# Nonparametrics and Local Methods

C.Conlon

March 2, 2020

Applied Econometrics

# Basic Non-parametrics

We all are familiar with models like this:

$$E[Y_i|X_i] = \beta_0 + \beta_1 X_1 + \dots$$

- Why is the conditional expectation function linear? Does it need to be?
- Plenty of real-world phenomenon are nonlinear.
- We can do things like this:

$$\log y_i = \beta_0 + \beta_1 X_1 + \beta_2 X^2 + \dots + \varepsilon_i$$

- But this still a linear model. Why?

# Basic Non-parametrics

We've seen some other examples:

$$
\begin{aligned}
Y_i &= 1 \quad \text{if } Y_i^* > 0 \\
Y_i &= 1 \quad \text{if } Y_i^* \leq 0 \\
Y_i^* &= \beta_0 + \beta_1 X_{1i} + \ldots + \varepsilon_i \quad \varepsilon_i \sim N(0, \sigma^2)
\end{aligned}
$$

- The probit estimator is a nonlinear function of its parameters.

# How about logit?

We've seen some other examples:

$$
\begin{aligned}
Y_i &= 1 \quad \text{if } Y_i^* > 0 \\
Y_i &= 1 \quad \text{if } Y_i^* \leq 0 \\
Y_i^* &= \beta_0 + \beta_1 X_{1i} + \ldots + \varepsilon_i \quad \varepsilon_i \sim TypeIEV/Gumbel
\end{aligned}
$$

This is not how we usually write things

# How about logit?

Instead we usually write:

$$E[Y_i|X_i] = Pr(Y_i = 1|X_i) = p(x_i)$$
$$Pr(Y_i = 1|X_i) = \frac{1}{1 + \exp^{\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots}}$$

Or with the log odds transformation:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots$$

Logit is linear again (in parameters). This is a generalized linear model.

# How else might we estimate $E[Y_i|X_i]$?

Obvious approach:

- With enough data: look at all values for $X_i = x$ and take the mean.
- Easy if $X$ is discrete or doesn't take on too many values (Gender, State/Country).
- Could work if $X$ is continuous but rounded (test scores, years of school, etc.).
- We could cut $x_i$ into distinct bins (like a histogram).

# A Fake Data Example

Following the THF textbook example, we can generate some fake data and let:

$$Y = \quad ORANGE \text{ if } Y* \quad > 0.5$$
$$Y = \quad\quad BLUE \text{ if } Y* \quad\quad \leq 0.5$$

- Easiest way to recover $Y*$ is by running OLS on the linear probability model.
- Draws from bivariate normal distribution with uncorrelated components but different means (2 overlapping types)
- Mixture of 10 low variance (nearly point mass) normal distributions where the individual means were drawn from another normal distribution. (10 nearly distinct types).

# Linear Probability Model


**FIGURE 2.1.** *A classification example in two dimensions. The classes are coded as a binary variable (*BLUE $= 0$, ORANGE $= 1$*), and then fit by linear regression. The line is the decision boundary defined by* $x^T\hat{\beta} = 0.5$*. The orange shaded region*

# Alternative

- Lots of potential alternatives to our decision rule.
- A simple idea is to hold a majority vote of neighboring points

$$Y^* = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

- To avoid including “yourself” in your neighborhood, we often estimate on one sample and validate on another
- How many parameters does this model have: None? One? $k$?
- Technically it has something like $N/k$.
- As $N \to \infty$ this means we have an infinite number of parameters! (This is a defining characteristic of non-parametrics).

# 15 Nearest Neighbor

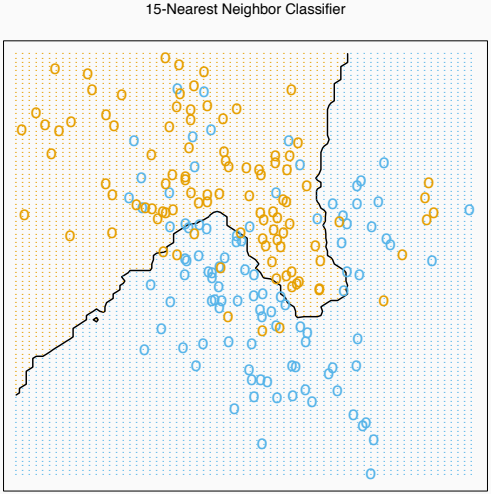


**FIGURE 2.2.** *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$) *and then fit by 15-nearest-neighbor averaging as in (2.8). The predicted class is hence*

