

Lecture 11

Introduction to Nonparametric Regression: Density Estimation

1

Non Parametric Regression: Introduction

- The goal of a regression analysis is to produce a reasonable analysis to the unknown response function f , where for N data points (X_i, Y_i) , the relationship can be modeled as

$$y_i = m(x_i) + \varepsilon_i, \quad i = 1, \dots, N$$

- Note: $m(\cdot) = E[y|x]$ if $E[\varepsilon|x]=0$ –i.e., $\varepsilon \perp x$

- We have different ways to model the conditional expectation function (CEF), $m(\cdot)$:
 - Parametric approach
 - Nonparametric approach
 - Semi-parametric approach.

2

Non Parametric Regression: Introduction

- Parametric approach: $m(\cdot)$ is known and smooth. It is fully described by a finite set of parameters, to be estimated. Easy interpretation. For example, a linear model:

$$y_i = x_i' \beta + \varepsilon_i, \quad i = 1, \dots, N$$

- Nonparametric approach: $m(\cdot)$ is smooth, flexible, but unknown. Let the data determine the shape of $m(\cdot)$. Difficult interpretation.

$$y_i = m(x_i) + \varepsilon_i, \quad i = 1, \dots, N$$

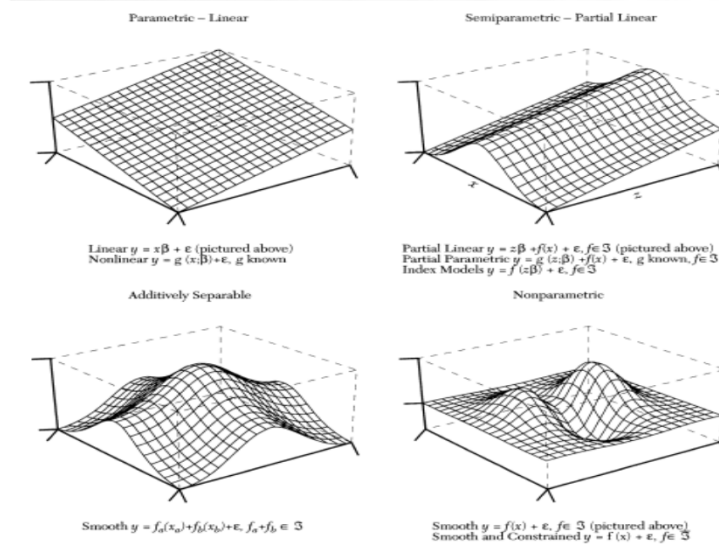
- Semi-parametric approach: $m(\cdot)$ have some parameters -to be estimated-, but some parts are determined by the data.

$$y_i = x_i' \beta + m_z(z_i) + \varepsilon_i, \quad i = 1, \dots, N$$

3

Non Parametric Regression: Introduction

Figure 2. Categorization of Regression Functions



\mathfrak{F} is a smooth family of functions. \mathfrak{F} is a smooth family with additional constraints such as monotonicity, concavity, symmetry or other constraints.

4

Non Parametric Regression: Introduction

- Parametric and non-parametric approaches use a weighted sum of the y 's to obtain the fitted values, \hat{y} . That is,

$$\hat{y}_i = \sum_i \omega_i y_i$$

- Instead of using equal weights as in OLS or weights proportional to the inverse of variance as often in GLS, a different rationale determines the choice of weights in nonparametric regression.
- In the single regressor case, the observations with the most information about $f(x_0)$ should be those at locations x_i closest to x_0 .
- Thus, a decreasing function of the distances of their locations x_i from x_0 determine the weights assigned to y_i 's.

5

Non Parametric Regression: Introduction

- A decreasing function of the distances of their locations x_i from x_0 determine the weights assigned to y_i 's.
- The points closest to x_0 receive more weight than those more remote from x_0 . Often, points remote from x_0 receive little or no weight.

6

Density Estimation: Univariate Case

- We have a large number of observations on a RV X . We would like to “draw” the pdf of X .

- Simplest method: Use a *histogram*. That is, divide the range of X into a small number of intervals (*bins*), h , and count the number of times X , n_p is observed in each interval:

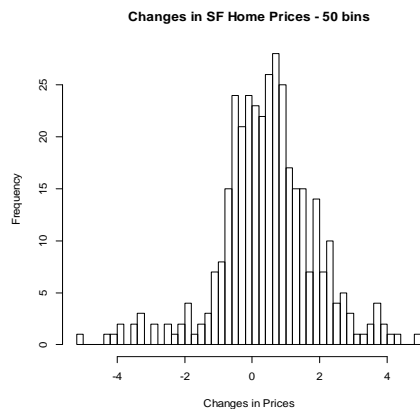
$$p_i = \frac{n_i(h)}{N}$$

- Q: How wide should the bins be? Too small (too many bins) distribution looks jerky, too large (few bins), shape is not easy to visualize.

- Two questions: - Do we want the same bin-width everywhere?
- Do we believe the density is zero for empty bins? 7

Density Estimation – Bins: Example

- We use two histograms to fit percentage changes in monthly San Francisco home prices (r_{sf} , with $N=359$), with two h (large h , 10 bins; small h , 50 bins). =>Smaller h , more resolution.



Density Estimation: Problems with Histograms

- The histogram is close to, but not truly density estimation. It does not estimate $f(x)$ at every x . Rather, it partitions the sample space into bins, and only approximate the density at the center of each bin.
- Two problems with histograms:
 - (1) For a given number of bins, moving their exact location (boundary points) can change the graph.
 - (2) The density function produced is a step function and the derivative either equals zero or is not defined (when at the cutoff point for two bins).
 - This is a problem if we are trying to maximize a likelihood function that is defined in terms of the densities of the distributions.

