Semiparametric and Nonparametric Methods in Political Science

Lecture 2: Density Estimation

Michael Peress, University of Rochester and Yale University

Overview of Density Estimation

- Nonparametric Model: statistical model characterized by infinite dimensional unknown parameter
  - $y_n \sim f$ where f is unknown (density estimation)
  - $y_n = g(x_n) + \varepsilon_n,\ E[\varepsilon_n \mid x_n] = 0$ (nonparametric regression)
  - $\Pr(y_n = 1) = G(x_n)$ (nonparametric binary choice)

- Unlike the "easy" semiparametric estimators we covered in Lecture 1, the nonparametric and semiparametric estimators we study in Lecture 3 will be "hard"'
- Lecture 2 provides the background for these problem by studying one problem- density estimation using kernel methods- in great detail

Overview of Density Estimation

- Kernel techniques generalize to problems beyond density estimation
- What we learn about nonparametric problems from kernel techniques generalize to alternative estimators such as:
  - k-nearest neighbor estimators (also called “matching” estimators)
  - Smoothing splines
  - Sieve estimators (estimation using orthogonal functions such as polynomials or Fourier series)
  - Histogram estimators (for density estimation)

Kernel Density Estimation

- The Density Estimation Problem:
  - We assume that $\{X_n\}_{n=1}^N$ are i.i.d. draws from a common distribution $f_0(x)$
  - The density estimation problem is the problem of estimating $f_0(x)$ while placing only minimal restrictions on f_0
  - We would like to develop an estimator $\hat{f}$ of the density f_0

Kernel Density Estimation

- The Kernel Density Estimator (KDE) is defined by,

$$\hat{f}(x;h) = \frac{1}{hN}\sum_{n=1}^{N} K\left(\frac{X_n - x}{h}\right)$$

- Here, K denotes the kernel and h denotes the bandwidth
- The Kernel satisfies:

(i) $K(u) \geq 0$

(ii) $\int K(u)dx = 1$

(iii) $\int uK(u)du = 0$

(iv) $\int u^2 K(u)du > 0$

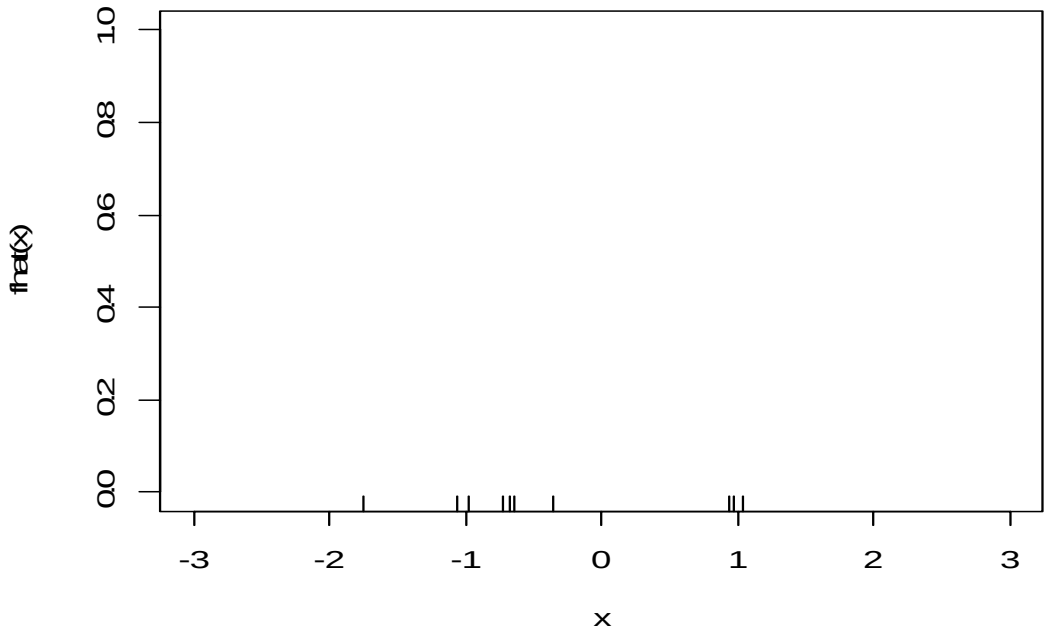
Kernel Density Estimation

- Examples of Kernels:

Name	$K(u)$
Uniform	$K(u) = \begin{cases} \frac{1}{2}, & -1 \le u \le 1 \\ 0, & otherwise \end{cases}$
Triangle	$K(u) = \begin{cases} 1-\lvert u \rvert, & -1 \le u \le 1 \\ 0, & otherwise \end{cases}$
Epanechnikov	$K(u) = \begin{cases} \frac{3}{4}(1-u^2), & -1 \le u \le 1 \\ 0, & 0 \end{cases}$
Gaussian	$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$

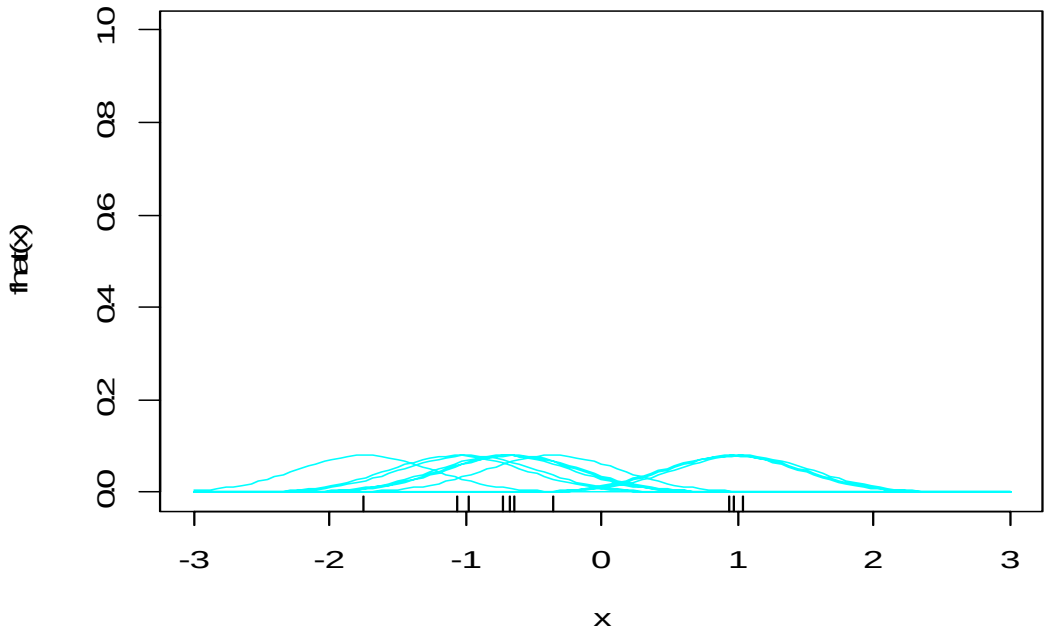
Kernel Density Estimation

- Example w/ N=10 Data Points – Rug Plot:



