Module 2: Splines and Kernel Methods

Local Polynomial Reg., Kernel Density Estimation

STAT/BIOSTAT 527, University of Washington
Emily Fox

April 18th, 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 $E[Y \mid X]$

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

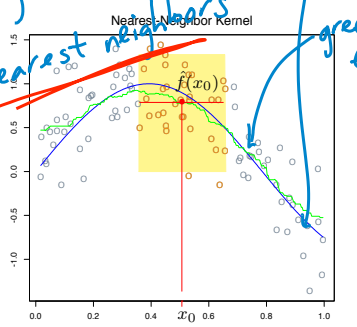
use all obs. y_i in a neighborhood of target x

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

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

k-nearest neighbors



blue = true

green = est.

- Discontinuity is unappealing

neighbors are either in or out → disc.

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

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

avg. obs. within distance h

- Results in a linear smoother

$\hat{f}(x) = \sum_{i=1}^{n} l_i(x)\, y_i$

$l_i(x) = \begin{cases} \frac{1}{n_x} & \text{if } |x_i - x| \leq h \\ 0 & \text{ow} \end{cases}$

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

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

More General Forms

- Instead of weighting all points equally, slowly add some in and let others gradually die off
- ***Nadaraya-Watson kernel weighted average***

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

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

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

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

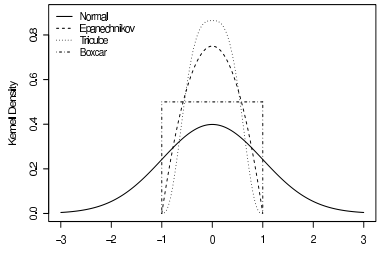
$\int K(u)\,du = 1$

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

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

Example Kernels

- *Gaussian* $K(x) = \frac{1}{2\pi} e^{-\frac{|x|}{2}}$
- *Epanechnikov* $K(x) = \frac{3}{4}(1-x)^2 I(x)$ ← ind. on -1, 1
- *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:

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

$$\hat{f} = L_\lambda y$$

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

Nadaraya-Watson Estimator

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

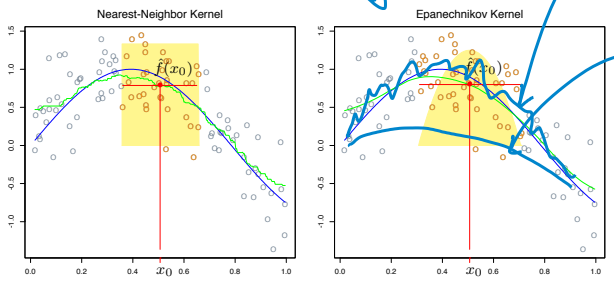
- Example:
  - Boxcar kernel → local avgs
  - Epanechnikov
  - Gaussian

typical

- Often, choice of kernel matters much less than choice of λ

small λ, low bias, high var

large λ, high bias, low var



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:

$\hat{f} = \arg\min_a \sum (y_i - a)^2$

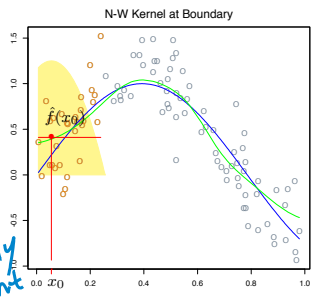
$\rightarrow \hat{f} = \bar{Y}$

$K\left(\frac{|x_i - x|}{\lambda}\right)$

$\hat{f}(x) = \arg\min_a \sum w_i(x)(y_i - a)^2$

restrict to locally constant

$\rightarrow \hat{f}(x) = \frac{\sum w_i(x) y_i}{\sum w_i(x)}$



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 :

$$\beta_{0x_0} + \beta_{1x_0}(x - x_0)$$

- Minimize:

$$\min_{\beta_{x_0}} \sum_i K_\lambda(x_0, x_i)\left(y_i - \beta_{0x_0} - \beta_{1x_0}(x_i - x_0)\right)^2$$

- Return: $\hat{f}(x_0) = \hat{\beta}_{0x_0}$ ← fit at x_0

Note: not equivalent to fitting a local constant!

