Kernel Smoothing Method

Yanrong Yang

RSFAS/CBE, Australian National University

16th August 2022

Contents of this week

Kernel Smoothing Method

- Estimation of Density Function
- Estimation of Regression Function
- Curse of Dimensionality on Multivariate Data
- Alternative Multivariate Models or Estimation Methods

Problems

Nonparametric Estimation

- Density Function. Let $X_1, X_2, \ldots, X_n$ be sample observations from the density function $p(x)$. How to estimate the density function $p(x)$?
- Regression Function. Consider a regression model

$$Y_i = f(X_i) + \varepsilon_i, \quad i = 1, 2, \ldots, n. \tag{1}$$

The aim is to estimate the unknown regression function $f(x)$, with the sample observations $(X_i, Y_i), i = 1, 2, \ldots, n$.

Evaluation of An Estimator

Take a density estimator $\widehat{p}(x)$ as an example.

Let $\widehat{p}$ be an estimate of p. As a loss function we will use

$$L(p, \widehat{p}) = \int (p(x) - \widehat{p}(x))^2 dx.$$

The risk is

$$R = \mathbb{E}\left(L(p, \widehat{p})\right) = \int \mathbb{E}(p(x) - \widehat{p}(x))^2 dx = \int (b^2(x) + v(x)) dx$$

where

$$b(x) = \mathbb{E}(\widehat{p}(x)) - p(x)$$

is the bias and

$$v(x) = \mathrm{Var}(\widehat{p}(x)).$$

Empirical Density Estimation

An intuitive and simple estimator for density function is histogram, i.e. empirical density function.

The simplest estimator is the histogram. Suppose, for simplicity, that $X_i \in [0,1]$. Divide $[0,1]$ into m bins

$$B_1 = [0, 1/m), B_2 = [1/m, 2/m), \ldots, B_m = ((m-1)/m, 1].$$

Let $h = 1/m$ be the width of the bins. Let $\theta_j = P(B_j)$. Let $x \in B_j$. Then

$$\theta_j = P(B_j) = \int_{B_j} p(u)du \approx p(x)h$$

and so $p(x) \approx \theta_j/h$. This suggests the following estimator:

$$\widehat{p}(x) = \frac{\widehat{\theta}_j}{h}$$

where

$$\widehat{\theta}_j = \frac{1}{n}\sum_i I(X_i \in B_j).$$

Evaluation of Empirical Density Estimation

$$\mathbb{E}[\widehat{\theta}_j] = \theta_j = \int_{B_j} p(u)du \approx \int_{B_j} [p(x) + (u-x)p'(x)]du = p(x)h + p'(x)\int_{B_j}(u-x)du$$

so that, for $x \in B_j$,

$$\mathbb{E}[\widehat{p}(x)] = \mathbb{E}\left[\frac{\widehat{\theta}_j}{h}\right] = p(x) + \frac{p'(x)}{h}\int_{B_j}(u-x)du$$

and the bias is

$$b = \frac{p'(x)}{h}\int_{B_j}(u-x)du.$$

So

$$|b| \leq \frac{|p'(x)|}{h}\int_{B_j}|u-x|du \approx \frac{|p'(x)|h^2}{h} = Ch$$

for some $C > 0$.

From now on we use C to represent various constants.

The variance is

$$\text{Var}\left(\widehat{p}(x)\right) = \text{Var}\left(\frac{\widehat{\theta}_j}{h}\right) = \frac{\theta_j(1-\theta_j)}{nh^2} \approx \frac{\theta_j}{nh^2} \approx \frac{Ch}{nh^2} = \frac{C}{nh}.$$

Optimal Bandwidth via MSE

So

$$b^2(x) + v(x) = Ch^2 + \frac{C}{nh}$$

and so the risk is $Ch^2 + \frac{C}{nh}$. This is minimized by choosing $h = (C/n)^{1/3}$. This results in

$$R = O\left(\frac{1}{n}\right)^{2/3}.$$

In practice, h is chosen by cross-validation.

Another Way to Empirical Density Estimation

Consider the empirical density estimator

$$\hat{f}(x) = \frac{\hat{F}(x+h) - \hat{F}(x-h)}{2h}$$

We can write this as

$$\begin{aligned} \frac{1}{2nh}\sum_{i=1}^{n} 1\left(x+h < X_i \leq x+h\right) &= \frac{1}{2nh}\sum_{i=1}^{n} 1\left(\frac{|X_i - x|}{h} \leq 1\right) \\ &= \frac{1}{nh}\sum_{i=1}^{n} k\left(\frac{X_i - x}{h}\right) \end{aligned}$$

where

$$k(u) = \begin{cases} \frac{1}{2}, & |u| \leq 1 \\ 0 & |u| > 1 \end{cases}$$

is the uniform density function on $[-1, 1]$.

Generalized Empirical Density Estimation

The estimator $\hat{f}(x)$ counts the percentage of observations which are clsoe to the point x. If many observations are near x, then $\hat{f}(x)$ is large. Conversely, if only a few X_i are near x, then $\hat{f}(x)$ is small. The **bandwidth** h controls the degree of smoothing.

$\hat{f}(x)$ is a special case of what is called a kernel estimator. The general case is

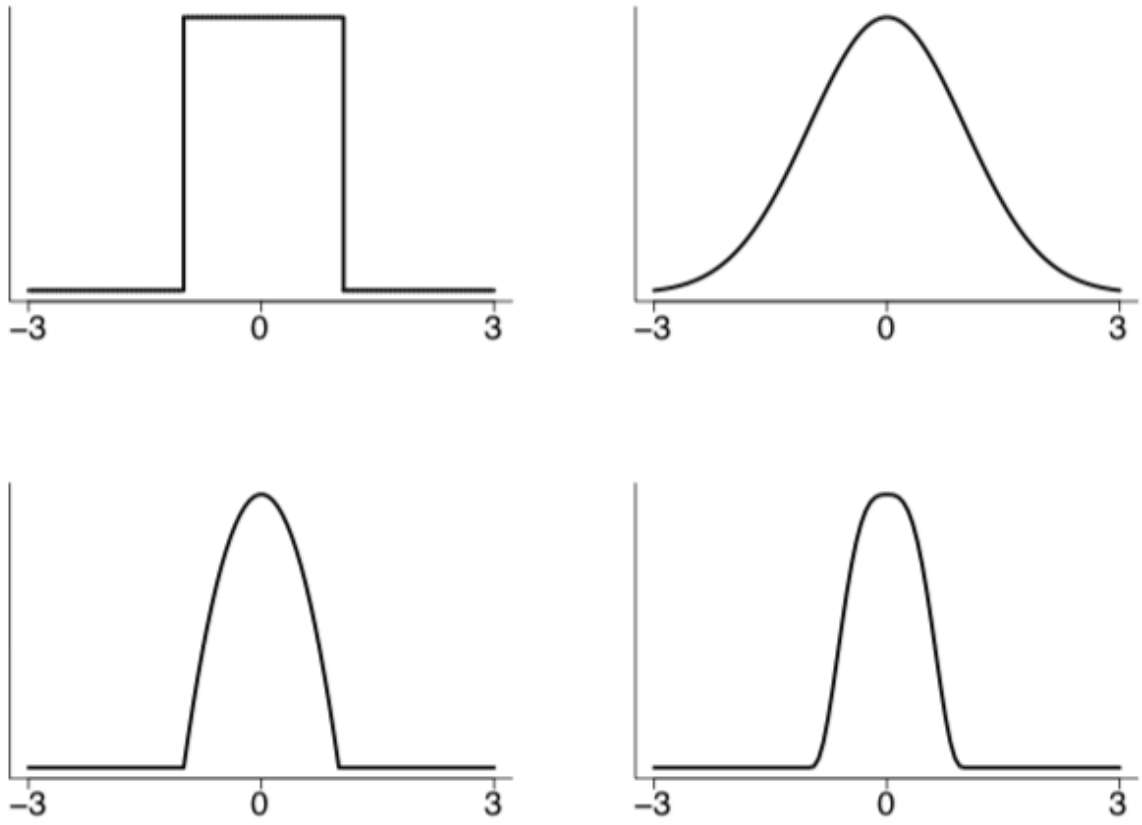
$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} k\left(\frac{X_i - x}{h}\right)$$

where $k(u)$ is a **kernel function**.