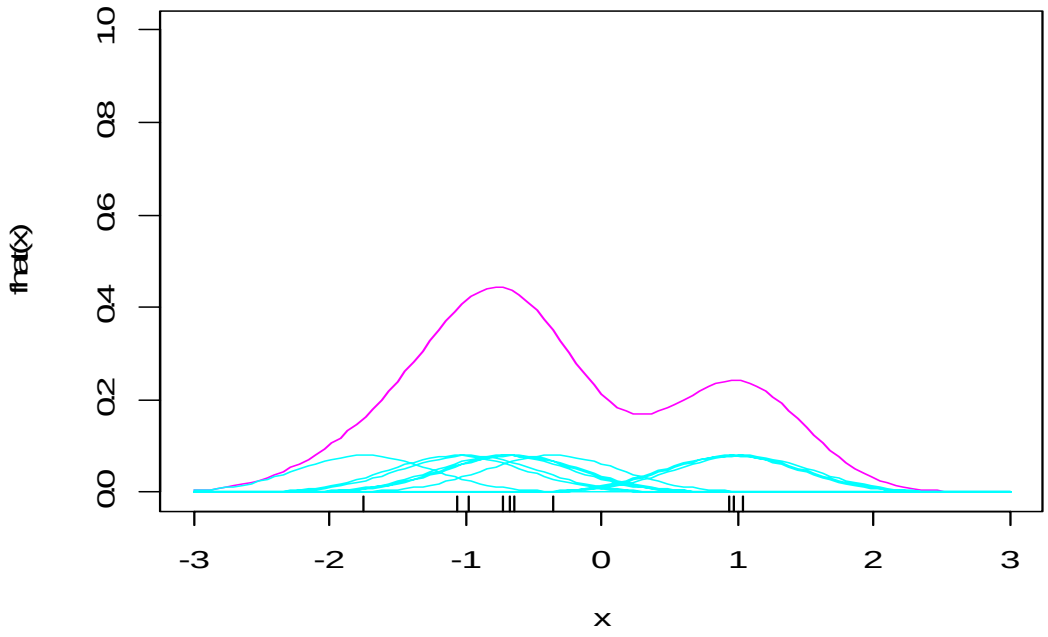
Kernel Density Estimation

- Example w/ N=10 Data Points – Individual Kernels (h=.5):



Kernel Density Estimation

- Example w/ N=10 Data Points – Density Estimate (h=.5):

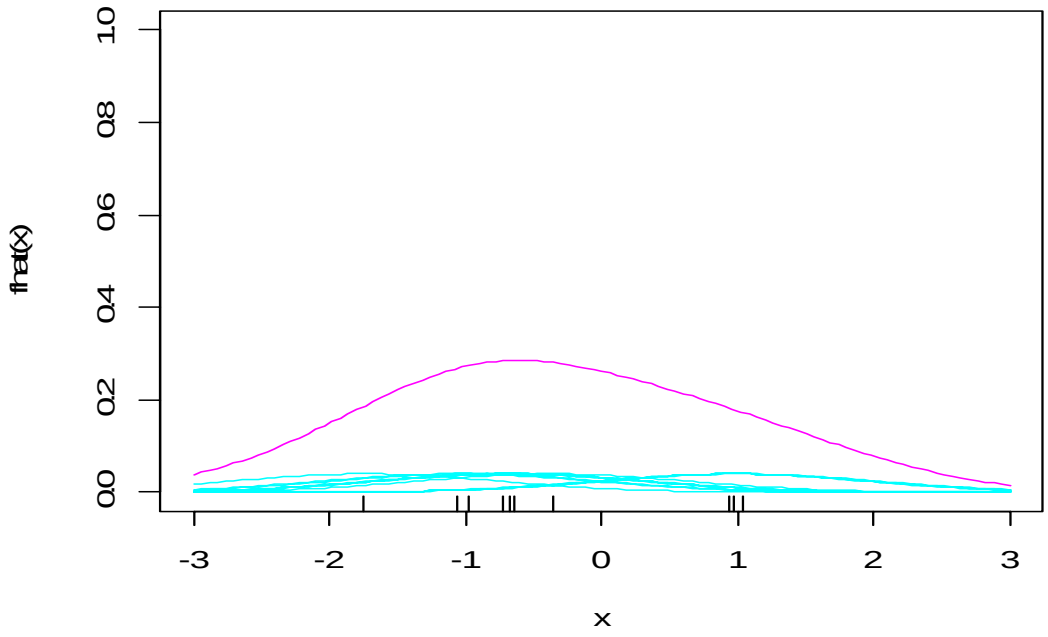


Kernel Density Estimation

- The bandwidth h controls the amount of smoothing
  - Large values of h denote a large degree of smoothing
  - Small values of h denotes a small degree of smoothing

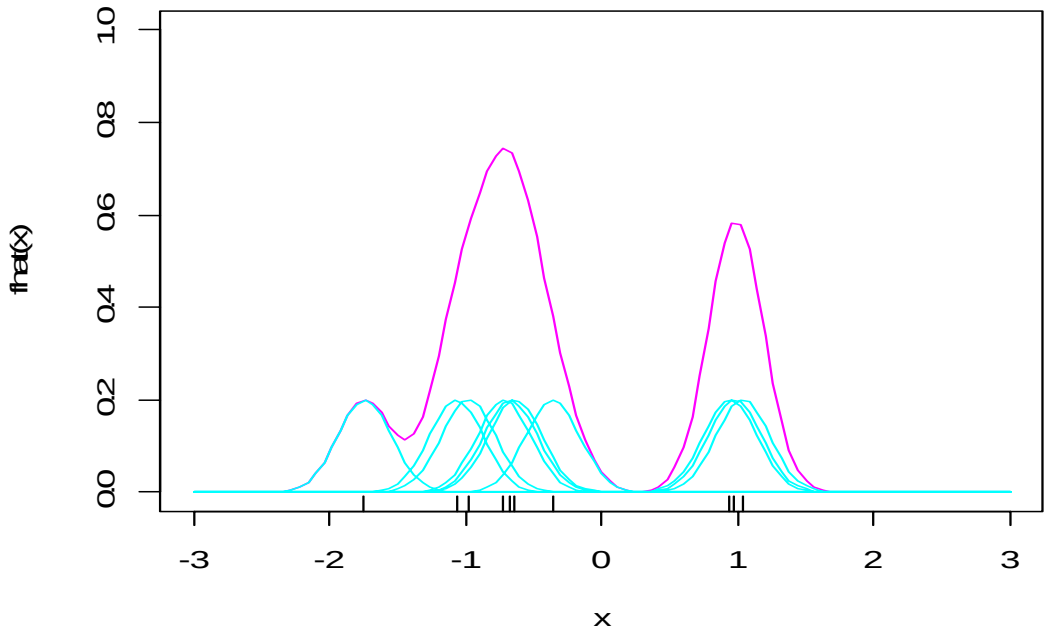
Kernel Density Estimation

- Example w/ N=10 Data Points – Density Estimate (h=1):



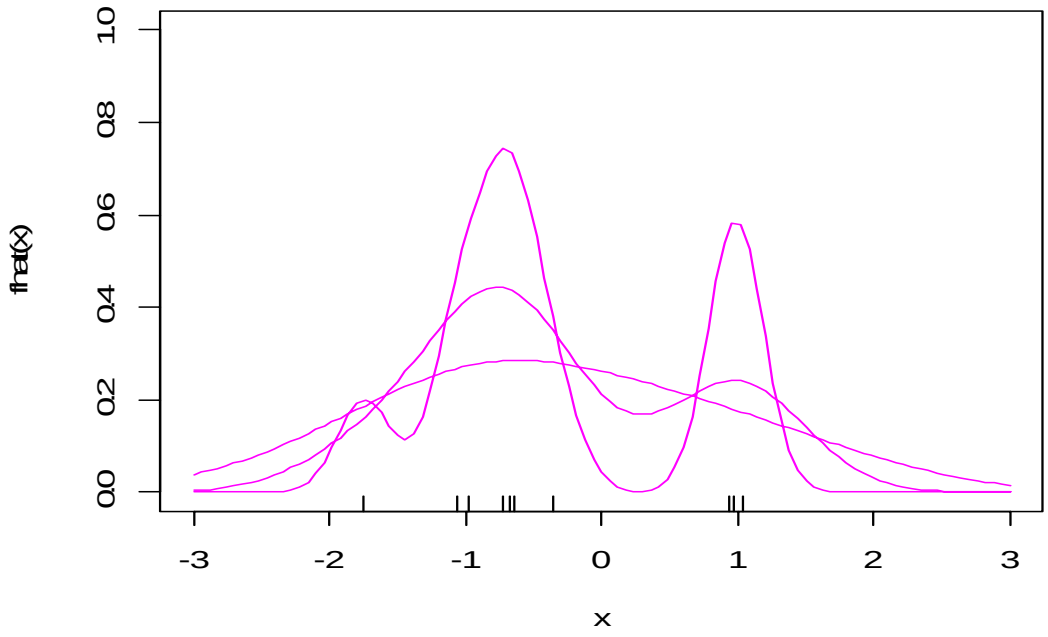
Kernel Density Estimation

- Example w/ N=10 Data Points – Density Estimate (h=.2):