- 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

$$(y - X_{x_0}\beta_{x_0})^T W_{x_0}(y - X_{x_0}\beta_{x_0})$$

$$W_{x_0} = \begin{bmatrix} K_\lambda(x_0, x_1) & & \\ & \ddots & \\ & & K_\lambda(x_0, x_n) \end{bmatrix}$$

$$X_{x_0} = \begin{bmatrix} 1 & x_1 - x_0 \\ \vdots & \vdots \\ 1 & x_n - x_0 \end{bmatrix}$$

- Solution:

$$\hat{\beta}_{x_0} = (X_{x_0} W_{x_0} X_{x_0})^{-1} X_{x_0}^T W_{x_0} y$$

$$\hat{f}(x_0) = e_1^T \hat{\beta}_{x_0}$$ ($e_1 = [1\ 0]$ grabs out 1st element)

$$= \sum_i l_i(x) y_i$$ (modified kernel accounting for LS operations)

Local Linear Regression

- Bias calculation:

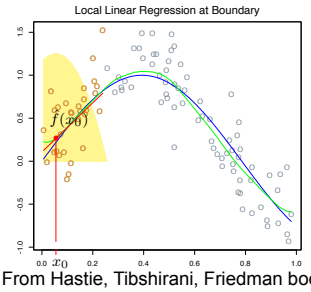
$$E[\hat{f}(x_0)] = \sum_i \ell_i(x_0) f(x_i)$$

$= f(x_0) \sum \ell_i(x_0) + f'(x_0) \sum (x_i - x_0)\ell_i(x_0) + \frac{f''(x_0)}{2} \sum (x_i - x_0)^2 \ell_i(x_0) + R$ ← higher order terms

$= f(x_0) + f''(x_0) \cdots$

- 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

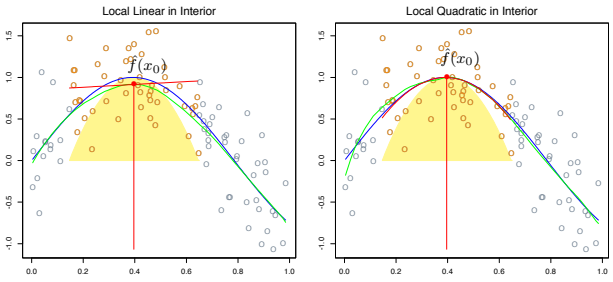


From Hastie, Tibshirani, Friedman book

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

$$P_{x_0}(x;\beta_{x_0}) = \beta_{0x_0} + \beta_{1x_0}(x-x_0) + \frac{\beta_{2x_0}}{2!}(x-x_0)^2 + \cdots + \frac{\beta_{dx_0}}{d!}(x-x_0)^d$$

- 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:

$$\min_{\beta_{x_0}} (y - X_{x_0}\beta_{x_0})^T W_{x_0} (y - X_{x_0}\beta)$$

$$X_{x_0} = \begin{bmatrix} 1 & x_1 - x_0 & \cdots & \frac{(x_1-x_0)^d}{d!} \\ \vdots & & & \\ 1 & x_n - x_0 & \cdots & \frac{(x_n-x_0)^d}{d!} \end{bmatrix}$$

- Return: $\hat{f}(x_0) = \hat{\beta}_0 x_0$
- 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 $X_1, \ldots, X_n \overset{iid}{\sim} P$
- Choice #1: empirical estimate? $\hat{P} = \frac{1}{n} \sum \delta_{X_i}$

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

$$\hat{p}(x_0) = \frac{\#x_i \in \text{Nbhd}(x_0)}{n\lambda}$$

λ ← width of nbhd

- Choice #3: again, consider kernel weightings instead

$$\hat{p}(x_0) = \frac{1}{n\lambda} \sum K_\lambda(x_0, x_i)$$

Parzen est.

Kernel Density Estimation

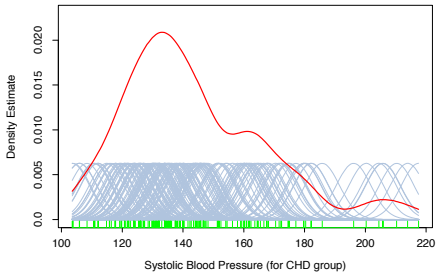
- Popular choice = Gaussian kernel → ***Gaussian KDE***

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

ϕ_λ

$$= (\hat{p} * \phi_\lambda)(x)$$

↑ empirical dist.