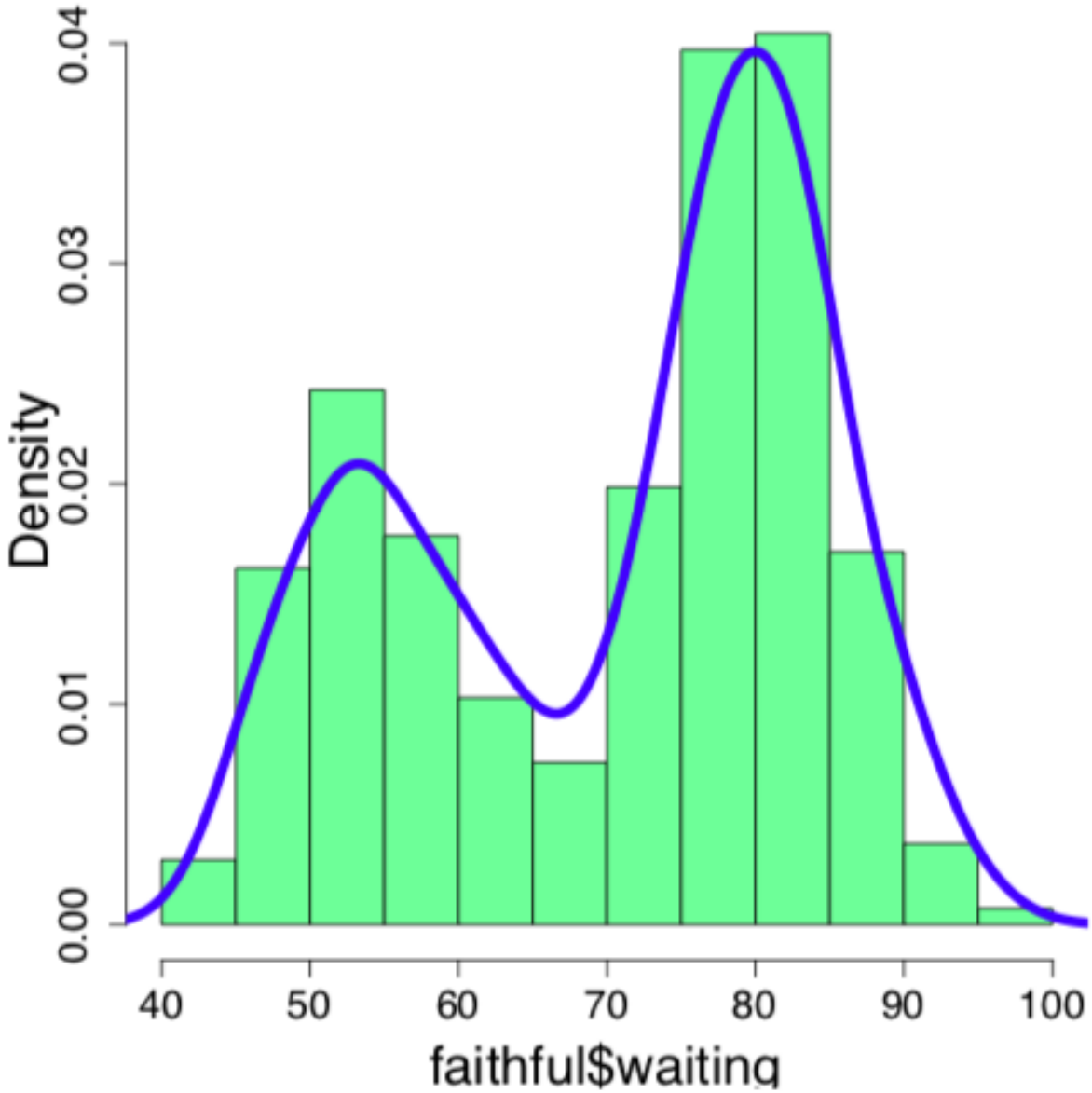
Kernel Functions

$$K(x) \geq 0, \quad \int K(x)dx = 1, \quad \int xK(x)dx = 0.$$

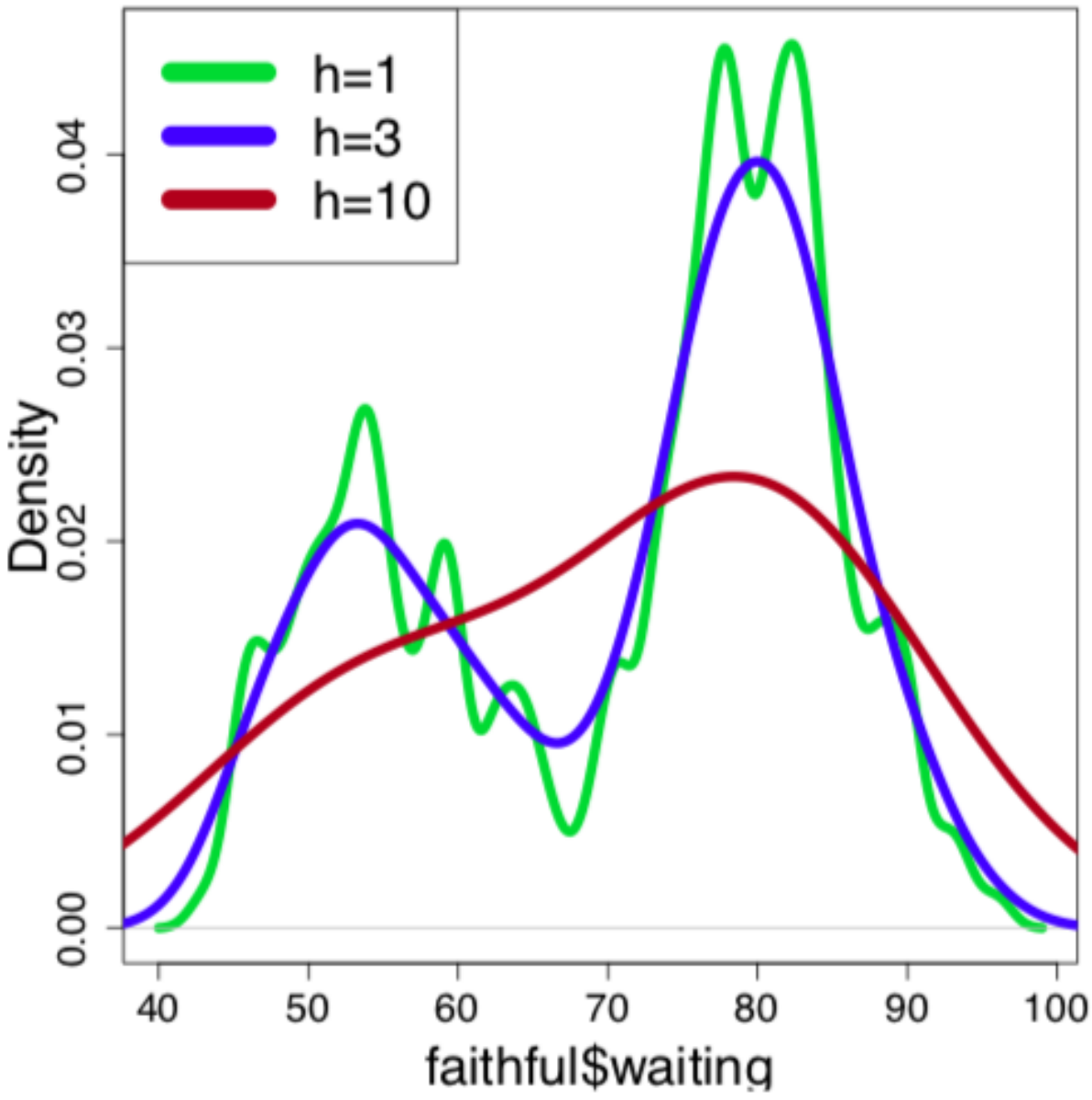
Examples are shown in Figure 1.



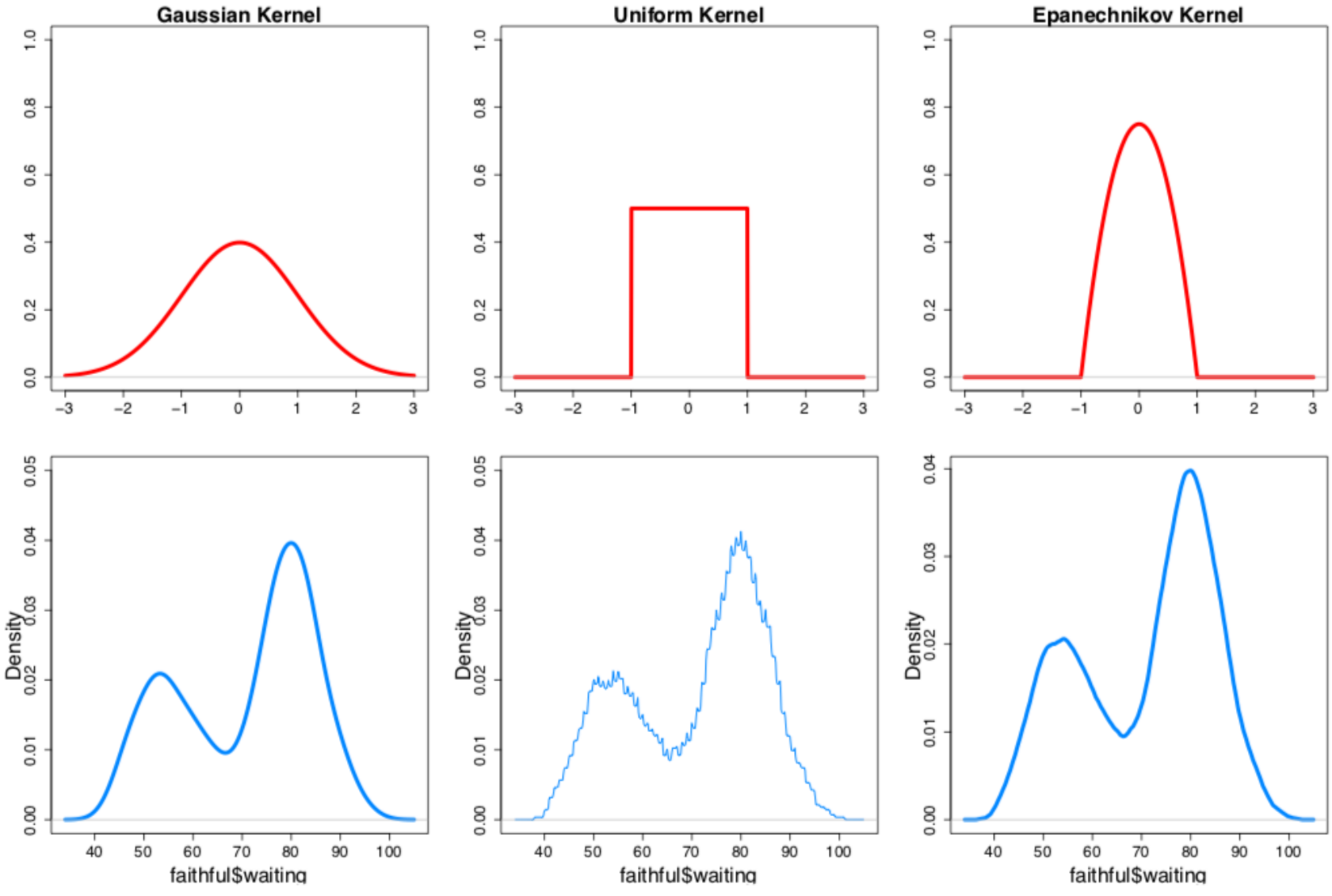
Example 1: Kernel Density Estimation for Faithful Data