9

Density Estimation: Definition of Histogram

- First, define the density function for a variable x . For a particular value of x , call it x_0 , the density function is:

$$f(x_0) = \lim_{h \rightarrow 0} \frac{F(x_0 + h) - F(x_0 - h)}{2h} = \lim_{h \rightarrow 0} \frac{\text{Prob}[x_0 - h < x < x_0 + h]}{2h}$$

- For a sample of data on x of size N , a histogram with a column width of $2h$, centering the column around x_0 can be approximated by:

$$\hat{f}_{Hist}(x_0) = \frac{1}{N} \sum_{i=1}^N \frac{I[x_0 - h < x_i < x_0 + h]}{2h} = \frac{1}{Nh} \sum_{i=1}^N I\left(\frac{|x_i - x_0|}{h} < 1\right)$$

- This function equals the fraction of the sample that lies within h of x_0 , divided by the column width ($2h$). We call this the *naive estimator*.
- x_0 is any value of X , not necessarily equal to any x_i 's in the sample.

10

Density Estimation: Problems Revisited

- Dealing with the two problems:

(1) Arbitrary location of the bin cutoff points

Solution: Define a “moving” bin that is defined for every possible value of x . Then, count how many actual x_i 's are within $b/2$ of the hypothetical point, and “normalizes” this count by the number of total observations (N) and the “bandwidth,” b .

(2) Discontinuity in the function.

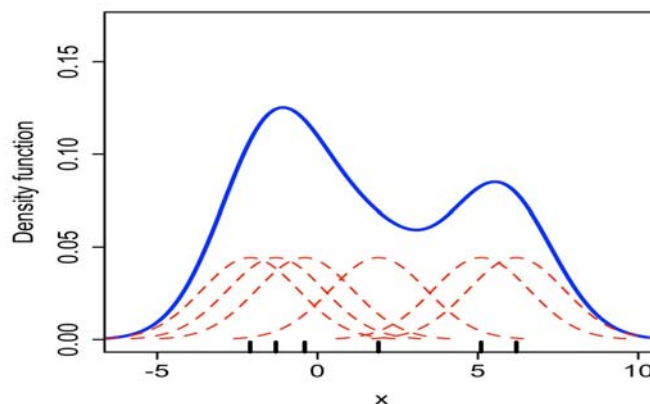
Solution: *Kernel density estimation (KDE)*. It avoids the discontinuities in the estimated (*empirical*) density function. In terms of histogram formula, the *kernel* is everything to the right of the summation sign. The general formula for the *kernel estimator (Parzen window)*:

$$\hat{f}_{Hist}(x_0) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x_i - x_0}{h}\right)$$

11

Kernel Density Estimation (KDE)

- The *kernel estimator* is given by: $\hat{f}_{Hist}(x_0) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x_i - x_0}{h}\right)$



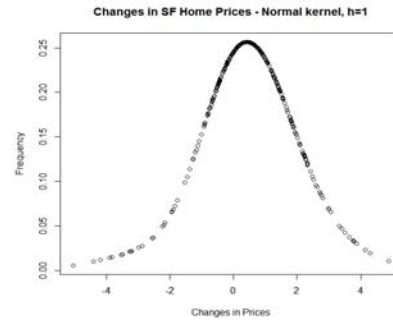
- That is, $\hat{f}_{Hist}(x)$ is a superposition of N density functions.

12

KDE: SF Prices Example With Normal Kernel

- Assume $K(\cdot) \sim N(0,1)$. Then, $\hat{f}_{Hist}(x_0) = \frac{1}{Nh} \sum_{i=1}^N dnorm\left(\frac{x_i - x_0}{h}\right)$

```
d_h <- matrix(0, N, 2)      # N=359
h <- 1                      # bandwidth
for (j in 1:N){
  d_h[j,1] <- r_sf[j]
  for (i in 1:N){
    d_h[j,2] <- d_h[j,2] + dnorm((r_sf[i]-d_h[j,1])/h)
  }
  d_h[j,2] <- d_h[j,2]/(N*h)
}
```



```
plot(d_h, xlab="Changes in Prices", ylab="Frequency", main = "Changes in SF Home Prices - Normal kernel, h=1")
```

- A lot of calculations: $N^2=128,881 \Rightarrow$ Not practical for large N .

13

KDE: Properties

- KDE: $\hat{f}_{Hist}(x_0) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x_i - x_0}{h}\right)$
- Q: Is $\hat{f}_{Hist}(x)$ a legitimate density function? It needs to satisfy:
 - nonnegative
 - integrate to one.
 - \Rightarrow Easy to do: Require the Kernel function, $K(\cdot)$ to satisfy:
 - $K(x) \geq 0$
 - $\int K(u) du = 1$

Define the function: $\delta_n(x) = \frac{1}{h} K\left(\frac{x}{h}\right)$

The $\hat{f}_{Hist}(x)$ can be written as $\hat{f}_{Hist}(x) = \frac{1}{N} \sum_{i=1}^N \delta(x_i - x_0)$

14

KDE: Properties

- Check the properties of $\hat{f}_{Hist}(x)$ and $\delta_n(x)$:

$$\int \delta_n(x - x_i) dx = \int \frac{1}{h} K\left(\frac{x - x_i}{h}\right) dx = \int K(u) du = 1$$

$$\int \hat{f}_{Hist}(x) = \int \frac{1}{N} \sum_i \delta_n(x - x_i) dx = \frac{1}{N} \sum_i \int \delta_n(x - x_i) dx = 1$$

- The kernel function can be generalized.

Note: Any density function satisfies our requirements. For example, $K(\cdot)$ can be a normal density.

15

KDE: Kernels

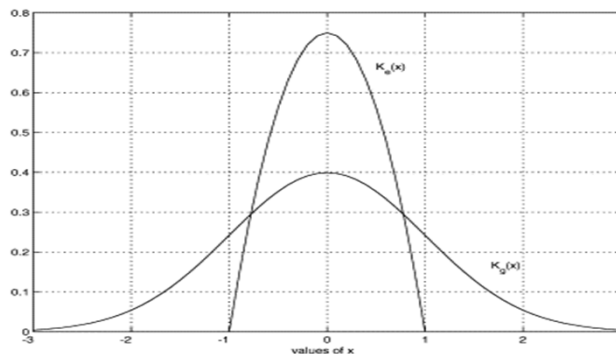
- The *kernel* function $K(\cdot)$ is a continuous and bounded (usually symmetric around zero) real function which integrates to 1.
- h is a smoothing parameter (*bandwidth*). $2h$ is called the *window width*.
- The order of a kernel, ν , is defined as the order of the first non-zero moment, κ_ν . For example, if $\kappa_1(K) = 0$ and $\kappa_2(K) > 0$ then K is a 2nd order kernel. The order of a symmetric kernel is always even.
- Symmetric non-negative kernels are second-order kernels. We will emphasize these kernels ($\nu=2$).
- Higher-order kernels are obtained by multiplying a second-order kernel by an $(2\nu-1)$ -th order polynomial in z^2 : See Hansen (2009).