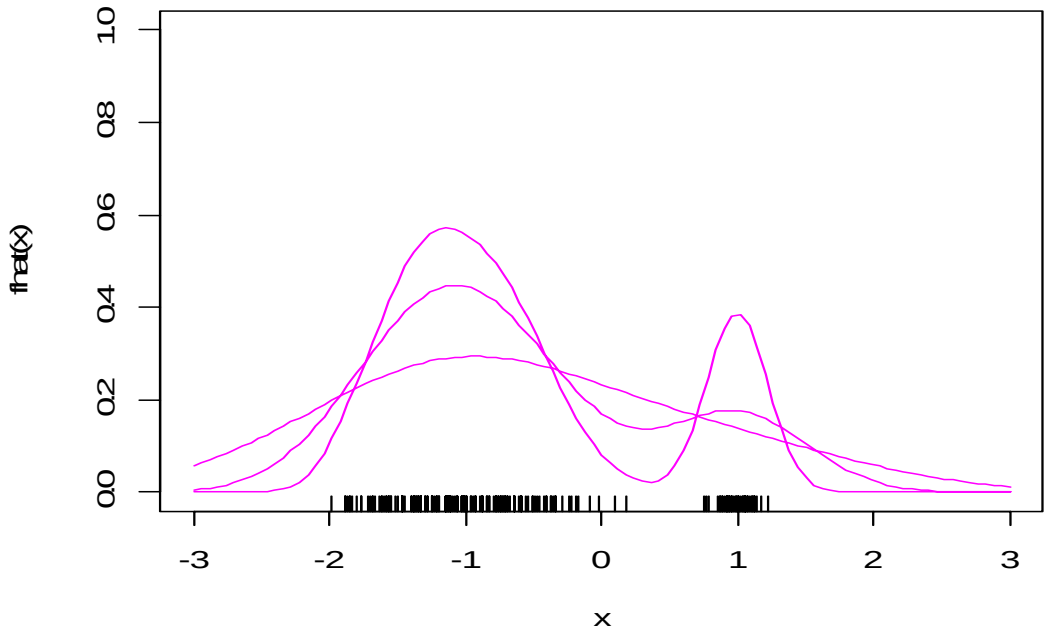
Kernel Density Estimation

- Example w/ N=10 Data Points – Density Estimate (h=.2,.5, and 1):



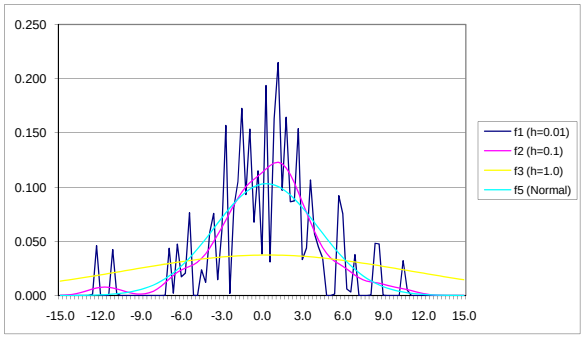
Kernel Density Estimation

- Example w/ N=200 Data Points – Density Estimate (h=.2,.5, and 1):



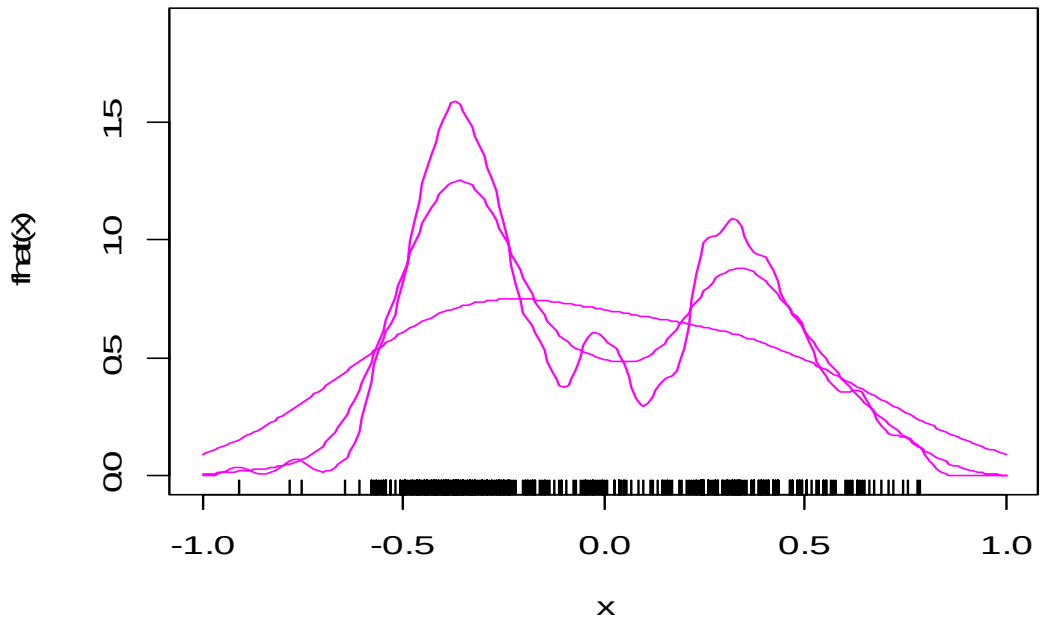
Kernel Density Estimation

- Example: Dow Jones returns for the period 1/1/01 through 12/1/07, using kernel density estimators (with different bandwidths) and an estimated normal density



Kernel Density Estimation

- Example: Positions of Senate incumbents (h=.3,.1, and 3)



Finite Sample Properties of KDEs

- Bias:

$$Bias[\hat{f}(x;h)] = E[\hat{f}(x;h)] - f_0(x) = E\left[\frac{1}{hN}\sum_{n=1}^{N} K\left(\frac{X_n - x}{h}\right)\right] - f_0(x)$$

$$= \frac{1}{hN}\sum_{n=1}^{N} E[K\left(\frac{X_n - x}{h}\right)] - f_0(x) = \frac{1}{h}E[K\left(\frac{X_n - x}{h}\right)] - f_0(x)$$

$$= \int_{x'}\left[\frac{1}{h}K\left(\frac{x' - x}{h}\right)\right] f_0(x')dx' - f_0(x)$$

- First part of the expression is a local weighted average, with weighting function $w(x' \mid x) = \frac{1}{h}K\left(\frac{x' - x}{h}\right)$
- Small values of h will mean that the weighting function will be concentrated around x, meaning that there will be little bias
- Large values of h means that we are counting values distant from x in the average, leading to large bias

Finite Sample Properties of KDEs

- Variance:

$$Var(\hat{f}(x;h)) = Var\left(\frac{1}{hN}\sum_{n=1}^{N} K\left(\frac{X_n - x}{h}\right)\right) = \frac{1}{h^2 N} Var\left(K\left(\frac{X_n - x}{h}\right)\right)$$

$$= \frac{1}{h^2 N}\int_{x'}\left(K\left(\frac{x'-x}{h}\right) - E\left[\frac{1}{h}K\left(\frac{X_n - x}{h}\right)\right]\right)^2 f_0(x')dx'$$

- The h^{-2} terms suggest that the variance decreases as h increases
- Intuitively, we are averaging a large number of quantities when h is large, so we should expect the variance to decreases.