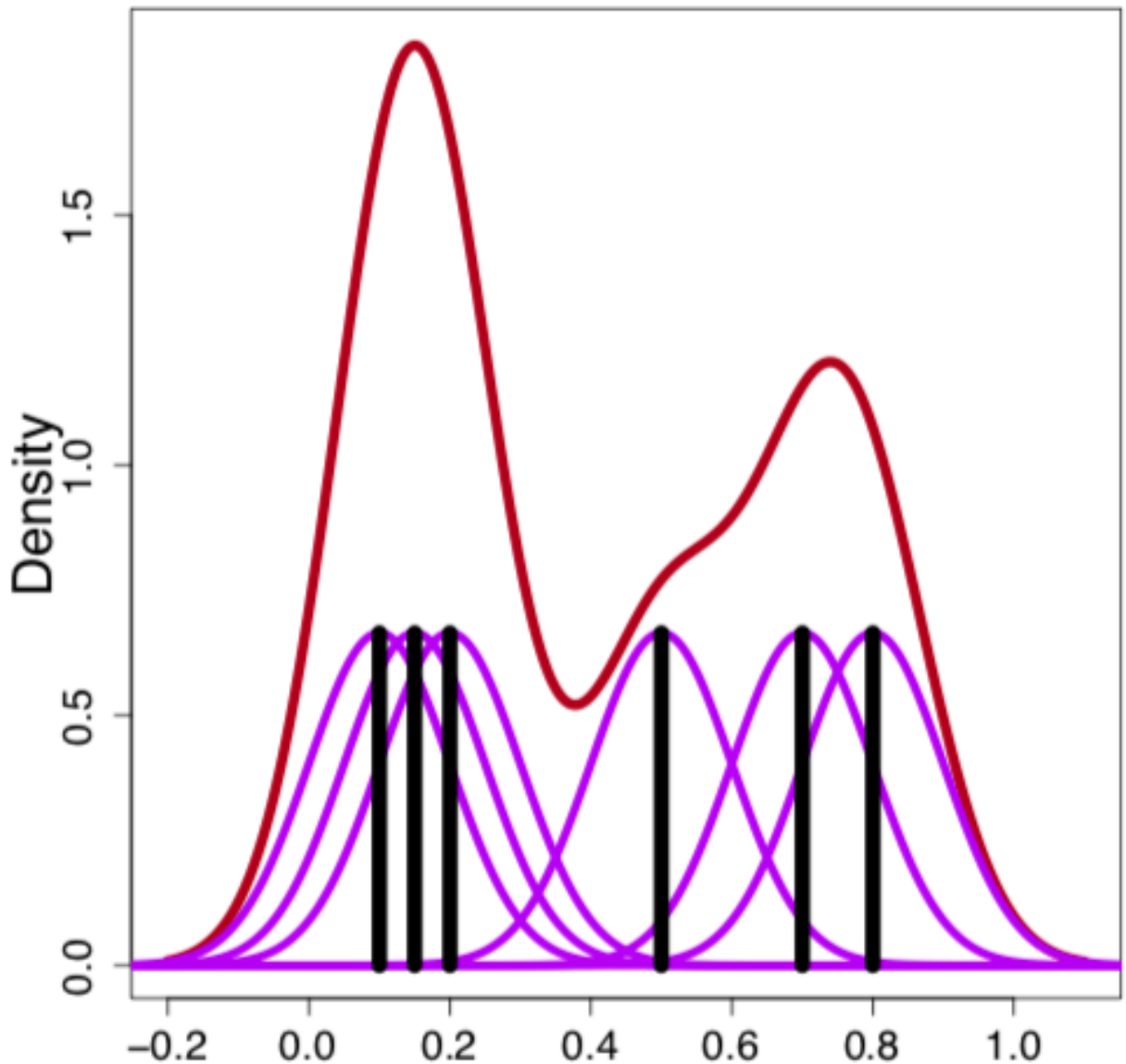
Example 1: Various Bandwidths



Example 1: Various Kernel Functions



Interpretation of Kernel Density Estimation



Multivariate Case

Now suppose that $X_1, \ldots, X_n \in \mathbb{R}^d$. The kernel estimator is

$$\widehat{p}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h^d} K\left(\frac{x - X_i}{h}\right)$$

where K is a multivariate kernel. For example, $K(x) \propto e^{-\sum_j x_j^2/(2h^2)}$.

In this case the squared bias is Ch^4 and the variance is C/nh^d so the risk is (up to constants)

$$R = h^4 + \frac{1}{nh^d}.$$

The best bandwidth is $h = (c/n)^{1/(4+d)}$. The resulting risk is

$$R = \left(\frac{C}{n}\right)^{\frac{4}{4+d}}$$

which is quite poor when d is large. This is known as *the curse of dimensionality*.

Drawback of KNN method

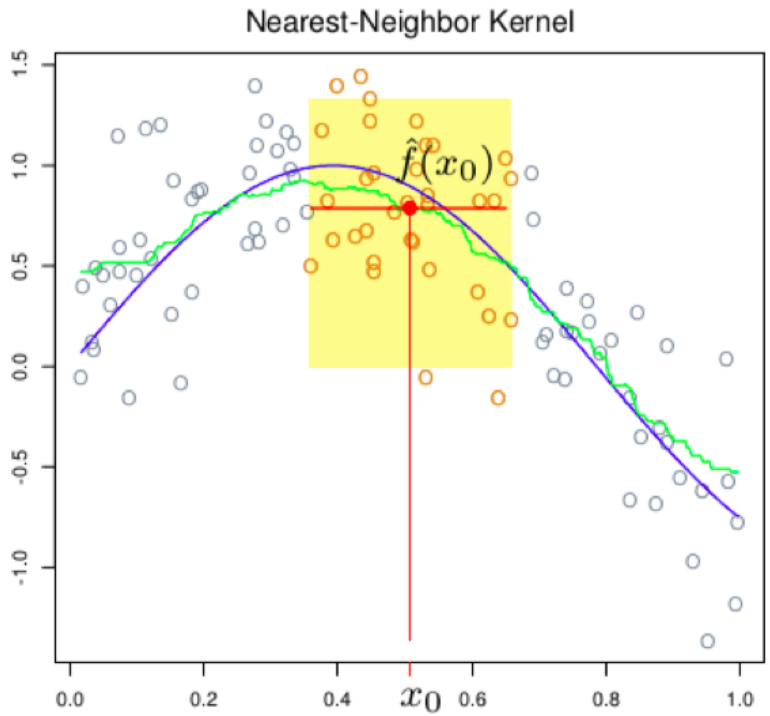
When we introduced the kNN algorithm,

$$\hat{f}(x) = \text{Ave}(y_i | x_i \in N_k(x))$$

- justified as an estimate of $E[Y|X = x]$.

Drawbacks:

- ugly discontinuities;
- same weight to all points despite their distance to x.