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 $x_1, \ldots, x_n \sim p$

$$E[\hat{p}^{\lambda}(x)] = \frac{1}{n\lambda}\sum_{i=1}^{n} E\left[K\left(\frac{x - x_i}{\lambda}\right)\right] = \frac{1}{n\lambda}\sum_{i=1}^{n}\int K\left(\frac{x-t}{\lambda}\right)p(t)\,dt$$

$$= \frac{1}{\lambda}\int K\left(\frac{x-t}{\lambda}\right)p(t)\,dt = \left(\lambda^{-1}K_{\lambda} * p\right)(x)$$

↑ true density

- 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

$\lambda \to 0$ as $n \to \infty$

"λ_n"

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, $p''(x)$ abs. cont. it's straightforward to show that

$$E[\hat{p}^{\lambda}(x)] = p(x) + \frac{1}{2}\lambda_n^2 p''(x)\sigma_k^2 + O(\lambda_n^2)$$

asy. unbiased — as $n \to \infty$ if $\lambda_n \to 0$, then $(\cdot) \to 0$

  - In peaks, negative bias and KDE underestimates p $(p''(x) < 0)$
  - In troughs, positive bias and KDE over estimates p $(p''(x) > 0)$
  - Again, "trimming the hills" and "filling the valleys"
- For var→0, require $n\lambda_n \to \infty$ $O(n^{-4/5})$
- 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'' (smoothness of target density)

Connecting KDE and N-W Est.

- Recall task:

$$f(x) = E[Y \mid x] = \int y p(y \mid x) dy = \frac{1}{p(x)} \int y\, p(x,y) dy$$

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

$$\hat{p}^{\lambda_x,\lambda_y}(x,y) = \frac{1}{n\lambda_x\lambda_y} \sum_{i=1}^{n} K_x\left(\frac{x-x_i}{\lambda_x}\right) K_y\left(\frac{y-y_i}{\lambda_y}\right)$$

- Estimate margin *p(x)* by

$$\hat{p}^{\lambda_x}(x) = \frac{1}{n\lambda_x} \sum K_x\left(\frac{x-x_i}{\lambda_x}\right)$$

Connecting KDE and N-W Est.

- Then,

$$\hat{f}(x) = \frac{\frac{1}{n\lambda_x\lambda_y} \sum \int y\, K_x\left(\frac{x-x_i}{\lambda_x}\right) K_y\left(\frac{y-y_i}{\lambda_y}\right) dy}{\frac{1}{n\lambda_x} \sum K_x\left(\frac{x-x_i}{\lambda_x}\right)}$$

$$= \frac{\sum K_x(\;) \int (y_i + u\lambda_y) K_y(u)\, du}{\sum K_x(\;)}$$

$$= \frac{\sum K_x\left(\frac{x-x_i}{\lambda_x}\right) y_i}{\sum K_x\left(\frac{x-x_i}{\lambda_x}\right)}$$

↖ use $\int uK(u)du = 0$, $\int K(u)du = 1$

- Equivalent to Naradaya-Watson weighted average estimator

Module 2: Splines and Kernel Methods

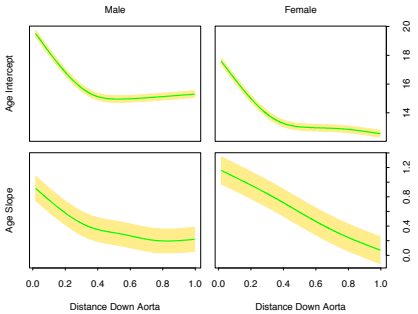
Inference for Linear Smoothers

STAT/BIOSTAT 527, University of Washington
Emily Fox
April 18th, 2013

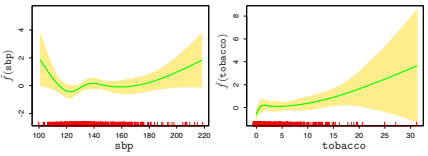
Confidence Bands

- So far we have focused on point estimation: $\hat{f}(x)$
- Often, we want to define a ***confidence interval*** for which $f(x)$ is in this interval with some pre-specified probability
- Looking over all *x*, we refer to these as ***confidence bands***

homoscedastic $\sigma(x) = \sigma$



heterosedastic $\sigma(x)$



From Hastie, Tibshirani, Friedman book

Bias Problem

- Typically, these are of the form
$$\hat{f}(x) \pm c \ \mathrm{se}(x)$$
est. of st. dev. of $\hat{f}(x)$
- This is really not a confidence band for $f(x)$, but for
$$\bar{f}(x) = E[\hat{f}(x)]$$
- In parametric inference, these are normally equivalent
- More generally,
$$\frac{\hat{f}(x) - f(x)}{s(x)} = \frac{\hat{f}_n(x) - \bar{f}(x)}{s(x)} - \frac{\bar{f}_n(x) - f(x)}{s(x)}$$
$$= Z_n(x) + \frac{\mathrm{bias}(\hat{f}(x))}{\sqrt{\mathrm{var}(\hat{f}(x))}}$$
st. dev. of $\hat{f}(x)$

Bias Problem

$$\frac{\hat{f}(x) - f(x)}{s(x)} = Z_n(x) + \frac{\mathrm{bias}(\hat{f}(x))}{\sqrt{\mathrm{var}(\hat{f}(x))}}$$

- Typically, $Z_n(x)$ → standard normal

bias → 0 faster var

- In parametric inference, 2nd term normally → 0 as *n* increases
- In nonparametric settings,
  - optimal smoothing = balance between bias and variance
  - 2nd term does *not* vanish, even with large *n*

est. this is harder than est. f!