Finite Sample Properties of KDEs

- Taylor series expansions give more precise characterization of bias and variance
  - Use change of variables $u = \frac{y-x}{h}$

$$Bias[\hat{f}(x;h)] = \int_{x'} \left[\frac{1}{h} K\left(\frac{x'-x}{h}\right)\right] f_0(x')dx' - f_0(x)$$

$$= \int_u K(u) f_0(x+hu)du - f_0(x)$$

  - Now, we will employ the fourth-order Taylor expansion,

$$f_0(x+hu) = f_0(x) + f_0'(x)hu + \tfrac{1}{2} f_0''(x)h^2u^2 + \tfrac{1}{6} f_0'''(x)h^3u^3 + o(h^3)$$

terms smaller than h^3

Finite Sample Properties of KDEs

- Define $\mu_2 = \int_u u^2 K(u)du$ and $\nu_2 = \int_u K^2(u)du$
- We have,

$$Bias[\hat{f}(x;h)]$$

$$= \int_u K(u)[f_0(x) + f_0'(x)hu + \tfrac{1}{2} f_0''(x)h^2u^2 + \tfrac{1}{6} f_0'''(x)h^3u^3 + o(h^3)]du - f_0(x)$$

$$= f_0(x)\underbrace{\int_u K(u)du}_{=1} + hf_0'(x)\underbrace{\int_u uK(u)du}_{=0} + \tfrac{1}{2}h^2 f_0''(x)\underbrace{\int_u u^2K(u)du}_{=\mu_2}$$

$$+ \tfrac{1}{6}h^3 f_0'''(x)\underbrace{\int_u u^3K(u)du}_{=0} - f_0(x) + o(h^3)$$

$$= \tfrac{1}{2}\mu_2 h^2 f_0''(x) + o(h^3)$$ (bias goes to 0 as h goes to 0)

Finite Sample Properties of KDEs

- We can characterize the variance similarly,

$Var(\hat{f}(x;h)) = \frac{1}{Nh}\nu_2 f_0(x) + O(N^{-1})$ (variance goes to 0 as Nh goes to ∞)

Selecting the Bandwidth

- Holding N fixed, bias increases with h while variance decreases
- Select h to minimize integrated mean-squared error,

$$MSE(\hat{f}(x;h)) = Var(\hat{f}(x;h)) + Bias[\hat{f}(x;h)]^2$$

$$= \tfrac{1}{Nh}\nu_2 f_0(x) + \tfrac{1}{4}\mu_2{}^2 h^4 f_0''(x)^2 + O(N^{-1}) + o(h^5)$$

$$IMSE(\hat{f};h) = \tfrac{1}{Nh}\nu_2 + \tfrac{1}{4}h^4\mu_2{}^2\int_x f_0''(x)^2\,dx + O(N^{-1}) + o(h^5)$$

- Theoretical bandwidth that minimizes IMSE (obtained via FOC):

$$h^* = \left(\frac{\nu_2}{\mu_2{}^2\int_x f_0''(x)^2\,dx}\right)^{1/5} N^{-1/5}$$

Selecting the Bandwidth

- Naturally, we would like $IMSE(\hat{f};h) \to 0$ (which is a property weaker than consistency)
- We require $h \to 0$ and $Nh \to 0$ as $N \to \infty$
- $h*$ will clearly satisfy the three asymptotic conditions
- This result tells us that rate at which to increase h to obtain optimal results, but we still need a way to determine the constant
- Notice that ν_2 and μ_2 can be computed easily
- Must have estimate of $\int_x f_0''(x)^2 dx$
- Estimating $f_0(x)$ requires estimating $\int_x f_0''(x)^2 dx$!

Selecting the Bandwidth

- Constants Characterizing Asymptotic Distribution:

Name	$K(u)$	$\mu_2 = \int_u u^2 K(u)du$	$\nu_2 = \int_u K^2(u)du$
Uniform	$K(u) = \begin{cases} \frac{1}{2}, & -1 \le u \le 1 \\ 0, & otherwise \end{cases}$	$\frac{1}{3}$	$\frac{1}{2}$
Triangle	$K(u) = \begin{cases} 1-\lvert u \rvert, & -1 \le u \le 1 \\ 0, & otherwise \end{cases}$	$\frac{1}{6}$	$\frac{2}{3}$
Epanech.	$K(u) = \begin{cases} \frac{3}{4}(1-u^2), & -1 \le u \le 1 \\ 0, & 0 \end{cases}$	$\frac{1}{5}$	$\frac{3}{5}$
Gaussian	$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$	1	$\frac{1}{2\sqrt{\pi}}$

Selecting the Bandwidth

- Normal Reference Rule:
  - Compute $\int_x f_0''(x)^2 dx$ for some special density (in this case, normal) and apply it to our data
  - The normal reference rule involves assuming that $f_0(x)$ is the $N(\mu, \sigma^2)$ distribution
  - We have that $\int_x f_0''(x)^2 dx = \frac{3}{8\sigma^5\sqrt{\pi}}$
  - Normal reference rule (or rule-of-thumb bandwidth) suggests,

$$h = \left(\frac{\nu_2 8\sqrt{\pi}}{3\mu_2^{\ 2}} \right)^{1/5} \sigma N^{-1/5} = c\sigma N^{-1/5}$$

Selecting the Bandwidth

- For the normal kernel, we can determine that $c = \left(\frac{4}{3}\right)^{1/5} \approx 1.059$, so that,

$$h = 1.059\sigma N^{-1/5}$$

- Other kernels will yield different constants. To estimate σ, we could use the variance of the data
- Silverman (1986) suggests employing a robust estimator,

$$\hat{\sigma} = \min\{s, 1.34(q_{0.75} - q_{0.25})\}$$

where $q_{0.25}$ and $q_{0.75}$ represent the 25 and 75% quantiles

Selecting the Bandwidth

- Plug-In Method:
  - Estimate $\int_x f_0''(x)^2 dx$ rather than guessing it
  - Use normal reference rule to obtain an initial kernel density estimator, $\hat{f}(x)$
  - Then, we can use this to approximate $\int_x f_0''(x)^2 dx$ by taking a second numerical derivative of $\hat{f}(x)$ and integrating
  - We then re-estimate $f_0(x)$ using the new bandwidth.
  - More sophisticated approach: iterating the plug-in rule to convergence, or solving for h as a nonlinear system

$$h - \left(\frac{\nu_2}{\mu_2^2 \int_x f_0''(x)^2 dx(h)} \right)^{1/5} N^{-1/5} = 0$$

Selecting the Bandwidth

- Cross Validation:
  - Obtain an estimate of the integrated mean-squared error as a function of h, and minimize it
  - The actual integrated mean squared error is,

$$IMSE(h) = \int_x (\hat{f}(x;h) - f_0(x))^2 dx$$

$$= \int_x \hat{f}^2(x;h)dx + \int_x f_0^2(x)dx - 2\int_x \hat{f}(x;h) f_0(x)dx$$

  - Estimate objective function using,

$$\hat{J}(h) = \frac{1}{N^2 h}\sum_{n=1}^{N}\sum_{m=1}^{N}(K \circ K)\left(\frac{x_n - x_m}{h}\right) - 2\frac{1}{Nh(N-1)}\sum_{n=1}^{N}\sum_{m\neq n} K\left(\frac{x_n - x_m}{h}\right)$$

  - Here, $K \circ K$ denote the convolution of the kernel with itself and can be (tediously) computed analytically for a given choice of kernel