An example: kernel weighted method

- Nadaraya-Watson kernel-weighted average

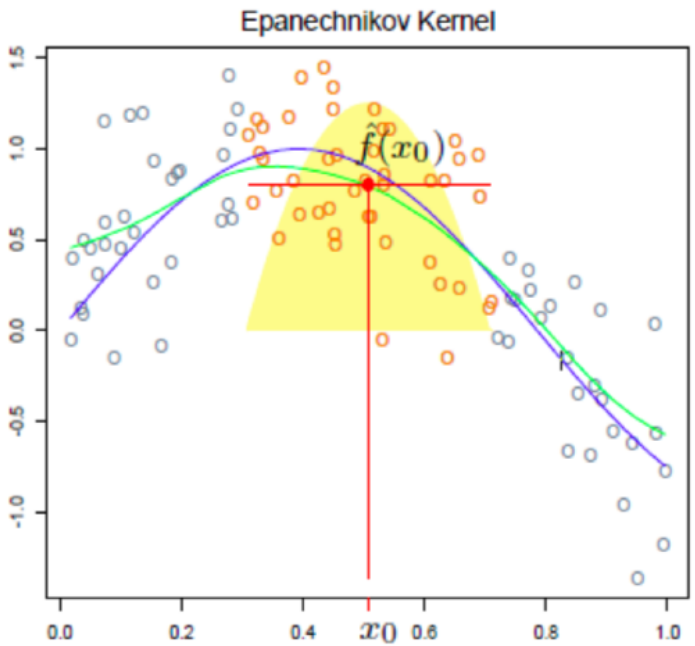
$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$

with the Epanechnikov quadratic kernel

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

with

$$D(t) = \begin{cases} \frac{3}{4}(1 - t^2) & \text{if } |t| \le 1; \\ 0 & \text{otherwise.} \end{cases}$$



Comparison

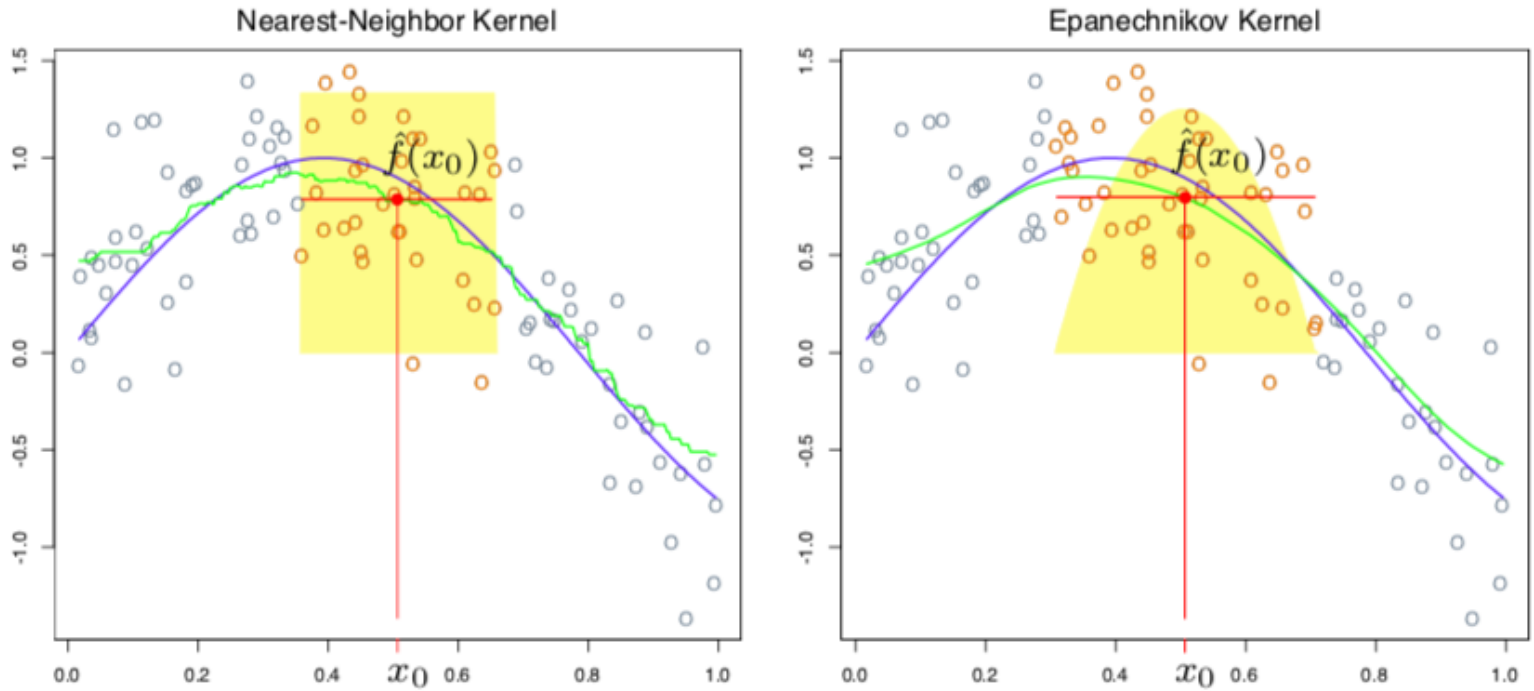


FIGURE 6.1. *In each panel* 100 *pairs* x_i, y_i *are generated at random from the blue curve with Gaussian errors:* $Y = \sin(4X) + \varepsilon$, $X \sim U[0,1]$, $\varepsilon \sim N(0, 1/3)$. *In the left panel the green curve is the result of a* 30*-nearest-neighbor running-mean smoother. The red point is the fitted constant* $\hat{f}(x_0)$, *and the red circles indicate those observations contributing to the fit at* x_0. *The solid yellow region indicates the weights assigned to observations. In the right panel, the green curve is the kernel-weighted average, using an Epanechnikov kernel with (half) window width* $\lambda = 0.2$.

Common Kernels

We need to choose $D(\cdot)$:

- symmetric around x_0;
- goes off smoothly with the distance.

Typical choices:

Nucleus	D(t)	Support
Normal	$\frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}t^2\}$	$\mathbb{R}$
Rectangular	$\frac{1}{2}$	$(-1, 1)$
Epanechnikov	$\frac{3}{4}(1-t^2)$	$(-1, 1)$
Biquadratic	$\frac{15}{16}(1-t^2)^2$	$(-1, 1)$
Tricubic	$\frac{70}{81}(1-\lvert t\rvert^3)^3$	$(-1, 1)$

Comparison of various kernels

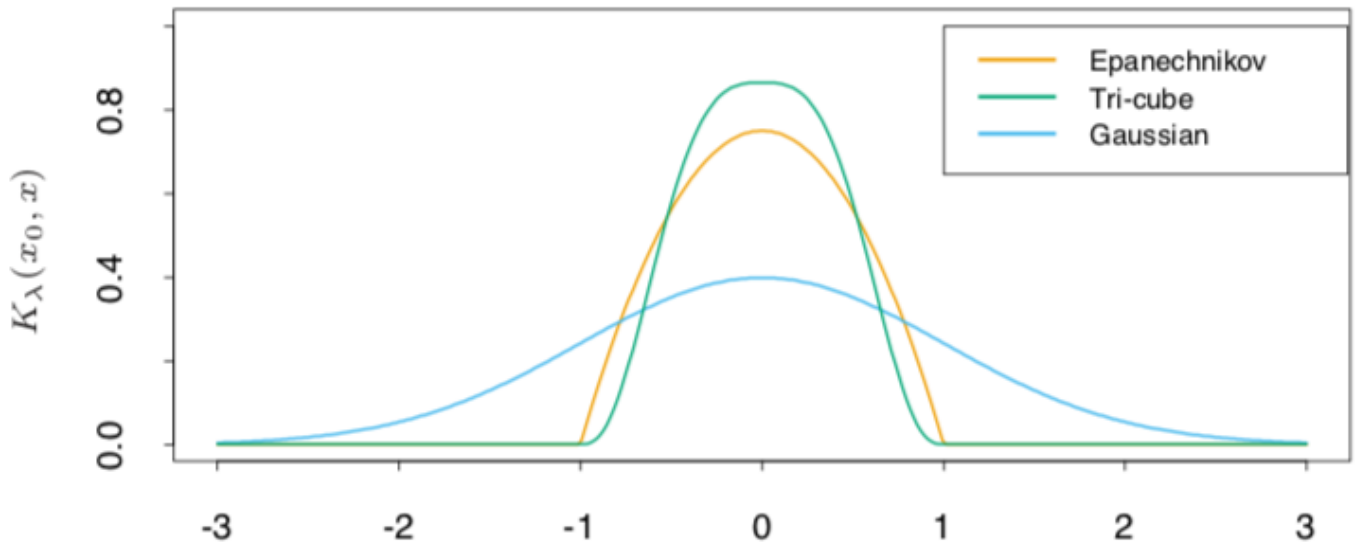
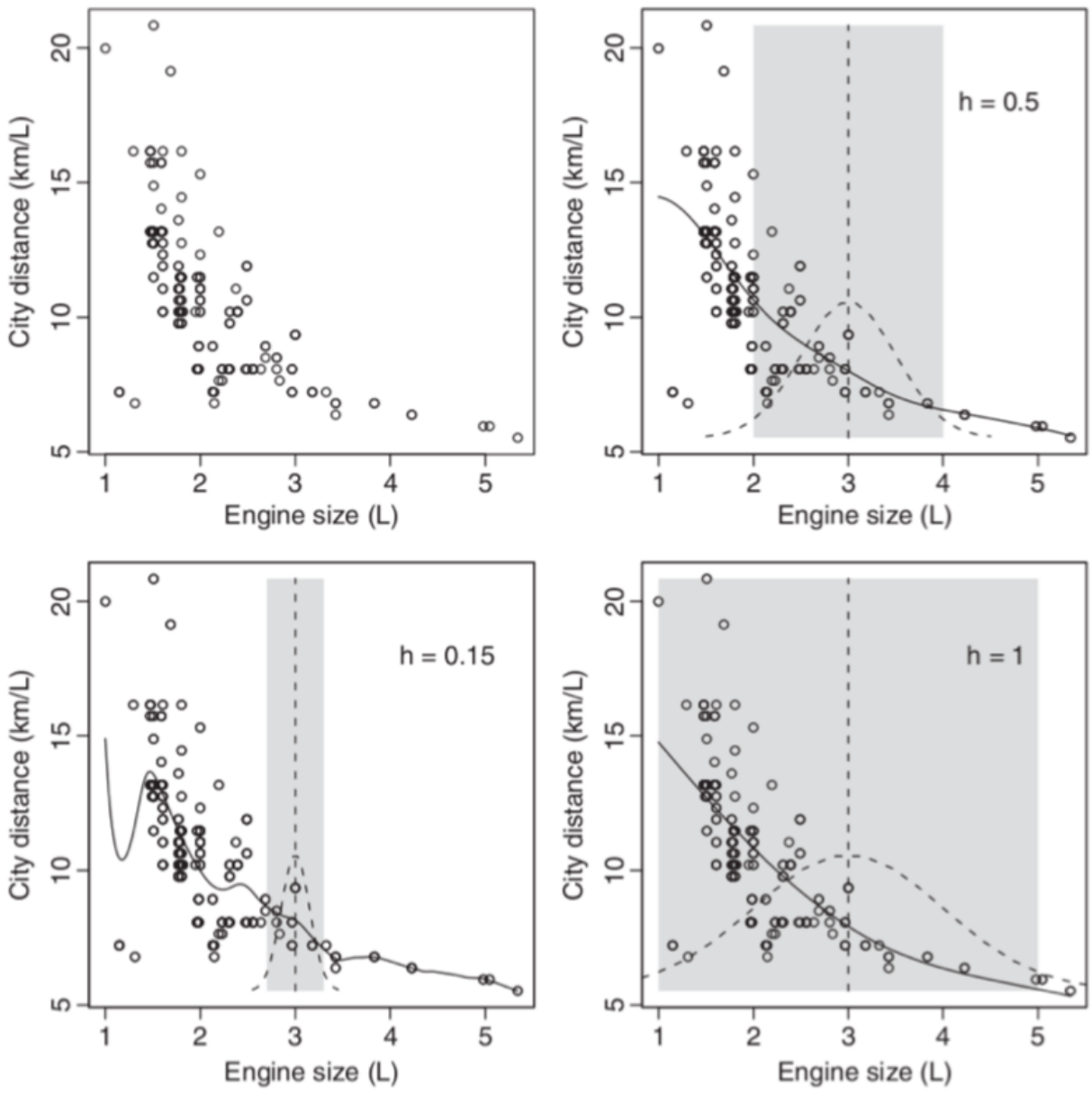