Hard. Lead term is $f''(x)$

- So, what should we do?
  - Option #1: Estimate the bias
  - Option #2: Live with it and just be clear that the CI's are for $\bar{f}(x)$ not $f(x)$

CIs for Linear Smoothers

- For linear smoothers, and assuming constant variance $\sigma(x) = \sigma$

$$\hat{f}(x) = \sum_{i=1}^{n} \ell_i(x) y_i$$

$\hat{f}(x) = \sum_{i=1}^{n} \ell_i(x) f(x_i)$

$\text{var}(\hat{f}(x)) = \sigma^2 \|\ell(x)\|^2$

- Consider confidence band of the form

$CI(x) = \hat{f}(x) \pm c\,\hat{\sigma}\|\ell(x)\|$, $a \le x \le b$; $c > 0$; $\hat{\sigma}$: est. of σ

- Using this, let's solve for *c*

CIs for Linear Smoothers

- Based on approach of Sun and Loader (1994)
  - Case #1: Assume σ known

$$P(\bar{f}(x) \notin \text{CI}(x) \text{ for some } x \in [a,b]) = P\left(\max_{x\in[a,b]} \frac{|\hat{f}(x) - \bar{f}(x)|}{\sigma\|\ell(x)\|} > c\right)$$

$\hat{f}(x) = \sum \ell_i(x) y_i$, $\bar{f}(x) = \sum \ell_i(x) f(x_i)$

$$= P\left(\max_{x\in[a,b]} \frac{\sum \epsilon_i \ell_i(x)}{\sigma\|\ell(x)\|} > c\right) = P\left(\max_{x} |W(x)| > c\right)$$

$$W(x) = \sum_i Z_i T_i(x) \quad Z_i = \frac{\epsilon_i}{\sigma} \sim N(0,1) \quad T_i(x) = \frac{\ell_i(x)}{||\ell(x)||}$$

- Good news: max of GP is well studied!

Gauss. process

$$P(\max_x |\sum_i Z_i T_i(x)| > c) \approx 2(1-\phi(c)) + \frac{\kappa_0}{\pi} e^{\frac{-c^2}{2}}$$

"Tube formula"

$\kappa_0 = \int_a^b \|T'(x)\| dx$

move later

- Assuming confidence level α, set equal to α and solve for c

CIs for Linear Smoothers

- Based on approach of Sun and Loader (1994)
  - Case #2: Assume σ unknown use est. $\hat{\sigma}$
  - Case #3: Assume $\sigma(x)$ non-constant

$$\text{var}(\hat{f}(x)) = \sum_i \sigma^2(x_i)\, l_i^2(x)$$

$$\text{CI}(x) = \hat{f}(x) \pm c \sqrt{\sum_i \sigma^2(x_i)\, l_i^2(x)}$$

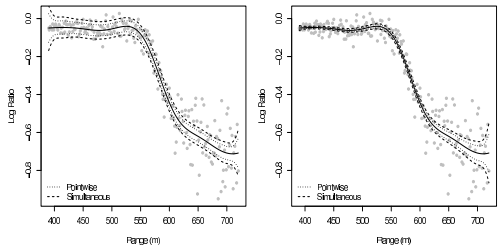
  - If $\hat{\sigma}(x)$ varies slowly with *x*, then (Faraway and Sun 1995)

$\sigma(x_i) \approx \sigma(x)$ for those x w/ $l_i(x)$ large

$\Rightarrow CI(x) = \hat{f}(x) \pm c\,\hat{\sigma}(x)\,\|l(x)\|$

CIs for Linear Smoothers

- Example from Wakefield textbook
  - Fit penalized cubic regression spline (penalty on trunc. power basis coef.)
  - For $\alpha = 0.05$, we calculate $c \approx 3.11$ ($\kappa_6 \approx 30$)
  - Estimate both constant and non-constant variance



- Notes: Ignored uncertainty introduced by choice of λ
  - Restrict search to finite set and do Bonferroni correction $\alpha \to \frac{\alpha}{m}$ (# of λ)
  - Sophisticated bootstrap techniques
  - Bayesian approach treats λ as a parameter with a prior and averages over uncertainty in λ for subsequent inferences