# Extreme: 1 Nearest Neighbor

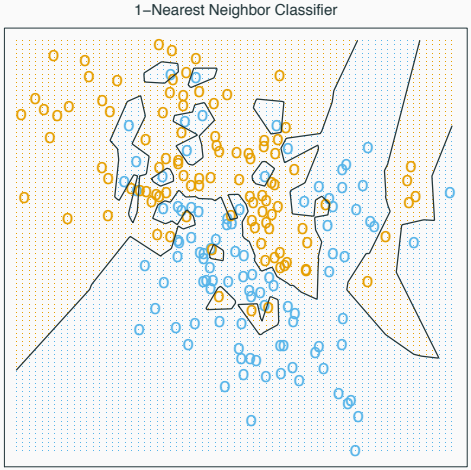


**FIGURE 2.3.** *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$)*, and then predicted by* 1*-nearest-neighbor classification.*

# Comparisons

- What would happen if $K \to N$?
- The k-NN model is locally constant.
- The k-NN approach tends to be really bumpy which can be undesirable.
- The OLS model is globally linear (is this always true?)

# What about?

- If we fixed the fact that there are discrete jumps in who is in the neighborhood by smoothly weighting observations and varying those weights instead (Kernels).
- Another drawback of $k - NN$ is that we consider distance in each $X$ dimension on the same scale, perhaps we could rescale the data to improve our "closeness" measure.
- Instead of fitting a constant locally, we fit a linear function locally (Lowess).
- Instead of using a global linear approximation in OLS use a more flexible nonlinear one.
- There is a bias/variance tradeoff. explain.

# Bias Variance Decomposition

We can decompose any estimator into two components

$$\underbrace{E[(y-\hat{f}(x))^2]}_{MSE} = \underbrace{\left(E[\hat{f}(x)-f(x)]\right)^2}_{Bias^2} + \underbrace{E\left[\left(\hat{f}(x)-E[\hat{f}(x)]\right)^2}_{Variance}$$

- In general we face a tradeoff between bias and variance.
- In k-NN as $k$ gets large we reduce the variance (each point has less influence) but we increase the bias since we start incorporating far away and potentially irrelevant information.
- In OLS we minimize the variance among unbiased estimators assuming that the true $f$ is linear.

# Bias Variance Decomposition

What minimizes MSE?

$$f(x_i) = E[Y_i|X_i]$$

- Seems simple enough (but we are back where we started).
- How do we compute the expectation ?
- k-NN tries to use local information to estimate conditional mean
- OLS uses entire dataset and adds structure $y = x\beta$ to the problem.

# Something Familiar

Let's start with something we all know, how to calculate:

$$E[Y_i|X_i] = \beta_0 + \beta_1 X_1 + \ldots + \varepsilon_i$$

- The Gauss-Markov theorem (remember that?) tells us that OLS is best among linear unbiased estimators.

# The Binary Case

Now let's think about a case where $Y \in \{0, 1\}$:

$$y_i = \mathbf{1}[F(X_i) - \varepsilon_i > 0]$$

- There are different choices of $F(X_i)$ and $f(\varepsilon)$
- Probit: $F(X_i) = \beta X_i$ where $\varepsilon_i \sim N(0, 1)$.
- Logit: same $F(\cdot)$ but different $f(\varepsilon)$ (Type I Extreme Value). $P(Y_i = 1|X_i) = 1/(1 + exp[-\beta X_i])$.
- These choices of $F$ are strong and somewhat arbitrary, we choose them out of convenience because both transformations are monotonic in $\beta X_i$, and continuously map $[-\infty, \infty] \rightarrow [0, 1]$

# The Binary Case

What is the least I can assume in the binary case and still learn something?

$$y_i = \mathbf{1}[F(X_i) - \varepsilon_i > 0]$$

- Suppose instead we observe $P(x) = Pr(y_i = 1|x_i)$ at many values of $x_i$. (Often refer to this as the CCP).
- Sometimes CCPs are all we care about. (if we are lucky).

# The Binary Case

Instead we can try and solve:

$$P(x) = \int \mathbf{1}(F(x) - \varepsilon > 0) dH(\varepsilon|x) = H(F(x)|x)$$

- Identification asks, what is the least we need to assume in order to recover $F, H$?
- Even with $\varepsilon \perp X$ we still have too many degrees of freedom.

# Are we stuck?

- We know that Logit and Probit are identified.
- Start with $\varepsilon \perp X$.
- $F(X_i) = G(X_i\beta)$ is a very powerful assumption often called (single) index model
- Can use Marginal Rate of Substitution (MRS) of $P(x)$ to identify $\beta_k/\beta_l$.
  - How do conditional probabilities respond to changes in $X^{(k)}$ versus $X^{(l)}$?
  - Suggests the average derivative estimator which doesn't require normality assumptions of probit
  - More robust, but potentially less efficient if true distribution of $\varepsilon \sim N(0,1)$.
- Can we extend the general intuition to more cases?