FIGURE 6.2. *A comparison of three popular kernels for local smoothing. Each has been calibrated to integrate to* 1. *The tri-cube kernel is compact and has two continuous derivatives at the boundary of its support, while the Epanechnikov kernel has none. The Gaussian kernel is continuously differentiable,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

Effect of the width parameter



Selection Criterion: based on MSE

Assume $y_i = f(x_i) + \epsilon_i$, ϵ_i i.i.d. s.t. $E[\epsilon_i] = 0$ and Var $= \sigma^2$, then

$$E[\hat{f}(x)] \approx f(x) + \frac{\lambda^2}{2}\sigma_D^2 f''(x)$$

and

$$\text{Var}[\hat{f}(x)] \approx \frac{\sigma^2}{N\lambda}\frac{R_D}{g(x)}$$

for N large and λ sufficiently close to 0 (Azzalini & Scarpa, 2012). Here:

- $\sigma_D^2 = \int t^2 D(t) dt$;
- $R_D = \int D(t)^2 dt$;
- $g(x)$ is the density from which the x_i were sampled.

Selection Criterion: based on MSE

Note:

- the bias is a multiple of λ^2;
  - $\lambda \to 0$ reduce the bias;
- the variance is a multiple of $\frac{1}{N\lambda}$;
  - $\lambda \to \infty$ reduce the variance.

The quantities $g(x)$ and $f''(x)$ are unknown, otherwise

$$\lambda_{\text{opt}} = \left(\frac{\sigma^2 R_D}{\sigma_D^4 f''(x) g(x) N} \right)^{1/5};$$

note that λ must tend to 0 with rate $N^{-1/5}$ (i.e., very slowly).

Comparison

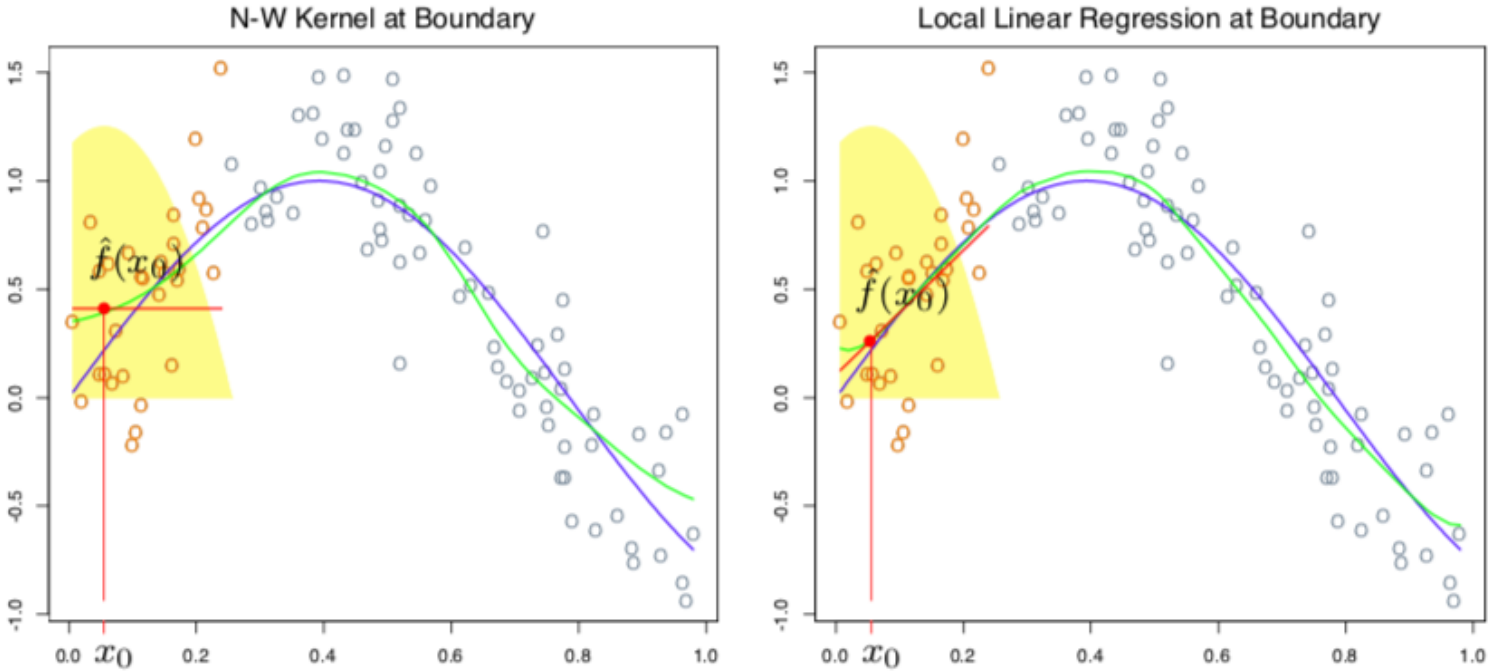


FIGURE 6.3. *The locally weighted average has bias problems at or near the boundaries of the domain. The true function is approximately linear here, but most of the observations in the neighborhood have a higher mean than the target point, so despite weighting, their mean will be biased upwards. By fitting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

Local linear regression

By fitting a straight line, we solve the problem to the first order.

$$\downarrow$$

Local linear regression

Locally weighted linear regression solve, at each target point x_0,

$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)[y_i - \alpha(x_0) - \beta(x_0)x_i]^2.$$

The estimate is $\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0$:

- the model is fit on all data belonging to the support of K_λ;
- it is only evaluated in x_0.

Estimation approach

Estimation

$$\begin{aligned}\hat{f}(x_0) &= b(x_0)^T (B^T W(x_0) B)^{-1} B^T W(x_0) y \\ &= \sum_{i=1}^{N} l_i(x_0) y_i,\end{aligned}$$

where:

- $b(x_0)^T = (1, x_0)$
- $B = (\vec{1}, X)$;
- $W(x_0)$ is a $N \times N$ diagonal matrix with i-th term $K_\lambda(x_0, x_i)$;
- $\hat{f}(x_0)$ is linear in y ($l_i(x_0)$ does not depend on y_i);
- the weights $l_i(x_0)$ are sometimes called equivalent kernels,
  - combine the weighting kernel $K_\lambda(x_0, \cdot)$ and the LS operator.