KDE: Kernels

- Most common kernel functions:
 - Uniform kernel: $K(z) = 0.5$ for $|z| \leq 1$
 $= 0$ for $|z| > 1$
 - Epanechnikov kernel: $K(z) = 0.75(1-z^2)$ for $|z| \leq 1$
 $= 0$ for $|z| > 1$
 - Gaussian (normal) kernel: $K(z) = 1/\sqrt{2\pi}\exp(-z^2/2)$
 - Quartic (biweight) kernel: $K(z) = 15/16 (1-z^2)^2$ for $|z| \leq 1$
 $= 0$ for $|z| > 1$
 - Triweight kernel: $K(z) = 35/32 (1-z^2)^3$ for $|z| \leq 1$
 $= 0$ for $|z| > 1$
- Density graph: Plot $\hat{f}_{H_{ht}}(x)$ against x_0 and connect points.

KDE: Kernels - Examples

- Two kernels:
 - Epanechnikov kernel $K_e(z) = 0.75(1-z^2) I(|z| \leq 1)$
 - Gaussian kernel $K_g(z) = 1/\sqrt{2\pi}\exp(-z^2/2)$

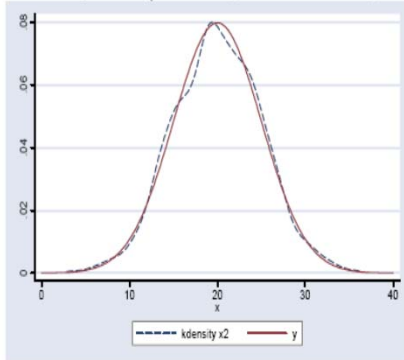


- A drawback of the Gaussian kernel is that its support is R ; in many situation, we want to restrict the support, like in the Epanechnikov kernel --at the cost of being not differentiable at ± 1 .

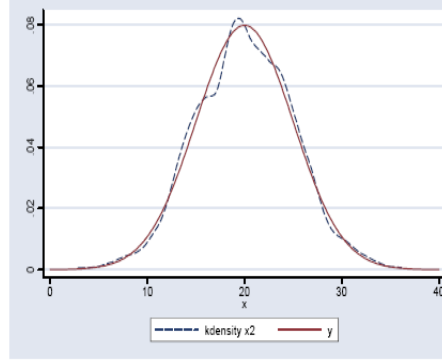
KDE: Kernels - Examples

- In practice, the choice of the kernel does not matter very much in terms of getting a good approximation to the true density function. Below, we show two estimations (gaussian and quartic) to simulated data.

Graph 5: Density of a Normally Distributed Variable, Mean=20, SD=5 (1000 obs., Gaussian kernel, bw=1)

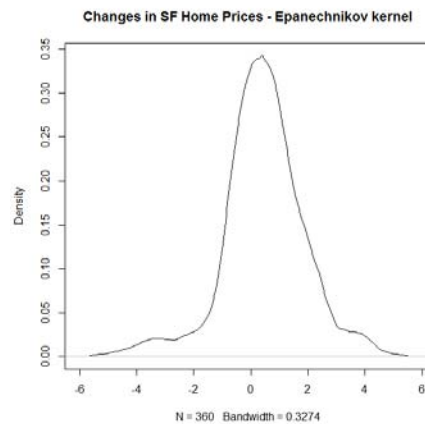
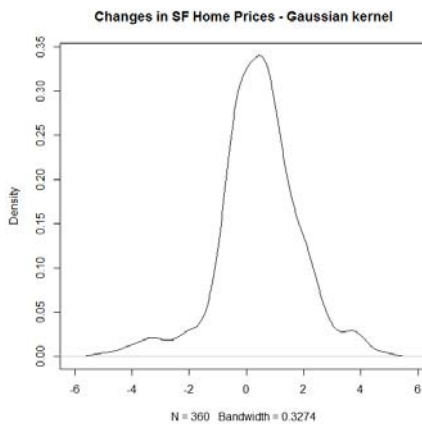


Graph 6: Density of a Normally Distributed Variable, Mean=20, SD=5 (1000 obs., Quartic kernel, bw=2)



KDE: Kernels - Examples

- We use a Gaussian and Epanechnikov kernels to fit percentage changes in monthly San Francisco home prices, with same h . Very similar results!



KDE: Statistical Inference

- *Consistency*

For an *i.i.d.* sample of the RV X , for any value x_0 and a fixed h , $\hat{f}(x_0)$ is a *biased* estimate of $f(x_0)$. Yet the bias goes to zero if $h \rightarrow 0$ as $N \rightarrow \infty$.

- The bias depends on h , the curvature of $f(\cdot)$, and $K(\cdot)$:

$$\text{bias}(\hat{f}(x_0)) = E[\hat{f}(x_0)] - f(x_0) = \frac{1}{2}h^2 f''(x_0) \int_{-\infty}^{\infty} z^2 K(z) dz$$

=> The “size” of this bias is $O(h^2)$.

- Assuming that $h \rightarrow 0$ as $N \rightarrow \infty$, the variance of $\hat{f}(x_0)$ is:

$$\text{Var}[\hat{f}(x_0)] = [1/(Nh)] f(x_0) \int (K(z))^2 dz + o(1/Nh)$$

=> The variance depends on the N , h , $f(\cdot)$ and $K(\cdot)$. It will go to 0 as $Nh \rightarrow \infty$, so h must converge to 0 at a *slower* rate than N goes to ∞ .