# A Fake Data Example

Following the THF textbook example, we can generate some fake data and let:

$$Y = \quad ORANGE \text{ if } Y* \quad > 0.5$$
$$Y = \quad\quad BLUE \text{ if } Y* \quad\quad \leq 0.5$$

- Easiest way to recover $Y*$ is by running OLS on the linear probability model.
- Draws from bivariate normal distribution with uncorrelated components but different means (2 overlapping types)
- Mixture of 10 low variance (nearly point mass) normal distributions where the individual means were drawn from another normal distribution. (10 nearly distinct types).

# Linear Probability Model


**FIGURE 2.1.** *A classification example in two dimensions. The classes are coded as a binary variable (*BLUE $= 0$, ORANGE $= 1$*), and then fit by linear regression. The line is the decision boundary defined by* $x^T\hat{\beta} = 0.5$*. The orange shaded region*

# Alternative

- Lots of potential alternatives to our decision rule.
- A simple idea is to hold a majority vote of neighboring points

$$Y^* = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

- To avoid including “yourself” in your neighborhood, we often estimate on one sample and validate on another
- How many parameters does this model have: None? One? $k$?
- Technically it has something like $N/k$.
- As $N \to \infty$ this means we have an infinite number of parameters! (This is a defining characteristic of non-parametrics).

# 15 Nearest Neighbor

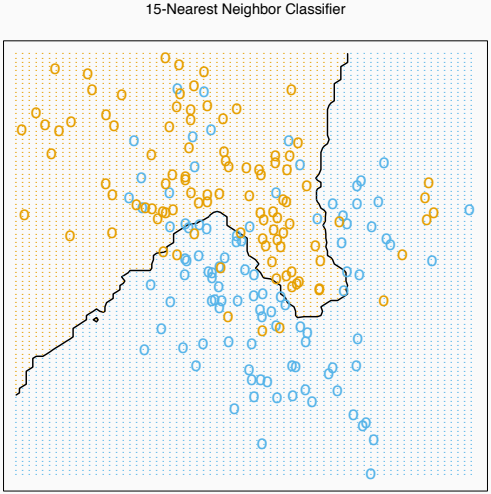


**FIGURE 2.2.** *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$) *and then fit by 15-nearest-neighbor averaging as in (2.8). The predicted class is hence*

# Extreme: 1 Nearest Neighbor

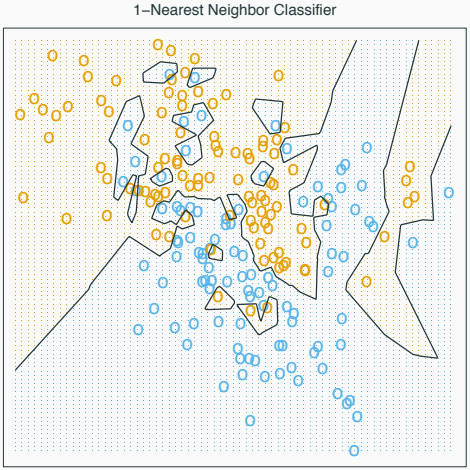


**FIGURE 2.3.** *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$)*, and then predicted by* 1*-nearest-neighbor classification.*

# Comparisons

- What would happen if $K \to N$?
- The k-NN model is locally constant.
- The k-NN approach tends to be really bumpy which can be undesirable.
- The OLS model is globally linear (is this always true?)

# What about?

- If we fixed the fact that there are discrete jumps in who is in the neighborhood by smoothly weighting observations and varying those weights instead (Kernels).
- Another drawback of $k - NN$ is that we consider distance in each $X$ dimension on the same scale, perhaps we could rescale the data to improve our "closeness" measure.
- Instead of fitting a constant locally, we fit a linear function locally (Lowess).
- Instead of using a global linear approximation in OLS use a more flexible nonlinear one.
- There is a bias/variance tradeoff. explain.

# Bias Variance Decomposition

We can decompose any estimator into two components

$$\underbrace{E[(y-\hat{f}(x))^2]}_{MSE} = \underbrace{\left(E[\hat{f}(x)-f(x)]\right)^2}_{Bias^2} + \underbrace{E\left[\left(\hat{f}(x)-E[\hat{f}(x)]\right)^2}_{Variance}$$

- In general we face a tradeoff between bias and variance.
- In k-NN as $k$ gets large we reduce the variance (each point has less influence) but we increase the bias since we start incorporating far away and potentially irrelevant information.
- In OLS we minimize the variance among unbiased estimators assuming that the true $f$ is linear.