Selecting the Bandwidth

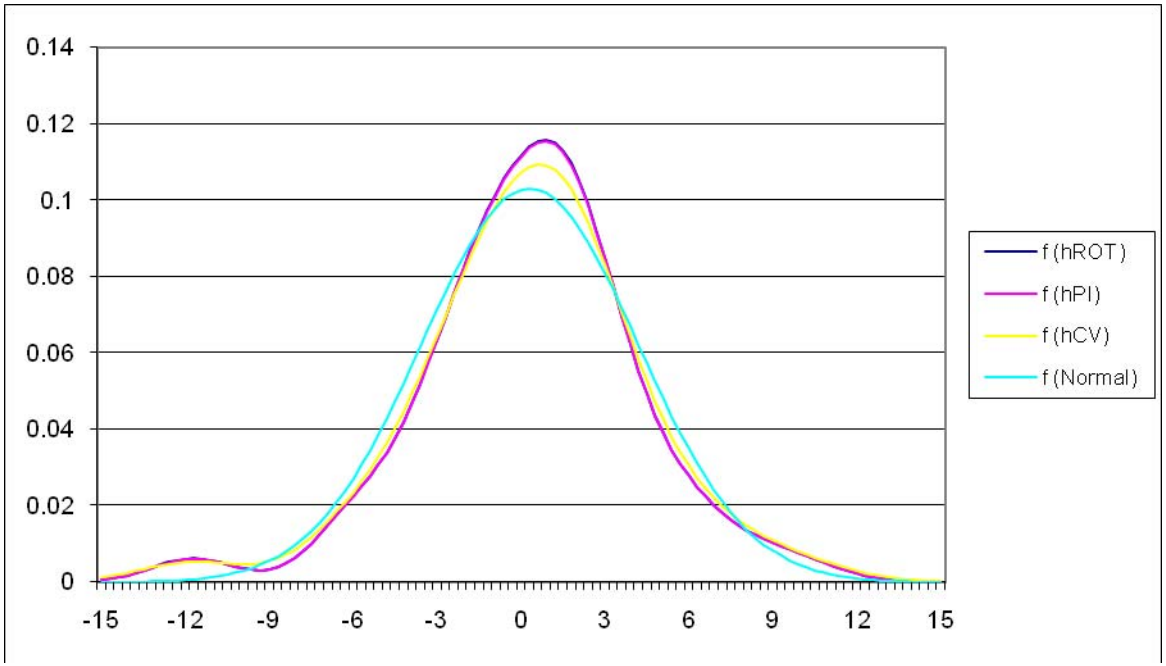
- The expression $\hat{J}(h)$ can then be numerically minimized to determine the cross validation bandwidth, h_{CV}
- One must be careful however, because if there are two data points such that $X_n = X_m$, then the cross validation function will have a minimum at 0
- A solution is to use,

$$\hat{J}(h) = \frac{1}{N^2 h}\sum_{n=1}^{N}\sum_{m=1}^{N}(K \circ K)\left(\frac{x_n - x_m}{h}\right) - 2\frac{1}{Nh(N-1)}\sum_{n=1}^{N}\sum_{x_m \neq x_n} K\left(\frac{x_n - x_m}{h}\right)$$

Selecting the Bandwidth

- Dow Jones Returns Example continued:
  - We can determine that,

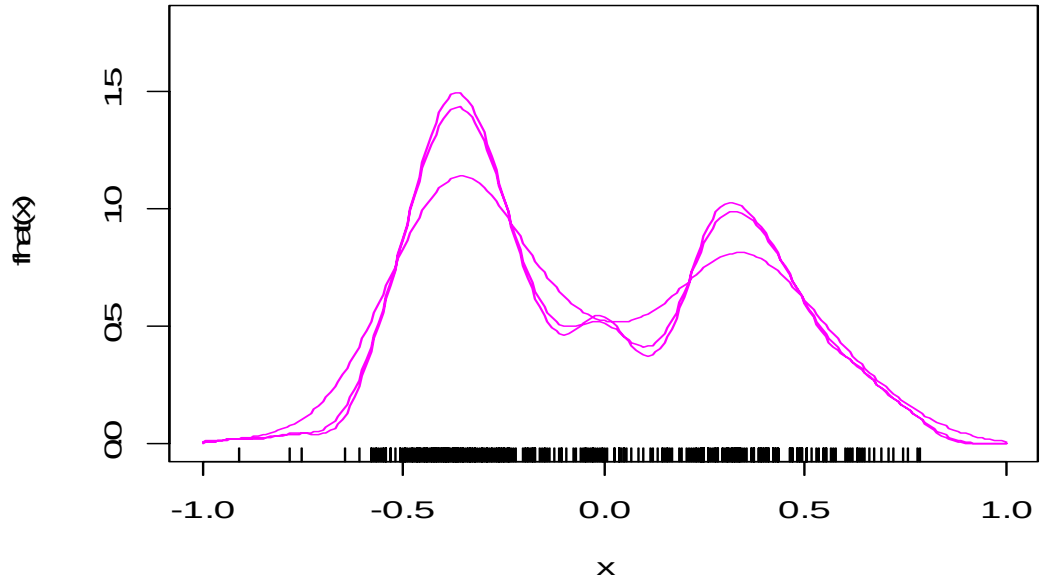
$$h_{ROT} = 1.371 \qquad h_{PI} = 1.381 \qquad h_{CV} = 1.739$$



Selecting the Bandwidth

- Senate Incumbent Positions example continued:
  - We can determine that,

$$h_{ROT} = 0.126 \qquad h_{PI} = 0.050 \qquad h_{CV} = 0.063$$



Selecting the Kernel

- Suppose that we plug the optimal bandwidth into the formula for the integrated mean squared error,

$$IMSE(\hat{f}, h) = \tfrac{5}{4}(\mu_2 \nu_2{}^2)^{2/5} \left(\int_x f''(x)^2 dx \right)^{1/5} N^{-4/5} + o(N^{-1})$$

- The efficiency of the Kernel therefore depends on the constant $\mu_2{}^{2/5} \nu_2{}^{4/5}$.
- We can choose K to solve the calculus of variations problem, minimize $IMSE(K)$ subject to constraints based on (i) through (iv)

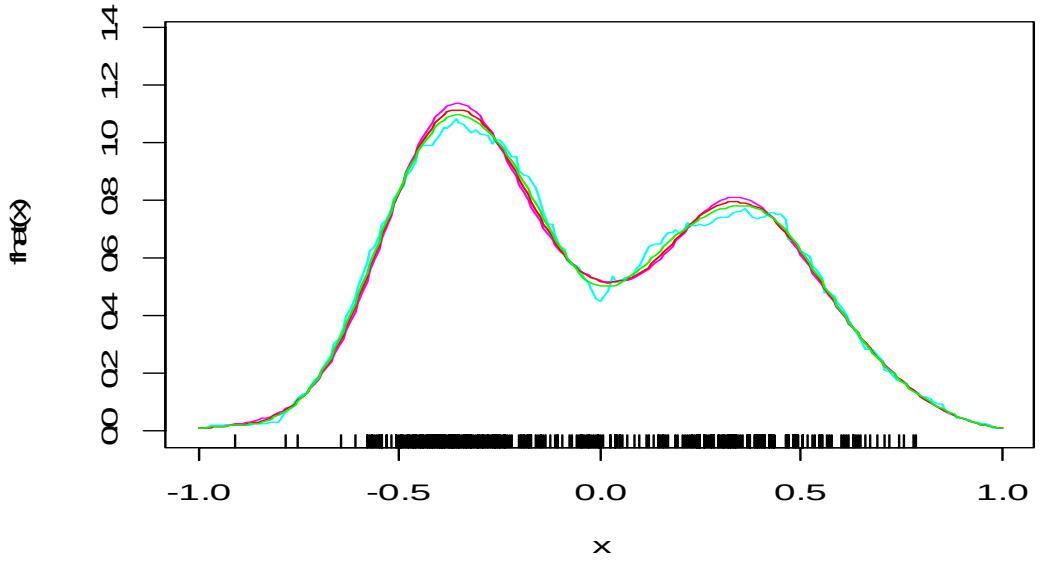
Selecting the Kernel

Name	μ_2	ν_2	Relative Efficiency
Uniform	$\frac{1}{3}$	$\frac{1}{2}$	1.060
Triangle	$\frac{1}{6}$	$\frac{2}{3}$	1.011
Epanechnikov	$\frac{1}{5}$	$\frac{3}{5}$	1.000
Gaussian	1	$\frac{1}{2\sqrt{\pi}}$	1.041

- Epanechnikov kernel is the most efficient, but the choice of a kernel in practice does not seem to matter much
- The effect of the kernel on mean-squared error is quite small. The popular normal kernel has an inefficiency of about 6%.