Comparison at boundary and interior

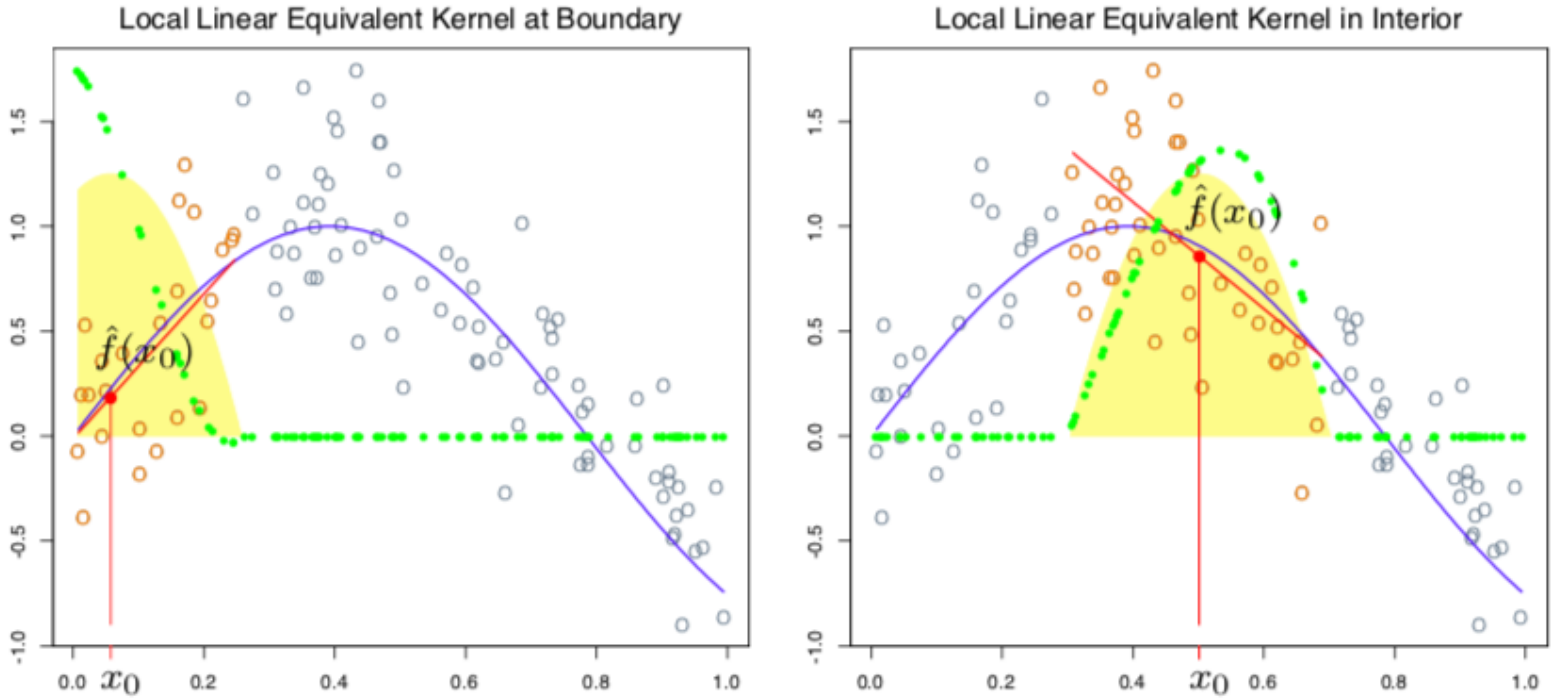


FIGURE 6.4. *The green points show the equivalent kernel $l_i(x_0)$ for local regression. These are the weights in $\hat{f}(x_0) = \sum_{i=1}^{N} l_i(x_0) y_i$, plotted against their corresponding x_i. For display purposes, these have been rescaled, since in fact they sum to 1. Since the yellow shaded 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.*

Analysis of bias

Using a Taylor expansion of $f(x_i)$ around x_0,

$$\begin{aligned} E[\hat{f}(x_0)] &= \sum_{i=1}^{N} l_i(x_0) f(x_i) \\ &= f(x_0) \sum_{i=1}^{N} l_i(x_0) + f'(x_0) \sum_{i=1}^{N} (x_i - x_0) l_i(x_0) + \\ &\quad + \frac{f''(x_0)}{2} \sum_{i=1}^{N} (x_i - x_0)^2 l_i(x_0) + \dots \end{aligned} \tag{2}$$

For local linear regression,

- $\sum_{i=1} l_i(x_0) = 1$;
- $\sum_{i=1}^{N} (x_i - x_0) l_i(x_0) = 0$.

Therefore,

- $E[\hat{f}(x_0)] - f(x_0) = \frac{f''(x_0)}{2} \sum_{i=1}^{N} (x_i - x_0)^2 l_i(x_0) + \dots$.

Local polynomial regression

Why limiting to a linear fit?

$$\min_{\alpha(x_0),\beta_1(x_0),\ldots,\beta_d(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i) \left[y_i - \alpha(x_0) - \sum_{j=1}^{d} \beta_j(x_0) x_i^j \right]^2,$$

with solution $f(\hat{x_0}) = \hat{\alpha}(x_0) + \sum_{j=1}^{d} \hat{\beta}(x_0) x_0^j$.

- it can be shown that the bias, using (2), only involves components of degree $d + 1$;
- in contrast to local linear regression, it tends to be closer to the true function in regions with high curvature,
  - no *trimming the hills and filling the gaps* effect.

Comparison

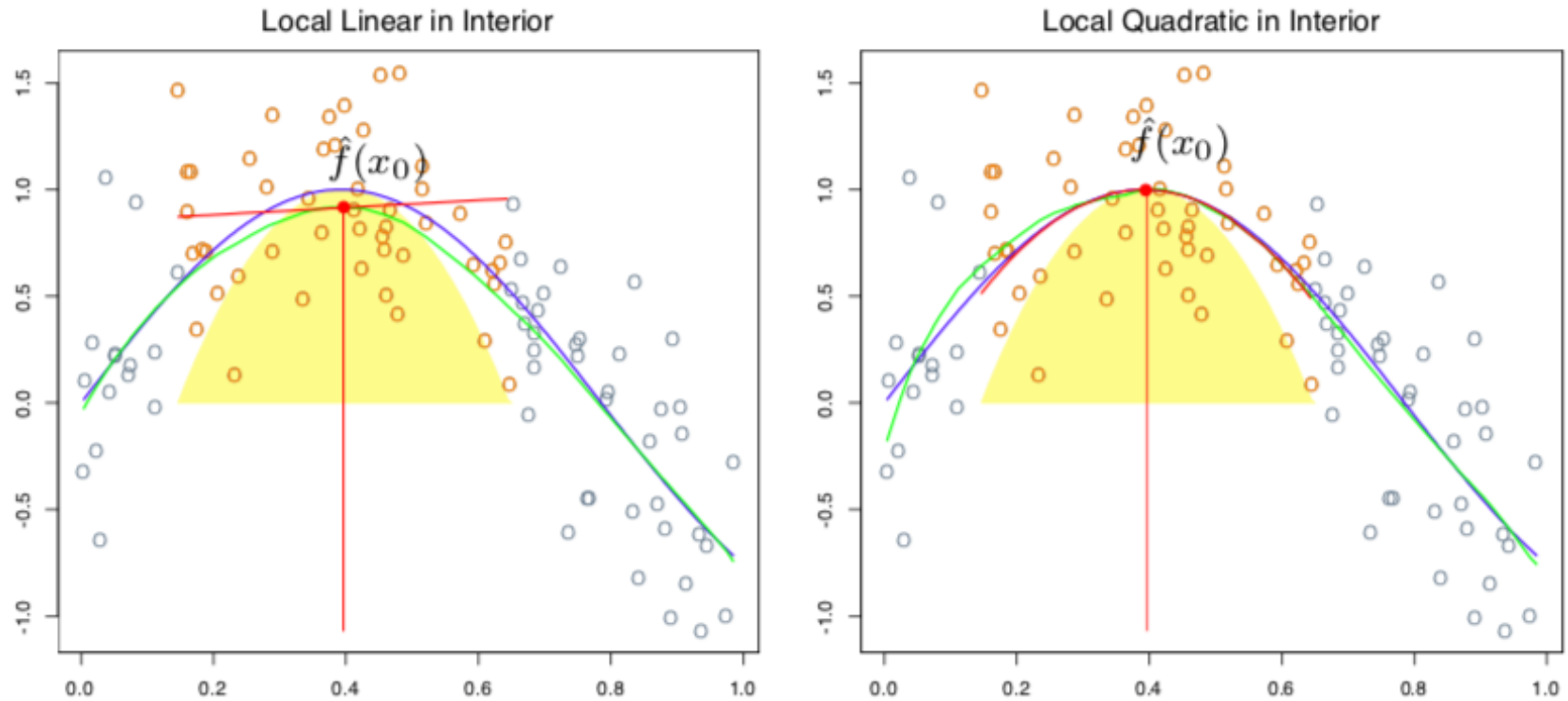


FIGURE 6.5. *Local linear fits exhibit bias in regions of curvature of the true function. Local quadratic fits tend to eliminate this bias.*

Analysis of variance

Not surprisingly, there is a price for having less bias.