# Bias Variance Decomposition

What minimizes MSE?

$$f(x_i) = E[Y_i|X_i]$$

- Seems simple enough (but we are back where we started).
- How do we compute the expectation ?
- k-NN tries to use local information to estimate conditional mean
- OLS uses entire dataset and adds structure $y = x\beta$ to the problem.

# Big Data

- It used to be that if you had $N = 50$ observations then you had a lot of data.
- Those were the days of finite-sample adjusted t-statistics.
- Now we frequently have 1 million observations or more, why can't we use k-NN type methods everywhere?

# Curse of Dimensionality

Take a unit hypercube in dimension $p$ and we put another hypercube within it that captures a fraction of the observations $r$ within the cube

- Since it corresponds to a fraction of the unit volume, $r$ each edge will be $e_p(r) = r^{1/p}$.
- $e_{10}(0.01) = 0.63$ and $e_{10}(0.1) = 0.80$, so we need almost 80% of the data to cover 10% of the sample!
- If we choose a smaller $r$ (include less in our average) we increase variance quite a bit without really reducing the required interval length substantially.

# Curse of Dimensionality

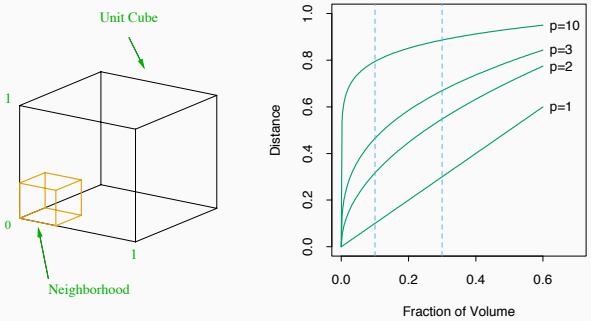


**FIGURE 2.6.** *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover* 80% *of the range of each coordinate to capture* 10% *of the data.*

# Curse of Dimensionality

Don't worry, it only gets worse:

$$d(p, N) = \left(1 - \left(\frac{1}{2}\right)^{1/N}\right)^{1/p}$$

- $d(p, N)$ is the distance from the origin to the closest point.
- $N = 500$ and $p = 10$ means $d = 0.52$ or that the closest point is closer to the boundary than the origin!
- Why is this a problem?
- In some dimension nearly every point is the closest point to the boundary – when we average over nearest neighbors we are extrapolating not interpolating.

# Density/Distribution Estimation

One of the more successful and popular uses of nonparametric methods is estimating the density or distribution function $f(x)$ or $F(x)$.

- Estimating the CDF is easy and something you have already done
- Q-Q plots, etc.

$$\hat{F}_{ECDF}(x_0) = \frac{1}{N}\sum_{i=1}^{N}(x_i \leq x_0)$$

- Differentiating to get density is unhelpful : $F'_{ECDF}(x) = 0$ in most places.

# Density/Distribution Estimation

One of the more successful and popular uses of nonparametric methods is estimating the density or distribution function $f(x)$ or $F(x)$.

- Think about the histogram (definition of derivative):

$$\hat{f}_{HIST}(x_0) = \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbf{1}(x_0 - h < x_i < x_0 + h)}{2h}$$

# Density/Distribution Estimation

- Divide the dataset into bins, count up fraction of observations in each bins
- Similar to k-NN except instead of windows that vary with $x_i$ we have fixed width bins
- Larger bin width $\rightarrow$ More Bias, Less Variance.
- Histogram will never be smooth! (Just like k-NN).

$$\hat{f}_{HIST}(x_0) = \frac{1}{Nh} \sum_{i=1}^{N} \frac{1}{2} \cdot \mathbf{1}\left(\left|\frac{x_i - x_0}{h}\right| < 1\right)$$

# Smooth Kernels

We can take our histogram and smooth it out:

$$\hat{f}(x_0) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right) \frac{1}{n} \sum_{i=1}^{n} K_h(y - y_i).$$

We call $K(\cdot)$ a Kernel function and $h$ the bandwidth. We usually assume

(i) $K(z)$ is symmetric about $0$ and continuous.

(ii) $\int K(z)dz = 1$, $\int zK(z)dz = 0$, $\int |K(z)|dz < \infty$.

(iii) Either (a) $K(z) = 0$ if $|z| \geq z_0$ for some $z_0$ or
(b) $|z|K(z) \to 0$ as $|z| \to \infty$.

(iv) $\int z^K(z)dz = \kappa$ where $\kappa$ is a constant.

# Kernel Smoothers

If $K$ is $C^k$, then so is $\hat{f}_n$, so we can plot it nicely.