KDE: Statistical Inference

- The previous results were derived by approximating integrals by a Taylor expansion of $f(x+hu)$ in the argument $hu \rightarrow 0$.

- The kernel estimator $\hat{f}(x_0)$ is *pointwise consistent* at any point x_0 if both the variance and bias disappear as $N \rightarrow \infty$, which requires that $h \rightarrow 0$ and $Nh \rightarrow \infty$.

- The *uniform convergence* (stronger) property holds if $Nh/\ln(h) \rightarrow \infty$.

- See Cameron and Trivedi's (CT) textbook for formal details.

KDE: Statistical Inference

- *Asymptotic normality*

The kernel estimator is the sample average. A CLT can be applied.

Using previous results:

- Given the order of the variance, the rate of convergence is $\sqrt{N/b}$, not \sqrt{N} as in standard regression estimates.
- The estimator is biased, so we center $\hat{f}(x_0)$ around its expectation.

That is, by the CLT we get:

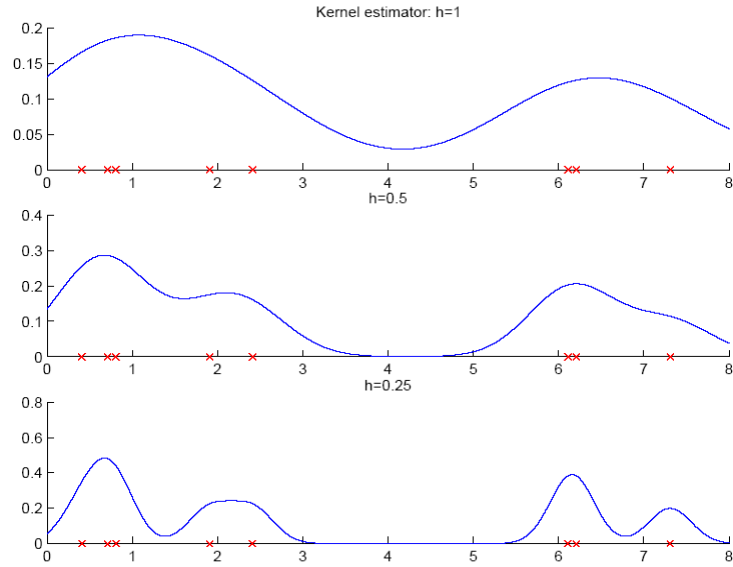
$$\sqrt{N/b} (\hat{f}(x_0) - E[\hat{f}(x_0)]) \rightarrow^d N(0, f(x_0) \int (K(z))^2 dz)$$

Note: Given the bias, $[\hat{f}(x_0) - E[\hat{f}(x_0)]]$, is also asymptotically normally distributed, but with a non-zero mean.

KDE: Bandwidth

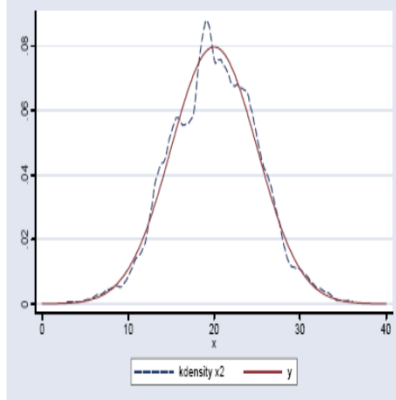
- As the previous formulas show, there is a genuine trade-off between avoiding bias and reducing the variance of the estimate at any given point x .
- In general, large h reduce the variance by smoothing over a large number of points, but this is likely to lead to bias because the points are “averaged” in a mechanical way that does not account for the particular shape of the distribution.
- In contrast, small h give higher variance but have less bias. In the limit, $h \rightarrow 0$, the kernel reproduced the data.
- We can play with different h 's, but we would like a data-driven bandwidth (“*automatic*”) selection process.

KDE: Bandwidth - Examples

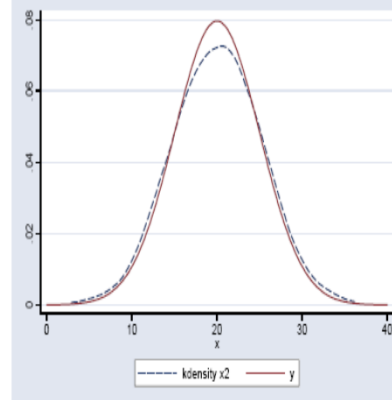


KDE: Bandwidth - Examples

Graph 8: Density of a Normally Distributed Variable, Mean=20, SD=5 (1000 obs., Epanechn. kernel, bw=0.5)

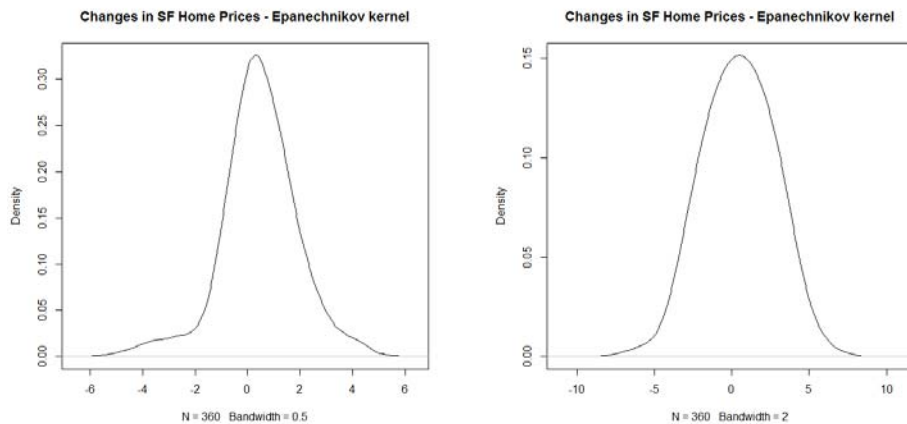


Graph 9: Density of a Normally Distributed Variable, Mean=20, SD=5 (1000 obs., Epanechn. kernel, bw=2)

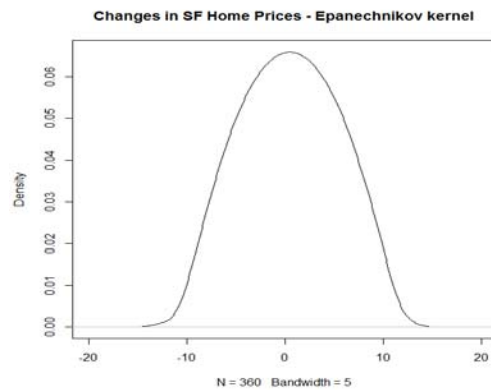


KDE: Bandwidth - Examples

- We use an Epanechnikov kernels to fit percentage changes in monthly San Francisco home prices, with two $h = .5, 2$ & 5 . Different results.