Assuming a model $y_i = f(x_i) + \epsilon_i$, where ϵ_i are i.i.d. with mean 0 and variance σ^2,

$$\text{Var}(\hat{f}(x_i)) = \sigma^2 ||l(x_0)||$$

It can be shown that $||l(x_0)||$ increase with d $\Rightarrow$ bias-variance trade-off in the choice of d.

Comparison

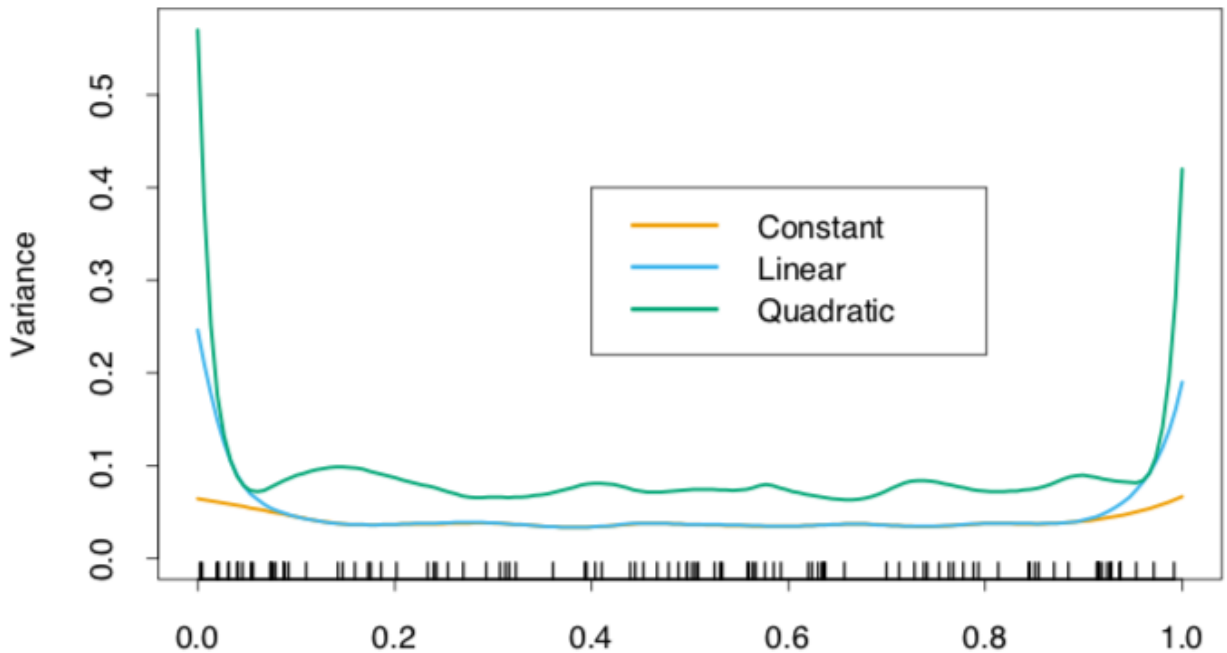


FIGURE 6.6. *The variances functions* $||l(x)||^2$ *for local constant, linear and quadratic regression, for a metric bandwidth* ($\lambda = 0.2$) *tri-cube kernel.*

Trade-off between bias and variance

- local linear fits help dramatically in alleviating boundary issues;
- quadratic fits do a little better, but increase variance;
- quadratic fits solve issues in high curvature regions;
- asymptotic analyses suggest that polynomials of odd degrees should be preferred to those of even degrees,
  - the MSE is asymptotically dominated by boundary effects;
- anyway, the choice of d is problem specific.

Multiple local polynomial regression

Let $b(X)$ be a vector of polynomial terms in X of maximum degree d. For example, with $d = 1$ and $p = 2$ we get $b(X) = (1, X_1, X_2)$; with $d = 2$ we get $b(X) = (1, X_1, X_2, X_1^2, X_2^2, X_1X_2)$; and trivially with $d = 0$ we get $b(X) = 1$. At each $x_0 \in \mathbb{R}^p$ solve

$$\min_{\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)(y_i - b(x_i)^T \beta(x_0))^2 \tag{6.12}$$

to produce the fit $\hat{f}(x_0) = b(x_0)^T \hat{\beta}(x_0)$. Typically the kernel will be a radial function, such as the radial Epanechnikov or tri-cube kernel

$$K_\lambda(x_0, x) = D\left(\frac{||x - x_0||}{\lambda}\right), \tag{6.13}$$

Example: local regression in $\mathbb{R}^2$

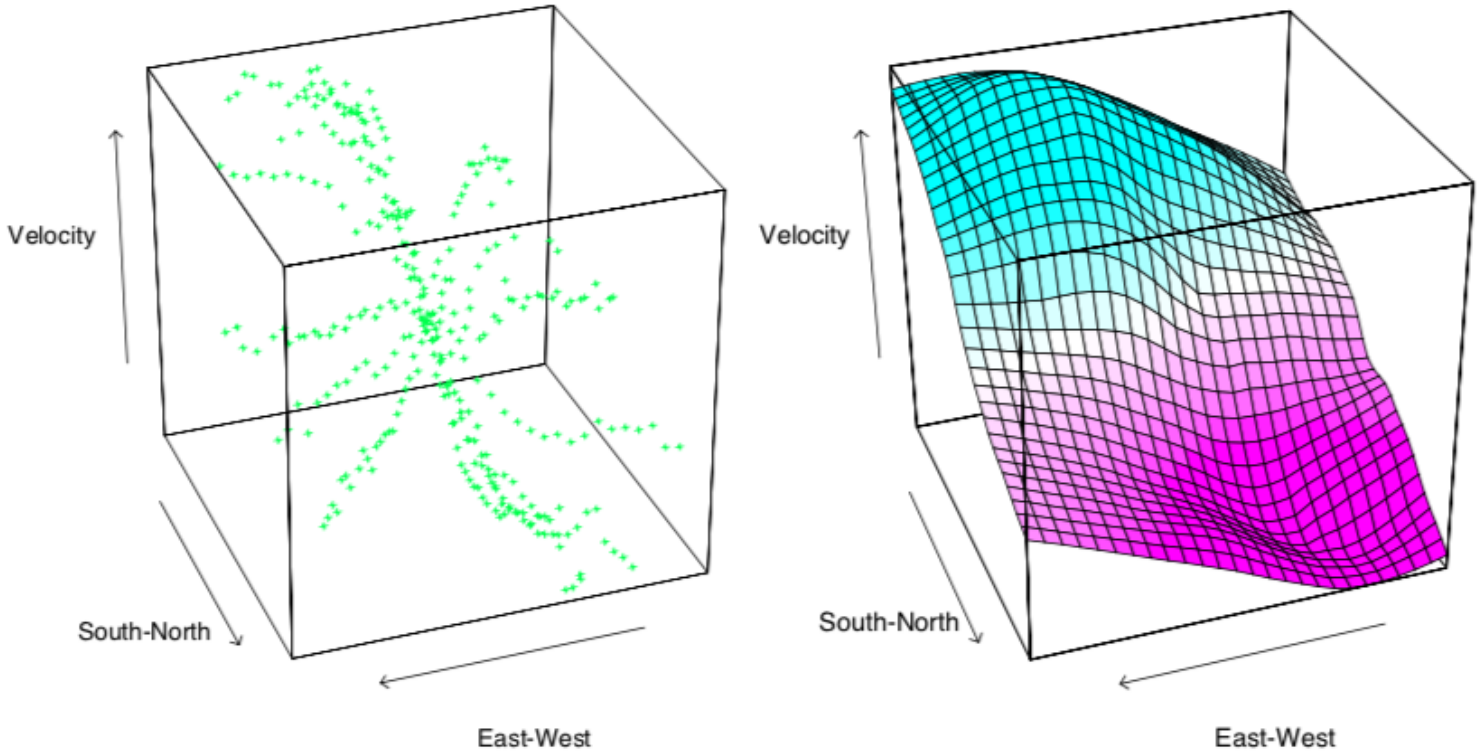


FIGURE 6.8. *The left panel shows three-dimensional data, where the response is the velocity measurements on a galaxy, and the two predictors record positions on the celestial sphere. The unusual "star"-shaped design indicates the way the measurements were made, and results in an extremely irregular boundary. The right panel shows the results of local linear regression smoothing in* $\mathbb{R}^2$*, using a nearest-neighbor window with* 15% *of the data.*

Difficulty of local polynomial regression

- boundary issues are even more dramatic than in one dimension;
  - the fraction of points at the boundary increases to 1 by increasing the dimensions;
  - curse of dimensionality;
- local polynomials still perform boundary corrections up to the desired order;
- local regression does not make really sense for $d > 3$,
  - it is impossible to maintain localness (small bias) and sizeable sample in the neighbourhood (small variance);
  - again, curse of dimensionality.