Usually we choose a smooth, symmetric $K$:

- $K = \phi$, density of $N(0, 1)$ (or some other symmetric density);
- $K$ with compact support: Epanechnikov (mildly) optimal

$$K(x) = \frac{3}{4} \max(1 - x^2, 0).$$

A common nonsmooth choice: $K(x) = (|x| < 1/2)$ gives the *histogram* estimate.

# Kernel Comparison

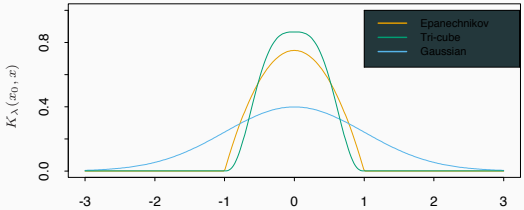


**FIGURE 6.2.** *A comparison of three popular kernels for local smoothing. Each has been calibrated to integrate to* 1*. The tri-cube kernel is compact and has two continuous derivatives at the boundary of its support, while the Epanechnikov kernel has none. The Gaussian kernel is continuously differentiable, but has infinite support.*

# How to Choose $h$

- We want both bias and variance to be as small as possible, as usual.
- In parametric estimation, it is not a problem: they both go to zero as sample size increases.

**Problem with nonparametrics:**

$$E\hat{f}_n(y) = \int K((y-t)/h)f(t)dt/h = \int K(-u)f(y+uh)du = f(y) + O(h^2)$$

$\rightarrow$ bias can be made tiny by having a very concentrated kernel ($h \simeq 0$); but

$$V\hat{f}_n(y) = \frac{1}{nh^2}VK((y-Y)/h) = O\left(\frac{1}{nh}\right)$$

$\rightarrow$ a small $h$ gives a high variance!

Reducing $h$ reduces bias, but increases variance; how are we to trade off?

# The AMISE

- Asymptotic Mean Integrated Square Error =asymptotic approximation of a quadratic loss function

$$E\left(\hat{f}_n(y) - f(y)\right)^2 dy$$

- Simple approximate expression (symmetric kernels of order 2):

$$\text{(bias)}^2 + \text{variance} = Ah^4 + B/nh$$

- **Why?** Bias in $y$ is

$$\int K(-u)\left(f(y+uh) - f(y)\right) du \simeq h^2 \frac{f''(y)}{2} \int K(u)u^2 du.$$

*Intuition:* if $f$ is close to linear around $y$, then averaging does not hurt us: $f''(y) \simeq 0$ and the bias is small. The bias is larger (and negative) at the mode of $f$.

# The Variance

$$V\hat{f}_n(y) = \frac{1}{nh^2} VK((y - Y)/h)$$

The important term in

$$VK((y - Y)/h)$$

is

$$h \int K(u)^2 f(y + uh) du \simeq hf(y) \int K(u)^2 du.$$

And we end up with

$$V\hat{f}_n(y) \simeq \frac{f(y)}{nh} \int K(u)^2 du.$$

*Intuition:* we are really taking an average over $nhf(y)$ points. In low-density region, this induces a high *relative* imprecision:

# Optimal bandwidth

- The AMISE is

$$Ah^4 + B/nh$$

  with $A = \int (f''(y))^2 \left( \int u^2 K \right)^2 /4$ and $B = f(y) \int K^2$

- AMISE is smallest in $h_n^* = \left( \frac{B}{4An} \right)^{1/5}$. Then,
  - bias and standard error are *both* in $n^{-2/5}$
  - and the AMISE is $n^{-4/5}$—**not** $1/n$ as it is in parametric models.
- But: $A$ and $B$ both depend on $K$ (known) and $f(y)$ (unknown), and especially "wiggliness" $\int (f'')^2$ (unknown, not easily estimated). Where do we go from here?

# Silverman's Rule of Thumb

- If $f$ is normal with variance $\sigma^2$ (may not be a very appropriate benchmark!), the optimal bandwidth is
$$h_n^* = 1.06\sigma n^{-1/5}$$
- Just do it with $\sigma = s$ empirical dispersion of the $y_i$'s , or something more robust/slightly less smooth:
$$h_n^* = 0.9 * \min(s, IQ/1.34) * n^{-1/5},$$ IQ=interquartile distance
- Investigate changing it by a reasonable multiple.

# Cross-validation

- General concept in the whole of nonparametrics: choose $h$ to minimize a criterion $CV(h)$ that approximates

$$AMISE(h) = \int E(\hat{f}_n(x) - f(x))^2 dx.$$

- Usually programmed in metrics software. *If you can do it, do it on a subsample, and rescale.*
- Two problems:
  - it is costly; often it involves computing "leave-one-out" estimators

$$\hat{f}_{(-i)}(x_i) = \frac{1}{nh} \sum_{j \neq i} K\left(\frac{x_i - x_j}{h}\right),$$

    for *every* observation $i$.
  - the resulting $h$ converges super-slowly ($n^{-1/10}$!) to the optimal one.