KDE: Bandwidth - Examples



- Q: How do we deal with the trade-off between bias and variance?

A: A “natural” approach is to minimize the MSE:

$$\text{MSE}(\hat{f}(x_0)) = \text{Var}[\hat{f}(x_0)] + [\text{bias}(\hat{f}(x_0))]^2$$

\Rightarrow *optimal bandwidth*

KDE: Bandwidth - Selection

- A “natural” approach is to minimize the MSE:

$$\text{MSE}(\hat{f}(x_0)) = \text{Var}[\hat{f}(x_0)] + [\text{bias}(\hat{f}(x_0))]^2$$

- As shown in previous formulas, the bias is $O(b^2)$ and the variance is $O(1/Nb)$. Intuitively, b should be chosen so that the $(\text{bias})^2$ and the variance are of the same order.

- The square of the bias is $O(b^4) \Rightarrow b^4 = 1/Nb,$
 $\Rightarrow b = (1/N)^{1/5}.$

That is, $b = O(N^{-0.2})$ and $\text{sqrt}(Nb) = O(N^{0.4})$.

- A more formal derivation is given on C&T.
- Note: Since the MSE is approximated using asymptotic expansion, it is called AMSE (asymptotic MSE).

KDE: Bandwidth and MISE

- Rosenblatt (1956) developed a global measure of accuracy for $\hat{f}(x_0)$: Minimizing the SSE at a large number of hypothetical points. As this number goes to infinity, this amounts to minimizing the *mean of the integrated squared error* (MISE). If the previous asymptotic approximations are used, the MISE becomes *AMISE*.

- That is, an optimal bandwidth minimizes

$$\text{MISE}(b) = E[\int (\hat{f}(x_0) - f(x_0))^2 dx_0] = E[\int \text{MSE}(\hat{f}(x_0)) dx_0]$$

- Differentiating $\text{AMISE}(b)$ w.r.t. b yields the optimal bandwidth:

$$b^* = \delta [\int f''(x_0)^2 dx_0]^{-0.2} N^{-0.2},$$

where δ depends on the kernel function used:

$$\delta = [\int (K(z))^2 dz]^{0.2} [\int z^2 K(z) dz]^{-0.4}.$$

- Note: $\int (K(z))^2 dz$ is called the *roughness* of $K(\cdot)$.

KDE: Optimal Bandwidth

- The optimal bandwidth, h^* :

$$h^* = \delta \left[\int (f''(x_0))^2 dx_0 \right]^{-0.2} N^{-0.2}.$$

- The optimal bandwidth decreases (very slowly) as N increases. Then, $h^* \rightarrow 0$ as $N \rightarrow \infty$ (as required for consistency).

- h^* depends on δ , which is a function of the kernel $K(\cdot)$. For example, if $K(\cdot)$ is Gaussian,

$$\begin{aligned} \delta &= \left[\int (K(z))^2 dz \right]^{0.2} \left[\int z^2 K(z) dz \right]^{-0.4} = [1/(2\sqrt{\pi})]^{0.2} [\sigma^2=1]^{-0.4} \\ &= [1/(2\sqrt{\pi})]^{0.2} (= .776388) \end{aligned}$$

- Values for δ are given in Table 9.1 in C&T.
- This result also shows that if the true density function has a lot of curvature ($f''(x)$ is large), the bandwidth should be smaller.

KDE: Optimality

- The optimal h^* is unknown –we do not know $f(x_0)$ or $f''(x_0)$. Approximations methods are required.

- In practice, a normal density is commonly used instead of $f(x_0)$. *Silverman's (1986) rule of thumb* assumes $f \sim N(\mu, \sigma^2)$, then:

$$h^* = (4\hat{\sigma}^5/3N)^{0.2} \approx 1.059 \hat{\sigma} N^{-0.2}$$

- As seen in the graphs, the choice of the kernel matters very little. More formally, $MISE(h^*)$ varies little across the different kernels.

- Technically speaking we can select the *best* kernel. The one that minimizes the AMISE. It is a calculus of variation problem

- The Epanechnikov (1969) kernel is “*optimal*,” but the advantage is small. It is often used to judge the efficiency of a kernel.

KDE: Confidence Intervals

- We can obtain confidence intervals for estimates of $f(x_0)$ for any point x_0 . Use the variance formula above to get the conventional C.I.:

$$\hat{f}(x_0) \in \hat{f}(x_0) - \text{bias}(x_0) \pm z_{\alpha/2} \sqrt{\frac{1}{N} \hat{f}(x_0) \int (K(z))^2 dz}$$

where $\text{bias}(x_0)$ is given above and we have assumed that $\hat{f}(x_0)$ is asymptotically normal.

- **Problem:** It can contain negative values.

Solution: Consider constructing the C.I. by inverting a test statistic.

$$C(x) = \{f: |\hat{f}(x_0) - \text{bias}(x_0)| / \sqrt{\frac{1}{N} \hat{f}(x_0) \int (K(z))^2 dz} \leq z_{\alpha/2}\}$$

This set must be found numerically.

- In practice, it is hard to calculate the bias, and, there may not be a reason to calculate the C.I. for $\hat{f}(x_0)$.