Structured kernels

$$K_{\lambda,A}(x_0, x) = D\left(\frac{(x - x_0)^T A(x - x_0)}{\lambda}\right)$$

- A is a matrix semidefinite positive;
- we can add structures through A:
  - A diagonal, increase or decrease the importance of the predictor X_j by increasing/decreasing a_{jj};
  - low rank versions of $A \rightarrow$ projection pursuit;

Structured regression functions

$$f(X_1, \ldots, X_p) = \alpha + \sum_{j=1}^{p} g_j(X_j) + \sum_{k<\ell} g_{k\ell}(X_k, X_\ell) + \ldots$$

- we can simplify the structure;
- examples:
  - remove all interaction terms,
    $f(X_1, \ldots, X_p) = \alpha + \sum_{j=1}^{p} g_j(X_j)$;
  - keep only the first order interactions,
    $f(X_1, \ldots, X_p) = \alpha + \sum_{j=1}^{p} g_j(X_j) + \sum_{k<\ell} g_{k\ell}(X_k, X_\ell)$;
  - ...

Varying coefficient models

The varying coefficient models:

- are a special case of structured regression functions;
- consider only $q < p$ predictors, all the remaining are in Z;
- assume the conditionally linear model,

$$f(X) = \alpha(Z) + \beta_1(Z)X_1 + \cdots + \beta_q(Z)X_q;$$

- given Z, it is a linear model,
  - solution via least squares estimator;
- the coefficients can vary with Z.

Example: varying-coefficient model

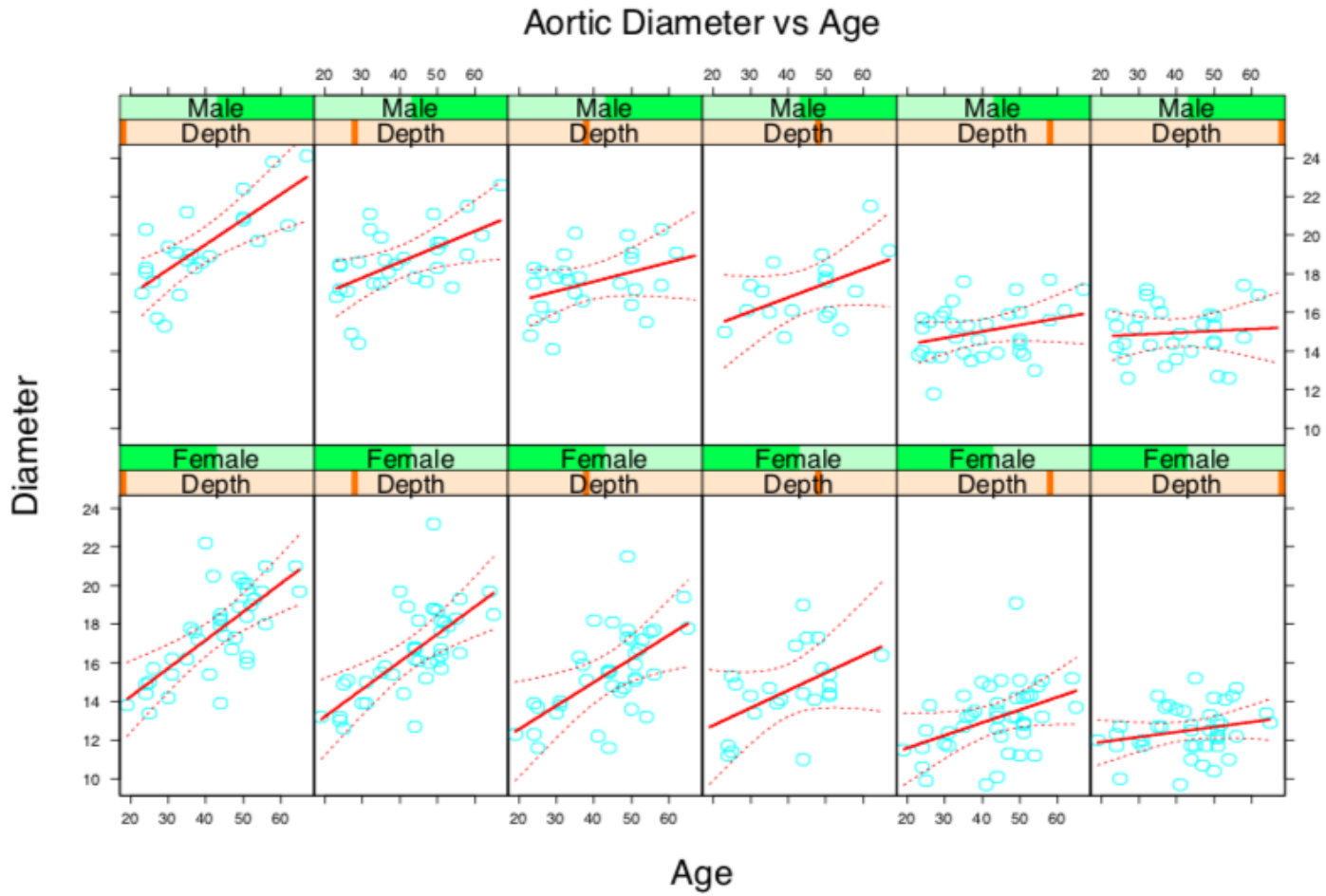
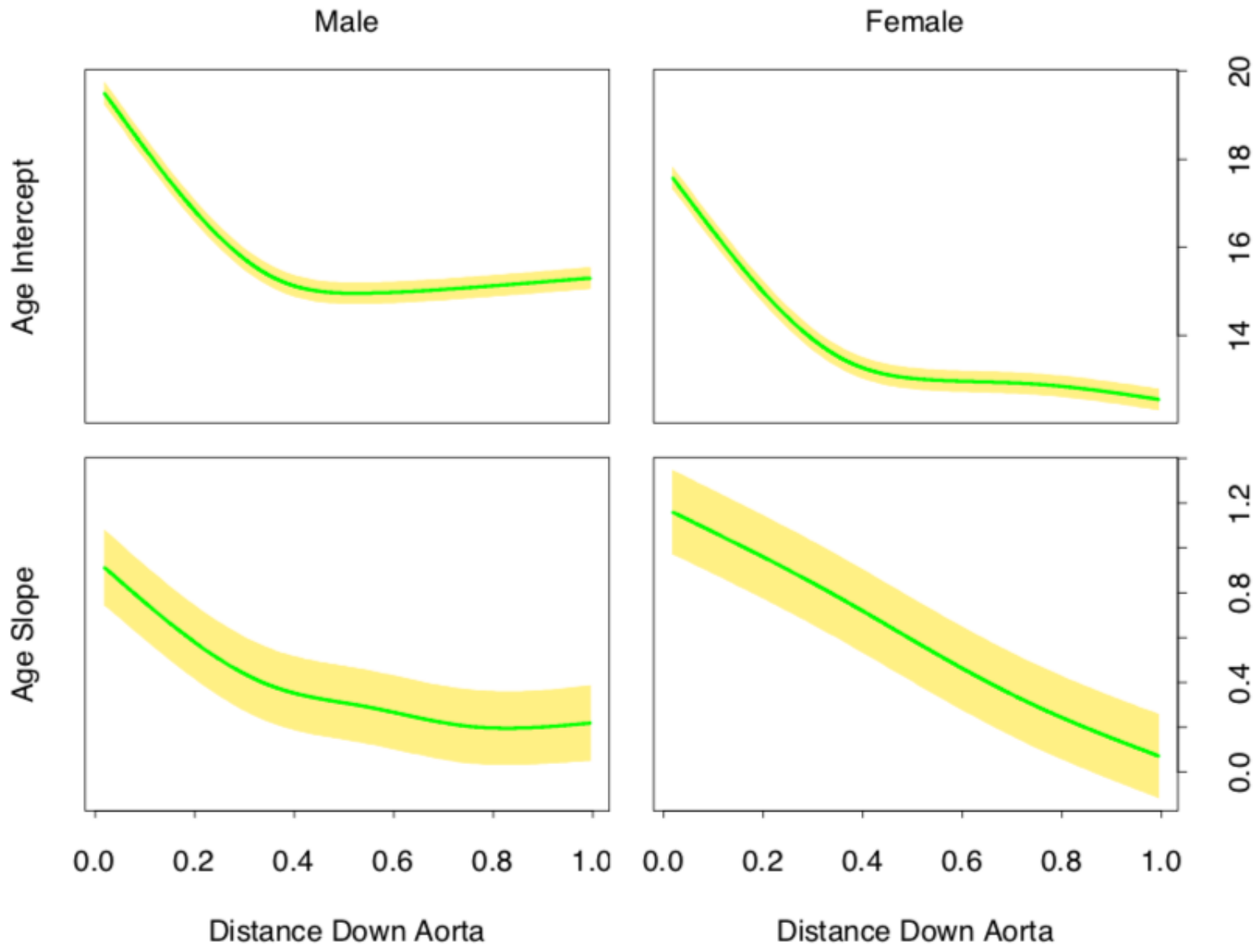


FIGURE 6.10. *In each panel the* `aorta diameter` *is modeled as a linear function of* `age`*. The coefficients of this model vary with* `gender` *and* `depth` *down the* `aorta` *(left is near the top, right is low down). There is a clear trend in the coefficients of the linear model.*

Example: varying-coefficient model



Conclusion

Understand

- Kernel Smoothing method to estimate Density Function and Regression Function
- Curse of Dimensionality of Kernel Smoothing Method on High-dimensional Data
- Alternative Non-parametric Methods (on model or estimation approach) to Overcome Curse of Dimensionality