Selecting the Kernel

- Senate Incumbent Positions example continued (h=PI):



- Notice that of all the kernels, the Gaussian kernel is the only one that has full support
- Full support is one reason that the Gaussian kernel is often chosen

Selecting the Kernel

- Senate Incumbent Positions example continued (h=PI):
  - Warning: different Kernels require different bandwidths
  - In this case, normal=0.050, unif=0.035, triangular=0.103, Epanechnikov=0.087

Large Sample Properties of KDEs

- Like most parametric estimators, kernel density estimators are consistent and asymptotically normal
- They do, however, converge at a slower rate than parametric estimators
- Recall that,

$$Var(\hat{f}(x;h)) = \frac{1}{Nh}\nu_2 f_0(x) + O(N^{-1})$$

- This implies that the estimator converges at the rate $(Nh)^{-1/2}$ rather than the usual $N^{-1/2}$
- When an optimal bandwidth is selected, the convergence rate is $N^{-2/5}$, which is of course slower that $N^{-1/2}$

Large Sample Properties of KDEs

- Under the assumption that $h = h^*$, we can show that the kernel density estimator is asymptotically normally distributed in the following sense,

$$\sqrt{hN}\,(\hat{f}(x;h) - f_0(x)) \xrightarrow{dist.} N(\tfrac{1}{2}\mu_2 f_0''(x) h^{5/2} N^{-1/2}, \nu_2 f_0(x))$$

- Notice that the asymptotic distribution is not centered at zero because (by construction) the bias and variance are of the same magnitude
- We can eliminate the bias term by over-smoothing, selecting $h = cN^{-1/5+k}$ where $k > 0$

$$\sqrt{hN}\,(\hat{f}(x;h) - f_0(x)) \xrightarrow{dist.} N(0, \nu_2 f_0(x))$$

Inferences for KDEs

- Two major approaches to conducting inferences for kernel density estimators – asymptotic formulas vs. the bootstrap
- Inference based on asymptotic formulas:
  - Asymptotic distribution w/ optimal smoothing

$$\hat{f}(x;h) - \tfrac{1}{2}\mu_2 \hat{f}_0''(x)h^{5/2}N^{-1/2} \pm \sqrt{\nu_2 \hat{f}(x)} / \sqrt{hN}$$

  - Asymptotic distribution w/ under-smoothing

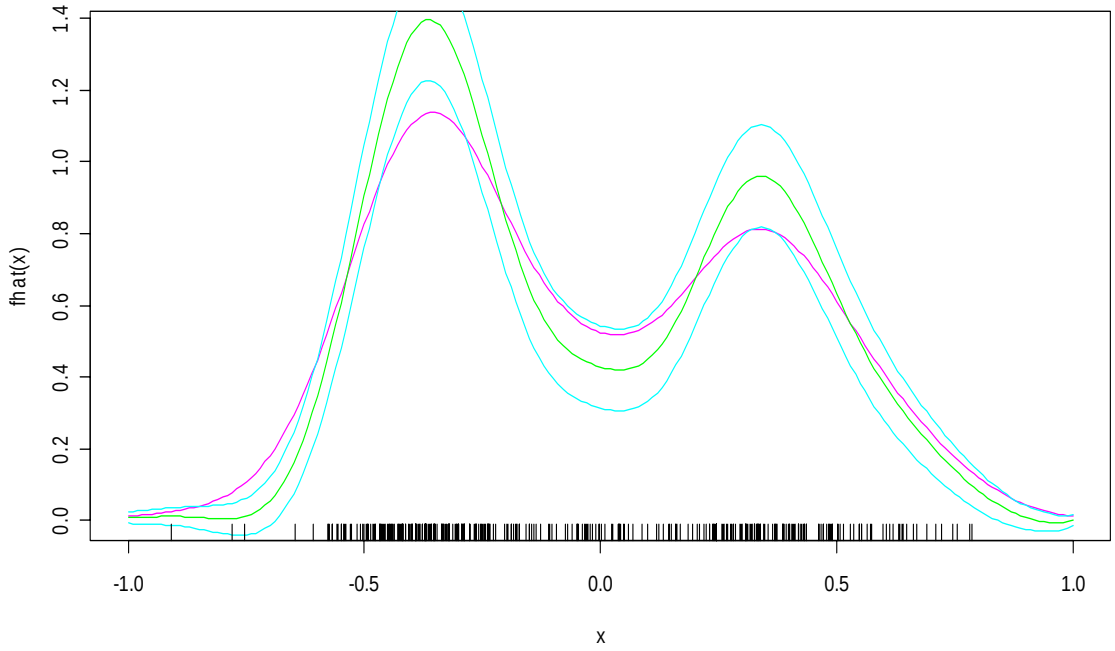
$$\hat{f}(x;h) \pm \sqrt{\nu_2 \hat{f}(x)} / \sqrt{hN}$$

Inferences for KDEs

- Inference based on the bootstrap:
  - We sample S draws, with replacement, form $\{X_n\}_{n=1}^{N}$.
  - To compute the 95% confidence interval of $\hat{f}(x;h)$, we simply take the 2.5% and 97.5% quantiles of the empirical distribution $\hat{f}_s(x;h)$

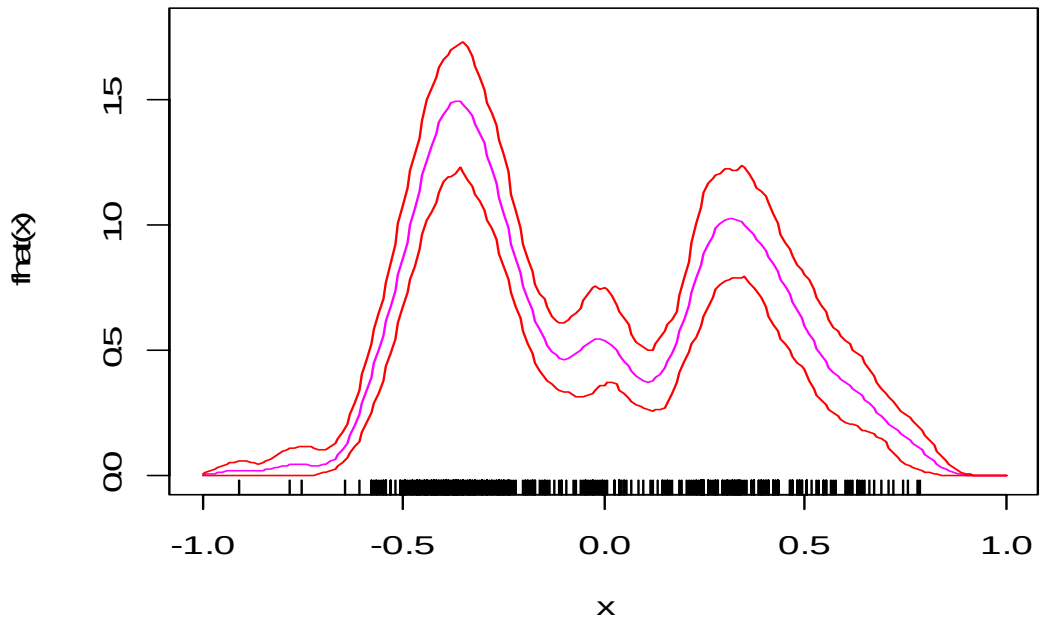
Inferences for KDEs

- Senate Incumbent Positions example continued (assymp. CI):