KDE in Practice: Bandwidth Selection

- As mentioned above to calculate h^* we need the unknown $f''(x_0)$. Approximations methods are required. In practice, a normal density is commonly used instead of $f(x_0)$.

$$1. \text{ If } X \sim \text{Normal, then we get } \left[\int (f''(x_0))^2 dx_0 \right]^{-0.2} = 1.3643$$

$$h^* = \delta \left[\int (f''(x_0))^2 dx_0 \right]^{-0.2} N^{-0.2} = 1.3643 \delta N^{-0.2} \hat{\sigma}, \quad \hat{\sigma} = \text{SD}(x).$$

$$\text{If in addition, } K(\cdot) \text{ is normal } (\delta = .776388) \quad \Rightarrow h^* = 1.059 N^{-0.2} \hat{\sigma}.$$

$$\text{If in addition, } K(\cdot) \text{ is the Epanechnikov} \quad \Rightarrow h^* = 2.34 N^{-0.2} \hat{\sigma}.$$

2. A refinement of the formula in (1), to account for outliers, is

$$h^* = 1.3643 \delta N^{-0.2} \min[\hat{\sigma}, \text{iqr}/1.349],$$

where “iqr” is the (sample) interquartile range.

- These rules for selecting h^* are generally called *rules of thumb*.

KDE in Practice: Boundary Effects

- So far, we have not paid much attention to the boundaries of the data, implicitly assuming that the density is supported on the entire R .
- In many situations, this is not the case. Then, the estimator can behave quite poorly due to what are called *boundary effects*.
- At the boundaries, $\hat{f}(x_0)$ usually underestimates $f(x_0)$. Suppose the data is positive, then $\hat{f}(x_0=0)$ penalizes $x_0=0$ for lack of data. At $x_0=0$, $\hat{f}(x_0)$ is inconsistent.
- Many proposed techniques to deal with boundary effects:
 - Reflection of data (“reflect” data at $x_0=0$, $-x_1$, $-x_2$, ..., $-x_N$).
 - Transformation of data (use a function $g(x)$; estimate $\hat{f}(x_0)$ instead).
 - Pseudo-Data Methods (“add” reasonable data, say by interpolation).
 - Boundary Kernel Methods (use a non-symmetric $K(\cdot)$ at $x_0=0$).

KDE in Practice: Computational Issues

- To get $\hat{f}_{Hist}(x)$ exactly, we must calculate for all x 's (x_1, \dots, x_N):

$$\hat{f}_{Hist}(x_j) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x_i - x_j}{h}\right), \quad j = 1, \dots, N$$

- Then, the number of evaluations of $K(\cdot)$ is proportional to N^2 (for a bounded Kernel, like the Epanechnikov, we have $h N^2$ evaluations). This increases the computation time if the N is large.

- For graphing the density, we do not need to evaluate $K(\cdot)$ at all x 's. Instead, $\hat{f}_{Hist}(x)$ can be computed at using some points, for example using an equidistant grid z_1, z_2, \dots, z_M :

$$z_k = x_{min} + (k/M)(x_{max} - x_{min}), \quad k = 1, 2, \dots, M \ll N$$

- Now, we only need $M \cdot h \cdot N$ $K(\cdot)$ evaluations. But, we can do better by “binning” the data –i.e., using a “*binned estimator*.”

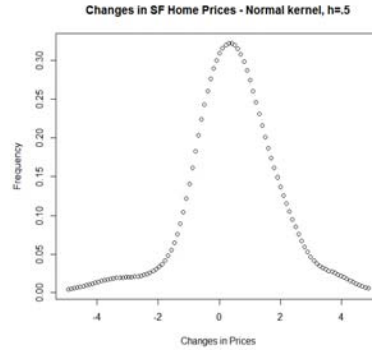
KDE in Practie: Computational Issues -

- For the SF changes in home prices data, we do KDE, with $M=100$.

```

M <- 100
d_h <- matrix(0, M, 2)
h <- .5
dist <- (max(r_sf) - min(r_sf))/1
for (j in 1:M){
  d_h[j,1] <- min(r_sf) + j/M*dist
  for (i in 1:N){
    d_h[j,2] <- d_h[j,2] + dnorm((r_sf[i] - d_h[j,1])/h)
  }
  d_h[j,2] <- d_h[j,2]/(N*h)
}
plot(d_h, xlab="Changes in Prices", ylab="Frequency", main = "Changes in SF Home Prices -
Normal kernel, h=.5")

```



- Still a lot of calculations: $M*N=35,900$ (better than $N^2 = 128881$) ³⁷

KDE in Practice: Computational Issues

- Binning or WARPing (Weighted Average of Rounded Points) “bins” the data in bins of length d starting at the origin x_j . Each x_i is replaced by the bincenter of the corresponding bin.
- A usual choice for d is to use $h/5$ or $(x_{min} - x_{max})/100$. In the latter case, the effective sample size (or number of grid points) R for the computation (the number of nonempty bins) can be at most 101.
- Now, $K(\cdot)$ needs to be evaluated only at $l d/h$, where $l=1, \dots, s$, where s is the number of bins which contains the support of $K(\cdot)$:

$$\hat{f}_{Hist}(w_j) = \frac{N_j}{Nh} \sum_{i=1}^R N_i K\left(\frac{(i-j)d_j}{h}\right), \quad j = 1, \dots, R$$

computed on the grid $w_j = (j+0.5)d$ (j integer) with N_i and N_j denoting the number of observations in the i -th and j -th bins, respectively.

KDE in Practice: Computational Issues

- The WARPing approximation requires (b^*R/d) evaluations of $K(\cdot)$ and $N + (b^*R/d)$ steps in total.
- Much faster than the exact computation, when N is large.
- The accuracy of binned estimators has been studied by Hall (1982), and Hall and Wand (1996), among others. The accuracy depends on the number of grid points R and can be made arbitrarily good by increasing R , at the cost of increasing the number of computations.
- Hall and Wand (1996) proposed that using R between 100 and 500 should give a reasonably good approximation.