# Alternatives to Cross-Validation

- LOOCV
- k-fold CV
- Sample Splitting
  - Training Set
  - Test Set
  - Validation Set

# Local Bandwidths

If you only care about $f(y)$ at some given point, then

$$A = f''(y)^2 \left( \int u^2 K \right)^2 /4 \text{ and } B = f(y) \int K^2.$$

So in a low-density region, worry about variance and take $h$ larger. In a curvy region, worry about bias and take $h$ small.

# Higher-Order Kernels

- $K$ of order $r$ iff $\int x^j K(x)dx = 0$ for $j < r$ and $\int x^r K(x)dx \neq 0$. Try $r > 2$?
- The beauty of it: bias in $h^r$ if $f$ is at least $C^r \ldots$ so AMISE can be reduced to $n^{-r/(2r+1)}$, almost $\sqrt{n}$-consistent if $r$ is large.
- But gives wiggly (and sometimes negative) estimates $\rightarrow$ leave them to theorists.

# Back to the CDF

Since now we have estimated the density with

$$\hat{f}_n(y) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{y - y_i}{h}\right),$$

a natural idea is to integrate; let $\mathcal{K}(y) = \int_{-\infty}^{y} K(t)dt$, try

$$\hat{F}_n(y) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{K}\left(\frac{y - y_i}{h}\right)$$

as a reasonable estimator of the cdf in $y$. Very reasonable indeed:

- when $n \longrightarrow \infty$ and $h$ goes to zero (at rate $n^{-1/3} \ldots$) it is consistent at rate $\sqrt{n}$
- it is nicely smooth and accords well with the density estimator
- $\ldots$ it is a much better choice than the empirical cdf.

## What if $y$ is of dimension $p_y > 1$?

"Easy": use $p_y$-dimensional $K$ (often a $p_y$-product of 1-dim kernels) and bandwidth $h$, and do

$$\hat{f}_n(y) = \frac{1}{nh_y^p} \sum_{i=1}^{n} K\left(\frac{y - y_i}{h}\right).$$

- **1st minor pitfall:** the various dimensions may have very different variances, so use $(h_1, \ldots, h_{p_y})$.
- **2nd minor pitfall:** they may be strongly correlated; then sphericize first.
- **Major problem:** next slide...

# The Curse of Dimensionality

- Computational cost increases exponentially.
- *Much worse:* to achieve precision $\epsilon$ in dimension $p_y$, the number of observations you need increases as

$$n \simeq \epsilon^{-(2+p_y/2)}.$$

- The *empty space* phenomenon: if $(y_1, \ldots, y_{p_y})$ all are iid uniform on $[-1, 1]$, then only $n/(10^{p_y})$ observations on average have all components in $[-0.1, 0.1]$. Bias still in $h^2$, but variance in $1/nh^{p_y}$ now.

# Silverman's Table

Silverman (1986 book) provides a table illustrating the difficulty of kernel estimation in high dimensions. To estimate the density at 0 of a $N(0,1)$ with a given accuracy, he reports:

| Dimensionality Required | Sample Size |
|---|---|
| 1 | 4 |
| 2 | 19 |
| 5 | 786 |
| 7 | 10,700 |
| 10 | 842,000 |

**Not to be taken lightly...** in any case convergence with the optimal bandwidth is in $n^{-2/(4+p_y)}$ now—and Silverman's rule of thumb for choosing $h_n^*$ must be adapted too.

# Usually we care about conditional densities

That is: we have covariates $x$, we want the density $f(y|x)$. Again, "easy":

1. get a kernel estimator of the joint density $f(y, x)$;
2. and one of of the marginal density $f(x)$;
3. then define

$$\hat{f}_n(y|x) = \frac{\hat{f}_n(y,x)}{\hat{f}_n(x)} = \frac{\frac{1}{nh_y^{p_y}h_x^{p_x}}\sum_{i=1}^n K\left(\frac{y-y_i}{h_y}\right)K\left(\frac{x-x_i}{h_x}\right)}{\frac{1}{h_y^{p_y}}\sum_{i=1}^n K\left(\frac{x-x_i}{h_x}\right)}.$$

But the joint density is $(p_x + p_y)$ dimensional... and the curse strikes big time.

# What if my distribution is discrete-continuous?

Very often in microeconometrics some covariates only take discrete values (e.g. gender, race, income bracket...). Say the only discrete variable is gender, we care about the density of income of men.