Inferences for KDEs

- Senate Incumbent Positions example continued (bootstrap CI):



Higher Order Kernels

- One can reduce the bias (and improve the convergence rate) of kernel density estimators by considering higher-order kernels
- These are kernels that have more even moments which are zero (the odd moments are always zero)
- This leads to more terms in the Taylor expansion canceling out
- It also means that we require f_0 to have more derivatives in characterizing the asymptotic distribution

Higher Order Kernels

- Properties of Higher Order Kernels:

$$Bias\left[\frac{1}{hN}\sum_{n=1}^{N}K\left(\frac{x_n-x}{h}\right)\right]=\frac{1}{(r+1)!}h^{r+1}\mu_{r+1}f_0^{(r+1)}(x)+o(h^{r+2})$$

$$Var\left(\frac{1}{hN}\sum_{n=1}^{N}K\left(\frac{x_n-x}{h}\right)\right)=\frac{1}{Nh}\nu_2 f_0(x)+o(N^{-1})$$

$$IMSE(h)=\frac{1}{Nh}\nu_2+(\frac{1}{(r+1)!})^2h^{2r+2}{\mu_{r+1}}^2\int_x(f_0^{(r+1)}(x))^2dx+o(N^{-1})+o(h^{2r+3})$$

$$h^*=\left(\frac{\nu_2}{(2r+2)(\frac{1}{(r+1)!})^2{\mu_{r+1}}^2\int_x(f_0^{(r+1)}(x))^2dx}\right)^{1/(2r+3)}N^{-1/(2r+3)}$$

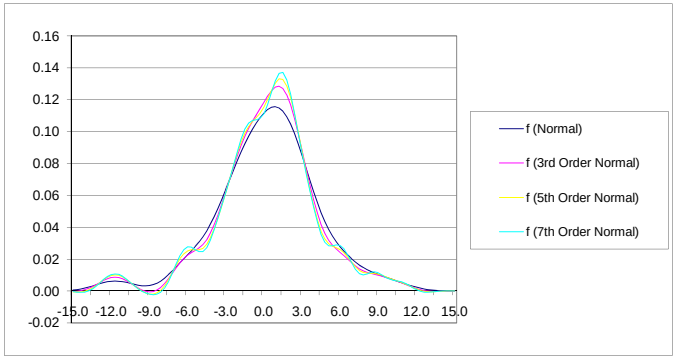
- Notice that when $r=1$, we have $h^*=O(N^{-1/5})$

Higher Order Kernels

- Next, notice that the order of the means square error is $o(N^{-1}h^{-1}) = o(N^{-1-1/(2r+3)})$
- Notice that this goes to $O(N^{-1})$ as $r \to \infty$, meaning that if the space of functions is sufficiently smooth, then we approximate the parametric rate

Higher Order Kernels

- Dow Jones Returns example continued:



Histogram Estimators

- An alternative to kernel density estimators are histogram estimators
  - Let h denote the bin size
  - Histogram estimator defined by,

$$\hat{f}(x;h) = \begin{cases} \frac{1}{Nh}\sum_{n=1}^{N}\sum_{i=1}^{m} 1\{h(i-1) \le x_n \le hi\}, & h(i-1) \le x \le hi \\ 0, & otherwise \end{cases}$$

  - Notice that,

$$Bias[\hat{f}(x;h)] = f_0(h(i-1)) - f_0(x) + \tfrac{1}{2} f_0'(h)h + o(h) \text{ for } h(i-1) \le x \le hi$$

$$Var(\hat{f}(x;h)) = \tfrac{1}{N} f_0(h(i-1))(1 - f_0(h(i-1)) + o(N^{-1}) \text{ for } h(i-1) \le x \le hi$$

$$IMSE(\hat{f}(x;h)) = \tfrac{1}{Nh} + \tfrac{h^2}{12}\int_x f'(x)^2 dx + o(h^2) + o(N^{-1})$$

Histogram Estimators

- Optimal bandwidth,

$$h_N^* = \left(\frac{6}{\int_x (f'(x))^2 dx} \right)^{1/3} N^{-1/3}$$

- Integrated mean squared error converges at rate $N^{-2/3}$
- The above result indicates that the histogram estimator converges at a slower rate than the kernel density estimator, but notice that we require fewer derivatives
- We can develop a parallel theory for histogram estimators that includes bin selection methods (including cross validation), asymptotic confidence intervals, etc.

Efficiency of Density Estimators

- Theorem (stated loosely): Consider the class of pdfs $\mathcal{F}_{m,M}$ such that the the mth derivative of f_0 exists and is bounded in total variation by M,

$$\mathcal{F}_{m,M} = \{ f \in \mathcal{F} : \int_x (f^{(m)}(x))^2 dx \leq M \}$$

  The optimal rate of convergence for the IMSE for any density estimator in this class is $N^{-2m/(2m+1)}$

- When $m = 1$, we have $N^{-2m/(2m+1)} = N^{-2/3}$, a bound which the Histogram estimator achieves
- When $m = 2$, we have $N^{-2m/(2m+1)} = N^{-4/5}$, a bound which the KDE achieves
- When $m = \infty$, we have N^{-1}, which is the parametric rate

Multivariate Density Estimation

- Consider now the problem of estimating a d-dimensional density $f_0(x)$
- Define the multivariate kernel density estimator to be,

$$\hat{f}(x;h) = \frac{1}{Nh^d} \sum_{n=1}^{N} \prod_{i=1}^{d} K\left(\frac{x_{n,i} - x_i}{h}\right)$$

- Bias:

$$Bias[\hat{f}(x;h)] = \frac{1}{2} h^2 \mu_2 \sum_{i=1}^{d} f_d\,''(x_1,...,x_d) + o(h^2)$$

- Variance:

$$Var(\hat{f}(x;h)) = \frac{1}{h^d N} f(x) \nu_2{}^d + o(N^{-1}h^{-d})$$

Multivariate Density Estimation

- IMSE:

$$IMSE(\hat{f}) = \frac{1}{h^d N}\nu_2{}^d + \frac{1}{4}\mu_2{}^2 h^4 \int_x \left(\sum_{i=1}^{d} f_d\,''(x) \right)^2 dx + o(h^4) + o(N^{-1}h^{-d})$$

- FOC for IMSE for optimal bandwidth,

$$h_N{}^* = \left(\frac{d\nu_2{}^d}{\mu_2{}^2 \int_x \left(\sum_{i=1}^{d} f_d\,''(x) \right)^2 dx} \right)^{1/(4+d)} N^{-1/(4+d)}$$

- Optimal bandwidth yields an IMSE with an error of size $N^{-4/(4+d)}$

Multivariate Density Estimation

- The rate of convergence decreases as d increases (curse of dimensionality!)
- Curse of dimensionality is not a drawback of KDEs, but a drawback of the nonparametric density estimation problem (i.e. KDEs achieve optimal rates under maintained assumptions about the derivatives of the density)
- No alternative estimator (k-NN, splines, etc.) can do better under maintained assumptions
- Same problem holds for kernel regression, kernel binary choice, etc.
- One solution: avoid fully nonparametric problems
- Estimators that combine parametric and nonparametric components are an attractive alternative (see Lecture 3)