KDE in Practice: Computational Issues

- The WARPing approximation requires (b^*R/d) evaluations of $K(\cdot)$ and $N + (b^*R/d)$ steps in total.
- Much faster than the exact computation, when N is large.
- The accuracy of binned estimators has been studied by Hall (1982), and Hall and Wand (1996), among others. The accuracy depends on the number of grid points R and can be made arbitrarily good by increasing R , at the cost of increasing the number of computations.
- Hall and Wand (1996) proposed that using R between 100 and 500 should give a reasonably good approximation.

KDE in Practice: Computational Issues

- For all but very small N , direct computation of $K(\cdot)$ is inefficient. By noticing that the KDE is based on a convolution of the data with the $K(\cdot)$, fast Fourier transformations (FFT) speed up computations:

$$\tilde{f}(x) = (2\pi)^{-1/2} \int \exp(ixt) f(t) dt \quad (\text{Fourier transform})$$

- Recall the convolution theorem:

If $g(x)$ and $k(x)$ are integrable functions with Fourier transforms $\tilde{g}(\xi)$ and $\tilde{k}(\xi)$ respectively, then the Fourier transform of the convolution is given by the product of the Fourier transforms $\tilde{g}(\xi)$ and $\tilde{k}(\xi)$. That is,

$$\text{if } f(x) = \int g(y) k(x-y) dy,$$

$$\text{then, } \tilde{f}(\xi) = \tilde{g}(\xi) \tilde{k}(\xi). \quad \Rightarrow \text{invert } \tilde{f}(\xi) \text{ to get } f(x)!$$

- Another approach: Fast Gaussian transformations (FGT).

KDE in R

- Kernel density estimates are available in R via the *density* function:

```
d <- density(r_sf, kernel=c("epanechnikov"), bw = .5)
plot(d, main = "Changes in SF Home Prices - Epanechnikov kernel")
```

which reproduces a previous density plot for SF home price changes, using the Epanechnikov kernel, with $b=.5$ (or $bw = .5$).

- By default, *density* uses a Gaussian kernel, but a large variety of other kernels are available by specifying the kernel option, like above with `kernel=c("epanechnikov")`,
- By default, *density* selects the bandwidth based on Silverman's (1986) rule of thumb. Other inputs (and manual inputs) can be used.

Estimating the Derivative of a Density

- Sometimes we need to estimate $f'(x)$ or even $f''(x)$ –like in a C.I.
- One approach for estimating $f'(x)$ is straightforward:

$$f'(x) = [\hat{f}(x_0+\Delta) - \hat{f}(x_0-\Delta)]/2\Delta,$$

for some small Δ .

- Alternatively, differentiate the expression $\hat{f}(x_0) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x_i - x_0}{h}\right)$ with respect to x_0 :

$$\hat{f}'(x_0) = \frac{1}{Nh^2} \sum_{i=1}^N K'\left(\frac{x_i - x_0}{h}\right)$$

- We can extend this approach to get the r -th derivative:

$$\hat{f}^{(r)}(x_0) = \frac{1}{Nh^{r+1}} \sum_{i=1}^N K^{(r)}\left(\frac{x_i - x_0}{h}\right)$$

Estimating the Derivative of a Density

- Since the Gaussian kernel has derivatives of all orders this is a common choice for derivative estimation.
- The estimator of $f^{(r)}(x)$ is biased –same order as for estimation of $f(x)$. But the variance is of a much larger order.
- We can derive an optimal bandwidth, we can optimize $\text{MISE}(h)$, as before.
- In practice, for either method, you should use a larger bandwidth than you use for estimating $\hat{f}(x_0)$.
- We can also ask the question of which kernel function is optimal. Muller (1984) found that the Biweight class is the optimal for the first derivative and for a second derivative the Triweight class.

Density Estimation: Adaptive Kernels

- So far, h has been fixed. But, this may not be optimal: what works fine in areas of high density may not necessarily be appropriate in a low-density regions.
- A possibility is to vary h , to use adaptive bandwidth kernel estimators, in which the bandwidth changes as a function of x_0 .
- Idea: Where there is a lot of data, we use a small neighborhood around x_0 ; but in areas with few data points, we expand the neighborhood. That is, $h_i = h(x_i)$
- But, note that these estimators introduce added bias in regions with little data in order to reduce variance there. The bias-variance trade-off is still there.

Density Estimation: k -Nearest Neighbor

- In k -Nearest Neighbor (k -NN), instead of fixing bin width h and counting the number of instances, we fix the instances (neighbors) k and check bin width.
- The neighborhood is defined through those X -variables which are among the k -nearest neighbors of x .
- The observations ranked by the distances, or “nearest neighbors,” are $(x_{(1)}, \dots, x_{(N)})$: The k -th nearest neighbor (or k -NN of x is $x_{(k)}$):
- The k -NN estimator is given by:
$$\hat{f}_{Hist, k-NN}(x_0) = \frac{k}{2Nd_k(x_0)}$$

where $d_k(x)$ is the ordered distance to k -th closest instance to x . $d_k(x)$ is usually the Euclidian distance (others, OK: Minkowski, Manhattan).

Density Estimation: k -Nearest Neighbor

- The k -NN estimator is given by: $\hat{f}_{Hist, k-NN}(x_0) = \frac{k}{2Nd_k(x_0)}$

a function of $d_k(x)$. If we use the Euclidian distance, $d_k(x) = \|x - x_{(k)}\|$.