- The kernel approach adapts directly: we separately estimate a density for men (on the corresponding subsample).
- *Better:* mix the two subsamples! Add women, but **with a small weight** $w$.
- Intuition: by doing so we increase the bias (the density for women is probably different than for men) $\rightarrow$ bad, in $w^2$ but we reduce the variance, by $O(w)$; and this dominates for small $w$. (cf Li-Racine).

# The Seminonparametric Approach

- If we are “pretty sure” that $f$ is almost $f_{m,\sigma}$ for some family of densities indexed by $(m, \sigma)$, then we can choose a family of positive functions of increasing complexity $P^1_\theta, P^2_\theta, \ldots$
- Choose some $M$ that goes to infinity as $n$ does (more slowly), and maximize over $(m, \sigma, \theta)$ the loglikelihood

$$\sum_{i=1}^{n} \log f_{m,\sigma}(y_i) P^M_\theta(y_i).$$

It works. . . but it is hard to constrain it to be a density for large $M$.

# Mixtures of Normals

A special case of seminonparametrics, and usually a very good approach: Let $y|x$ be drawn from

$$
\begin{aligned}
&N(m_1(x,\theta), \sigma_1^2(x,\theta)) \text{ with probability } q_1(x,\theta);\\
&\ldots\\
&N(m_K(x,\theta), \sigma_K^2(x,\theta)) \text{ with probability } q_K(x,\theta).
\end{aligned}
$$

where you choose some parameterizations, and the $q_k$'s are positive and sum to 1.

Can be estimated by maximum-likelihood:

$$
\max_{\theta} \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \frac{q_k(x_i,\theta)}{\sigma_k(x,\theta)} \phi\left(\frac{y_i - m_k(x_i,\theta)}{\sigma_k(x_i,\theta)}\right)\right).
$$

Usually works very well with $K \leq 3$ (perhaps after transforming $y$ to $\log y$, e.g).

# Nonparametric Regression

Data $(y_i, x_i)_{i=1}^n$ now, we are after $E(g(y,x)|x) = m(x)$ for some function $g$.

- Best-fit approach, quite unbiased:
  - if $x = x_i$ then $\hat{m}_n(x) = g(y_i, x_i)$; otherwise . . . whatever.

  But: very jagged estimate; variance independent of $n$, so not consistent.
- Better and most usual: Nadaraya-Watson, inspired from kernel idea:

$$\hat{m}_n(x) = \frac{\sum_{i=1}^n g(y_i, x_i) K\left(\frac{x-x_i}{h}\right)}{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)}.$$

  again, bias in $h^2$ and variance in $1/nh$ if $p_x = 1$.
  *Pitfall 1:* very unreliable where $f(x)$ is small.
  *Pitfall 2:* the formal for the optimal bandwidth $h$ is very ugly.

# Choosing $h$

- Plug-in estimates work badly.
- Fortunately, cross-validation amounts to

$$\min_h \sum_{i=1}^{n} \frac{(g(y_i, x_i) - \hat{m}_n(x_i; h))^2}{1 - k_i(h)}$$

  where $k_i(h) = K_h(0) / \sum_{j=1}^{n} K_h(x_i - x_j)$.
- So not that hard, and can be done on a subsample and rescaled.

# Local Linear Regression

- The Nadaraya-Watson estimator in $x$ can be obtained very simply by regressing $g(y_i, x_i)$ on 1, weighting each observation by $K((x - x_i)/h)$.
- We could also regress on 1 and $(x - x_i)$ (going to higher terms has problems) instead;
  Advantages:
  - the bias becomes 0 if the true $m(x)$ is linear.
  - the coefficient of $(x - x_i)$ estimates $m'(x)$.
  - behaves better in “almost empty” regions.

  Disadvantages: hardly any, just do it!

# Local Linear

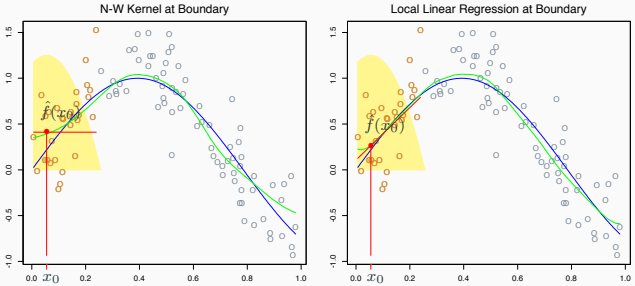


**FIGURE 6.3.** *The locally weighted average has bias problems at or near the boundaries of the domain. The true function is approximately linear here, but most of the observations in the neighborhood have a higher mean than the target point, so despite weighting, their mean will be biased upwards. By fitting a locally weighted linear regression (right panel), this bias is removed to first order.*