Computational Tricks for KDEs

- Much “folk wisdom” in applying KDEs (and nonparametric estimators more generally)
- Here, we will cover some of the secret tricks often used
- Consider computation of density of Senate incumbent positions
- Load data in r:

```
library(xlsReadWrite) # load library
xls1 <-
read.xls("D:\\Teaching\\Spring_2010_Yale_Lecture\\sen
ate.xls", colNames=TRUE) # read data (change this to
the location on your hard drive)
N <- dim(xls1)[1] # sample size
X <- xls1$inc_pos # generate data
```

Computational Tricks for KDEs

- The kernel density estimator is an infinite dimensional quantity

$$\hat{f}(x;h) = \frac{1}{hN}\sum_{n=1}^{N} K\left(\frac{X_n - x}{h}\right)$$

- In practice, estimation means computing $\hat{f}(x;h)$ on a finite grid of points (typically equally spaced)

```
I <- 201 # grid size
xlow <- -1 # low point of grid
xhigh <- 1 # high point of grid
grid <- xlow+(xhigh-xlow)*(0:(I-1))/(I-1) # create
grid
```

Computational Tricks for KDEs

- Select the bandwidth using normal reference rule:

```
# select bandwidth using normal reference rule
hROT <- (nu2 * 8 * pi^.5)^.2 * (3 * mu2^2)^-.2 *
min(sd(X),IQR(X) / 1.34) * N^-.2 # normal reference
rule
```

Computational Tricks for KDEs

- Estimate kernel on a grid:

```
h=hROT # set bandwidth
ker1=matrix(rep(0,N*I),N) # allocate matrix
for(n in 1:N) ker1[n,1:I]=kerfunc((grid-
X[n])/h)/(N*h)
kerest1 <- rep(1,N) %*%  ker1
```

Computational Tricks for KDEs

- Asymptotic standard errors:

$$\hat{f}(x;h) - \tfrac{1}{2}\mu_2 \hat{f}_0''(x) h^{5/2} N^{-1/2} \pm \sqrt{\nu_2 f_0(x)} / \sqrt{hN}$$

- Requires estimating $f_0''(x)$:

- One approach,

$$\hat{f}''(x;h) = \tfrac{1}{hN} \sum_{n=1}^{N} K''\left(\tfrac{X_n - x}{h}\right)$$

- For Normal kernel,

$$\hat{f}''(x;h) = \tfrac{1}{hN} \sum_{n=1}^{N} \tfrac{4}{\sqrt{2\pi}} \left(\tfrac{X_n - x}{h}\right)^2 e^{-\left(\frac{X_n - x}{h}\right)^2} - \tfrac{1}{h^2 N} \sum_{n=1}^{N} \tfrac{2}{\sqrt{2\pi}} e^{-\left(\frac{X_n - x}{h}\right)^2}$$

- Optimal rate for h will be different, but $\hat{f}''$ estimate f_0'' consistently

Computational Tricks for KDEs

- Now consider estimating $\int_x f_0''(x)^2 dx$ using $\int_x \hat{f}''(x)^2 dx$, as is required for plug-in rule
- $\int_x \hat{f}''(x)^2 dx$ involves very messy expression
- Alternative, using finite difference approximations to derivatives and integrals

Computational Tricks for KDEs

- Discrete derivatives:

```
discrete_deriv <- function(x,f)
{
   n <- length(x)
   fp <- rep(n,0)
   fp[1] <- (f[2] - f[1]) / (x[2] - x[1])
   fp[2:(n-1)] <- (f[3:n]-f[1:(n-2)]) / (x[3:n]-
   x[1:(n-2)])
   fp[n] <- (f[n] - f[n-1]) / (x[n] - x[n-1])
   return(fp)
}
```

Computational Tricks for KDEs

- Discrete integrals:

```
discrete_int <- function(x,f)
{
   n <- length(x)
   F <- rep(0,n)
   for(i in 2:n) F[i] = F[i-1] + f[i-1]*(x[i]-x[i-1])
   return(F)
}
```

Computational Tricks for KDEs

- Select the bandwidth using plug-in rule:

```
plug_in <- function(h,N,I,grid,X)
{
   ker1=matrix(rep(0,N*I),N)
   for(n in 1:N) ker1[n,1:I]=kerfunc((grid-
   X[n])/h)/(N*h)
   kerest1 <- rep(1,N) %*%  ker1
   kerest1p <- discrete_deriv(grid,kerest1)
   kerest1pp <- discrete_deriv(grid,kerest1p)
   F <- discrete_int(grid,kerest1pp^2)
   return(h - nu2^.2 * (mu2^2 * F[I] * N)^-0.2)
}
opt2 <-
uniroot(f=plug_in,interval=c(0.1*hROT,10*hROT),N,I,gr
id,X,mu,nu,kerfunc)
hPI <- opt2$root
```

Computational Tricks for KDEs

- Bootstrap standard errors:

```
R <- 100 # number of bootstrap replications
kerestCurr <- matrix(rep(0,R*I),R)
for(r in 1:R)
{
   XCurr = sample(X,replace=T)
   kerCurr=matrix(rep(0,N*I),N)
   for(n in 1:N) kerCurr[n,1:I]=kerfunc((grid-
   XCurr[n])/h)/(N*h)
   kerestCurr[r,1:I] <- rep(1,N) %*%  kerCurr
}
```

Computational Tricks for KDEs

- Bootstrap standard errors (con't):

```
lower95b <- rep(I,0)
upper95b <- rep(I,0)
for(i in 1:I)
{
   lower95b[i]=quantile(kerestCurr[1:R,i],probs=.025,
   type=4)
   upper95b[i]=quantile(kerestCurr[1:R,i],probs=.975,
   type=4)
}
```

Computational Tricks for KDEs

- Naïve computational cost – O(NI)
- Binning – O(I^2)
  - For large data sets, bin data using equally spaced grid $(\tilde{x}_1, ..., \tilde{x}_I)$
  - Basically, round X_n to the nearest grid point
  - Define $w_i = \frac{1}{N} \#\{n : n = \arg\min_i | X_n - \tilde{x}_i |\}$
  - Binned KDE is $\hat{f}(\tilde{x}_i; h) = \frac{1}{hN} \sum_{j=1}^{I} w_j K\left(\frac{\tilde{x}_i - \tilde{x}_j}{h}\right)$
- Binning w/ Fast Fourier Transform – O(I * log(I))

Take Away Points

- Purely nonparametric problems are difficult:
  - Curse of dimensionality
  - Best ways to avoid the curse of dimensionality ($N^{-2/(d+4)}$):
    - Focus on a finite dimensional parameter of interest ($N^{-1/2}$)
    - Focus on one-dimensional function of interest ($N^{-2/5}$)
  - Often, we are really interested in a single β, the maximum value, the average derivative, and integrals (expected values) of the distribution, etc.
  - Often, only a one-dimensional function is of interest
  - How would we report high-dimensional functions? (we would end up focusing on low dimensional problems anyway)

Take Away Points

- Every nonparametric problem is different:
  - We can derive large sample approximation, obtain formulas for optimal bandwidth choices, formulas for standard errors, obtain efficiency bounds, one problem at a time
  - Better solution is to focus on methods which most easily generalize
  - Unfortunately, often may have to code from scratch for your problem

Take Away Points

- When applying Kernel methods more generally
  - Avoid procedures that require analytical derivations, since they may not be available for your problem
  - Use normal reference rule for density (“lazy” rule of thumb)
  - Use bootstrap to construct CIs and test statistics
  - Avoid bootstrap for non-smooth statistics for which bootstrap may not be consistent

Take Away Points

- Some stuff to try at home:
  - Use code on website to replicate plots in lecture
  - Perform similar calculations for incumbent spending
  - Derive normal reference rule for multivariate KDE using optimal bandwidth formula given in lecture notes
  - Estimate the joint density of incumbent position and incumbent spending
- Next lecture:
  - Kernel regression and semiparametric estimation