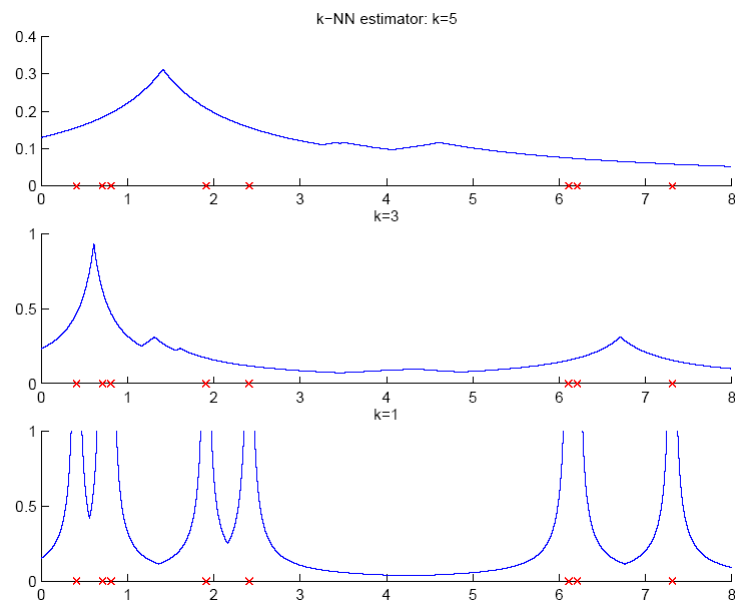
- Intuitively, we allow the bandwidth to vary depending on the density of the function. At areas of low density, we use a higher bandwidth to average over a larger number of (dispersed) points.

- While the traditional k -NN estimator uses a uniform kernel, smooth kernels can also be used. For example:

$$\hat{f}_{k-NN}(x_j) = \frac{1}{Nd_k(x_j)} \sum_{i=1}^N K\left(\frac{x_i - x_j}{d_k(x_j)}\right), \quad j = 1, \dots, N$$

In this case, the estimator is not just a function of $d_k(x)$.

Density Estimation: k -Nearest Neighbor



Density Estimation: k -Nearest Neighbor

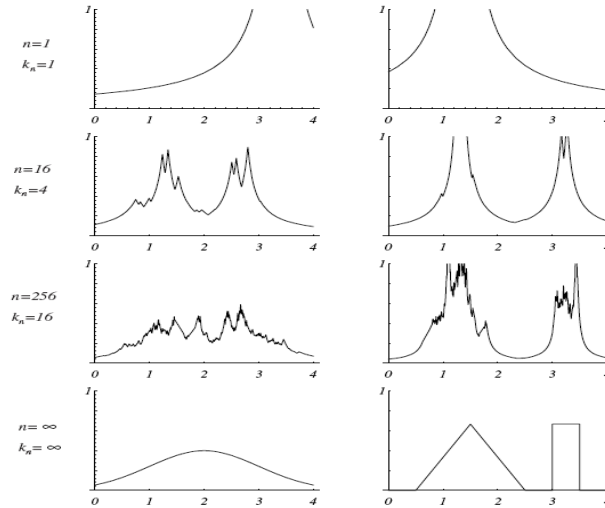


FIGURE 4.12. Several k -nearest-neighbor estimates of two unidimensional densities: a Gaussian and a bimodal distribution. Notice how the finite n estimates can be quite “spiky.” From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Density Estimation: Observations

- k -NN density estimation has a lot of discontinuities (very spiky, not differentiable). For small N , it is not even a density!
- Even for large regions with no observed data the estimated density is far from zero (tails are too heavy).
- Same trade-off as in selecting h : A smaller k allows only nearby data points to be considered (reduce bias); but a larger k allows for smoothness (reduce variance). Not easy to balance both issues.
- Given the variance-bias trade-off, selecting k is similar to selecting h (though k is an integer). There is no clear rule of thumb or optimality rule. Some proposals exist, but practitioners insist on “*know your data*” to select k .

Density Estimation: Kernel or k -NN?

- The asymptotic analysis of the k -NN estimator are complicated by the fact that $d_k(x)$ is random. The solution is to condition on $d_k(x)$, which is similar to treating it as fixed.
- Then, the conditional bias and variance are identical to those of the standard kernel estimator.
- For the unconditional bias, we need moments of $d_k(x)$, under Euclidian distance, given by the k -th order statistics. It turns out that the MSE behaves similarly to the kernel estimator's MSE.
- Q: Which one is better?
Not clear. In the tails, the Kernel estimator has smaller bias, but larger variance (the k -NN tends to be smoother in the tails).

Density Estimation: Multivariate Case

- Now suppose that \mathbf{X} is a d -vector and we want to estimate its density $f(\mathbf{x}) = f(x_1, \dots, x_d)$. Easy to extend the idea to this multivariate cases. Computations and interpretation get complicated once we move beyond 3 dimensions.

- Multivariate Kernel density estimator

$$\hat{f}(\mathbf{x}) = \frac{1}{Nh^d} \sum_{t=1}^N K\left(\frac{\mathbf{x} - \mathbf{x}^t}{h}\right)$$

- Multivariate Gaussian kernel

- spheric
$$K(\mathbf{u}) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \exp\left[-\frac{\|\mathbf{u}\|^2}{2}\right]$$

- ellipsoid
$$K(\mathbf{u}) = \frac{1}{(2\pi)^{d/2} |\mathbf{S}|^{1/2}} \exp\left[-\frac{1}{2} \mathbf{u}^T \mathbf{S}^{-1} \mathbf{u}\right]$$

Density Estimation: Multivariate Case

- We can estimate an optimal bandwidth, as before, by minimizing $MISE(b)$. The optimal bandwidth for the j -th variable is:

$$h_j^* = \delta \left[\int (f''(x_0))^2 dx_0 \right]^{-0.2} N^{-0.2} = C_v(K, d) N^{-1/(2v+d)},$$

- The optimal bandwidths will all be of order $N^{-1/(2v+d)}$ and the optimal MISE of order $N^{-2v/(2v+d)}$. These rates are slower than the univariate ($d = 1$) case.
- The fact that dimension has an adverse effect on convergence rates is called the *curse of dimensionality*.
- Rules of thumb can be derived for the constant $C_v(K, d)$. For example, for the Epanechnikov kernel, $C_{v=2}(K, d)$, is for $d=2$, 2.20, for $d=3$, 2.12; for $d=4$, 2.07.

Readings

- Cameron, A. and P. Trivedi (2003), **Microeconometrics: Methods and Applications**, Cambridge University Press.
- Hardle (1990), **Applied Nonparametric Regression**
- Yatchew, A (2003), **Semiparametric Regression for the Applied Econometrician**, Cambridge University Press.
- Silverman, B. W. (1986), **Density Estimation for Statistics and Data Analysis**.