# Local Linear

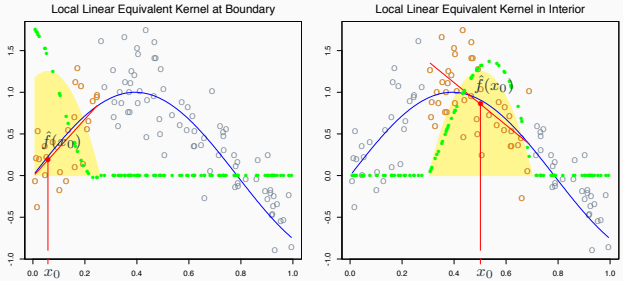


**FIGURE 6.4.** *The green points show the equivalent kernel $l_i(x_0)$ for local regression. These are the weights in $\hat{f}(x_0) = \sum_{i=1}^{N} l_i(x_0) y_i$, plotted against their corresponding $x_i$. For display purposes, these have been rescaled, since in fact they sum to 1. Since the yellow shaded region is the (rescaled) equivalent kernel for the Nadaraya–Watson local average, we see how local regression automatically modifies the weighting kernel to correct for biases due to asymmetry in the smoothing window.*

# Local Quadratic

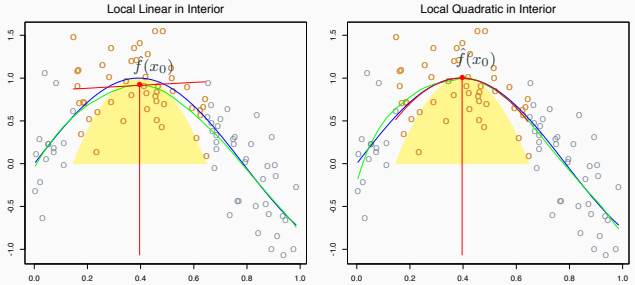


**FIGURE 6.5.** *Local linear fits exhibit bias in regions of curvature of the true function. Local quadratic fits tend to eliminate this bias.*

# Nonparametric Regression, summary, 1

Nadaraya–Watson for $E(y|x) = m(x)$

$$\hat{m}(x) = \frac{\sum_i y_i K_h(x - x_i)}{\sum_i K_h(x - x_i)}$$

- bias in $O(h^2)$, variance in $1/(nh^{p_x})$
- optimal $h$ in $n^{-1/(p+4)}$: then bias, standard error and RMSE all converge at rate $n^{-2/(p+4)}$
- to select $h$, no rule of thumb: cross-validate on a subsample and scale up.

# Nonparametric Regression, summary, 2

Nadaraya–Watson=**local constant regression**: to get $\hat{m}(x)$,

1. regress $y_i$ on $1$ with weight $K_h(x - x_i)$
2. take the estimated coeff as your $\hat{m}(x)$.

Better: **local linear regression**

1. regress $y_i$ on $1$ and $(x_i - x)$ with weight $K_h(x - x_i)$
2. take the estimated coeffs as your $\hat{m}(x)$ and $\hat{m}'(x)$.

To estimate the standard errors: bootstrap on an *undersmoothed* estimate (so that bias is negligible.)

# Seminonparametric (=Flexible) Regression

**Idea:** we add regressors when we have more data

→ **series or sieve estimators**: choose a basis of functions $P_k(x_i)$ ($x_i^k$, or orthogonal polynomials, or sines. . . )

→ run *linear regression* $y_i = \sum_{k=1}^{M} P_k(x_i)\theta_k + \epsilon_i$

a reasonable compromise (again, $M$ must go to infinity, more slowly than $n$).

Still curse of dimensionality, and nonparametric asymptotics.

# Splines: trading off fit and smoothness

Choose some $0 < \lambda < \infty$ and

$$\min_{m(.)} \sum_i (y_i - m(x_i))^2 + \lambda J(m),$$

Then we "obtain" the natural cubic spline with knots=$(x_1, \ldots, x_n)$:

- $m$ is a cubic polynomial between consecutive $x_i$'s
- it is linear out-of-sample
- it is $C^2$ everywhere.

"Consecutive" implies one-dimensional... harder to generalize to $p_x > 1$.

**Orthogonal polynomials:** check out Chebyshev, $1, x, 2x^2 - 1, 4x^3 - 3x \ldots$ (on $[-1, 1]$ here.)

# Review: What was the point?

- OLS is lowest variance among linear unbiased estimators.
- But there are nonlinear estimators and potentially biased estimators.
  - Everything faces a bias-variance tradeoff.
  - Nearly anything can be written as Kernel.