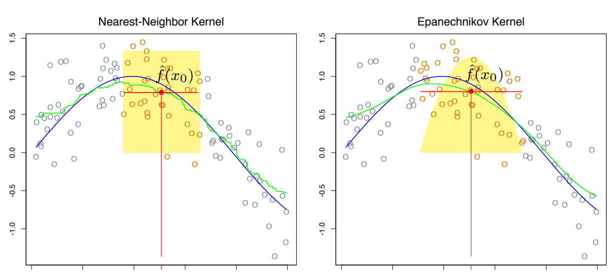
$$\hat{f}(x_0) = \frac{\sum_{i=1}^n K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^n K_\lambda(x_0, x_i)}$$

- Linear smoother:

Nadaraya-Watson Estimator

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

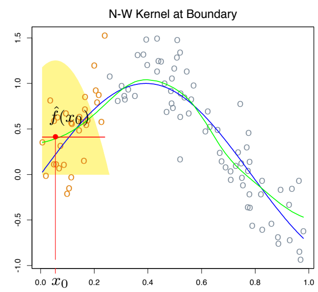
- Example:
  - Boxcar kernel →
  - Epanechnikov
  - Gaussian
- Often, choice of kernel matters much less than choice of λ



From Hastie, Tibshirani, Friedman book

Local Linear Regression

- Locally weighted averages can be badly biased at the boundaries because of asymmetries in the kernel
- Reinterpretation:



From Hastie, Tibshirani, Friedman book

- Equivalent to the Nadaraya-Watson estimator
- Locally constant estimator obtained from weighted least squares

Local Linear Regression

- Consider locally weighted linear regression instead
- Local linear model around fixed target x_0 :
- Minimize:
- Return:
- Fit a new local polynomial for *every* target x_0

Local Linear Regression

$$\min_{\beta_{x_0}} \sum_{i=1}^{n} K_\lambda(x_0, x_i)(y_i - \beta_{0x_0} - \beta_{1x_0}(x_i - x_0))^2$$

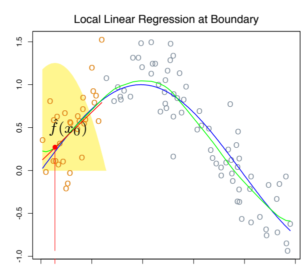
- Equivalently, minimize

- Solution:

Local Linear Regression

- Bias calculation:

$$E[\hat{f}(x_0)] = \sum_i \ell_i(x_0) f(x_i)$$



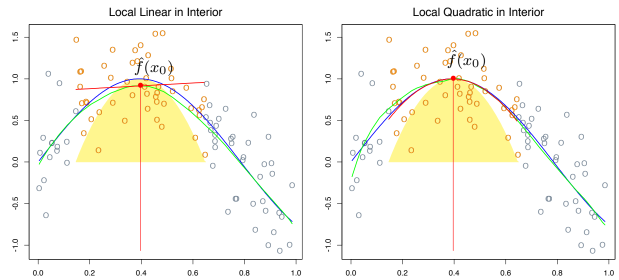
From Hastie, Tibshirani, Friedman book

- Bias $E[\hat{f}(x_0)] - f(x_0)$ only depends on quadratic and higher order terms

- Local linear regression corrects bias exactly to 1st order

Local Polynomial Regression

- Local linear regression is biased in regions of curvature
  - "Trimming the hills" and "filling the valleys"
- Local quadratics tend to eliminate this bias, but at the cost of increased variance



From Hastie, Tibshirani, Friedman book

Local Polynomial Regression

- Consider local polynomial of degree *d* centered about x_0
  $P_{x_0}(x; \beta_{x_0}) =$
- Minimize: $\min_{\beta_{x_0}} \sum_{i=1}^{n} K_\lambda(x_0, x_i)(y_i - P_{x_0}(x; \beta_{x_0}))^2$
- Equivalently:
- Return:
- Bias only has components of degree *d+1* and higher

Local Polynomial Regression

- Rules of thumb:
  - Local linear fit helps at boundaries with minimum increase in variance
  - Local quadratic fit doesn't help at boundaries and increases variance
  - Local quadratic fit helps most for capturing curvature in the interior
  - Asymptotic analysis →
    local polynomials of odd degree dominate those of even degree
    (MSE dominated by boundary effects)

  - Recommended default choice: **local linear regression**

Kernel Density Estimation

- Kernel methods are often used for density estimation (actually, classical origin)

- Assume random sample

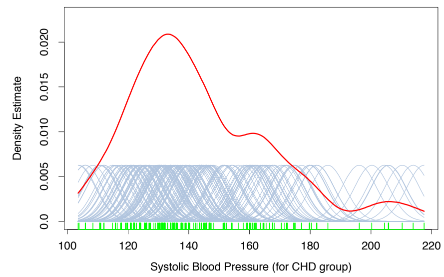
- Choice #1: empirical estimate?

- Choice #2: as before, maybe we should use an estimator

- Choice #3: again, consider kernel weightings instead

Kernel Density Estimation

- Popular choice = Gaussian kernel → ***Gaussian KDE***



From Hastie, Tibshirani, Friedman book

KDE Properties

$$\hat{p}^{\lambda}(x) = \frac{1}{n\lambda}\sum_{i=1}^{n} K\left(\frac{x - x_i}{\lambda}\right)$$

- Let's examine the bias of the KDE

$$E[\hat{p}^{\lambda}(x)] =$$

- Smoothing leads to biased estimator with mean a smoother version of the true density
- For kernel estimate to concentrate about *x* and bias→0, want

KDE Properties

$$\hat{p}^{\lambda}(x) = \frac{1}{n\lambda}\sum_{i=1}^{n} K\left(\frac{x - x_i}{\lambda}\right)$$

- Assuming smoothness properties of the target distribution, it's straightforward to show that

  $E[\hat{p}^{\lambda}(x)] =$

  - In peaks, negative bias and KDE underestimates *p*
  - In troughs, positive bias and KDE over estimates *p*
  - Again, "trimming the hills" and "filling the valleys"
- For var→0, require
- More details, including IMSE, in Wakefield book
- Fun fact: There does not exist an estimator that converges faster than KDE assuming only existence of p''

Connecting KDE and N-W Est.

- Recall task:

  $f(x) = E[Y \mid x] = \int yp(y \mid x)dy$

- Estimate joint density *p(x,y)* with product kernel

  $\hat{p}^{\lambda_x,\lambda_y}(x,y) =$

- Estimate margin *p(y)* by

  $\hat{p}^{\lambda_x}(x) =$

Connecting KDE and N-W Est.

- Then,

  $\hat{f}(x) =$

- Equivalent to Naradaya-Watson weighted average estimator

Reading

- Hastie, Tibshirani, Friedman: 6.1-6.2, 6.6
- Wakefield: 11.3

What you should know…

- Definition of a kernel and examples
- Nearest neighbors vs. local averages
- Nadarya-Watson estimation
  - Interpretation as local linear regression
- Local polynomial regression
  - Definition
  - Properties/ rules of thumb
- Kernel density estimation
  - Definition
  - Properties
  - Relationship to Nadarya-Watson estimation