Nonparametric Estimation

Statistics for Data Science
CSE357 - Fall 2021

Nonparametric Estimation

Uses

- Data visualization and exploration
- Estimating a function without knowing function structure

How? examples...

- Kernel Density Estimation,
- Histograms
- Local regression (lowess, loess)
- Smoothing

Nonparametric Estimation

Why?

Besides tools for exploring data, can yield a deeper understand of the trade-offs at play with fitting a model to data.

Bias, Variance Tradeoff

Why?

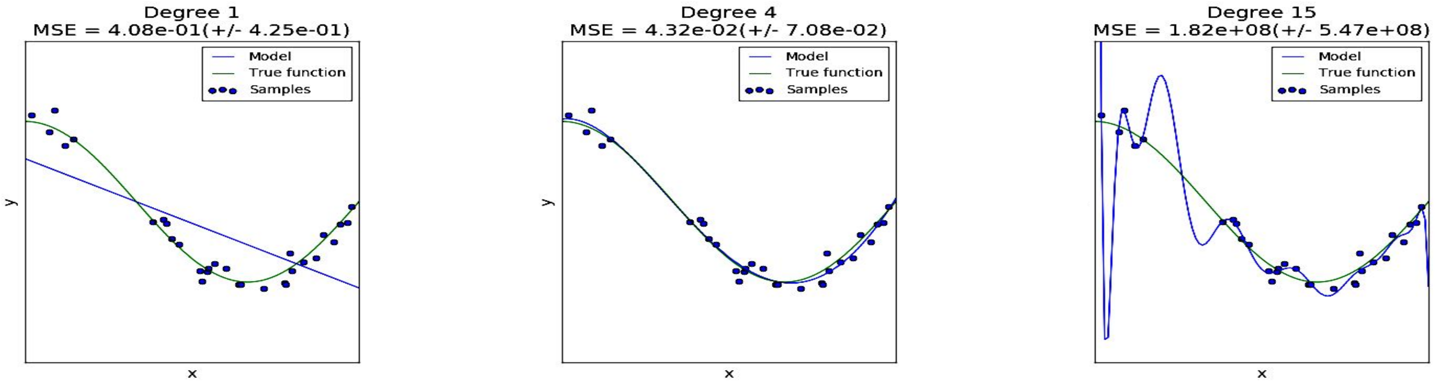
Besides tools for exploring data, can yield a deeper understand of the **trade-offs** at play with fitting a model to data.

Bias, Variance Tradeoff

Conceptually: As a model produces less <u>variance</u> (i.e. by penalizing the beta coefficients), it increases <u>bias.</u>

Bias, Variance Tradeoff

Conceptually: As a model produces less __variance__ (i.e. by penalizing the beta coefficients), it increases __bias.__



too much bias: underfit

too much variance: overfit

Bias, Variance Tradeoff

Formally,

$\hat{g}_n$-- estimator of true function, g (regression or density)

bias: $b(x) = \mathbb{E}(\hat{g}_n(x)) - g(x)$

variance: $v(x) = \mathbb{V}(\hat{g}_n(x)) = \mathbb{E}((\hat{g}_n(x)) - \mathbb{E}(\hat{g}_n(x))^2)$

Bias, Variance Tradeoff

Formally,

$\hat{g}_n$ -- estimator of true function, g (regression or density)

Risk: $R(g, \hat{g}_n) = \int b^2(x)dx + \int v(x)dx$

bias: $b(x) = \mathbb{E}(\hat{g}_n(x)) - g(x)$

variance: $v(x) = \mathbb{V}(\hat{g}_n(x)) = \mathbb{E}((\hat{g}_n(x)) - \mathbb{E}(\hat{g}_n(x))^2)$

Bias, Variance Tradeoff

$$R(g, \hat{g}_n) = \int b^2(x)dx + \int v(x)dx$$
$$b(x) = \mathbb{E}(\hat{g}_n(x)) - g(x)$$
$$v(x) = \mathbb{V}(\hat{g}_n(x)) = \mathbb{E}((\hat{g}_n(x)) - \mathbb{E}(\hat{g}_n(x))^2)$$

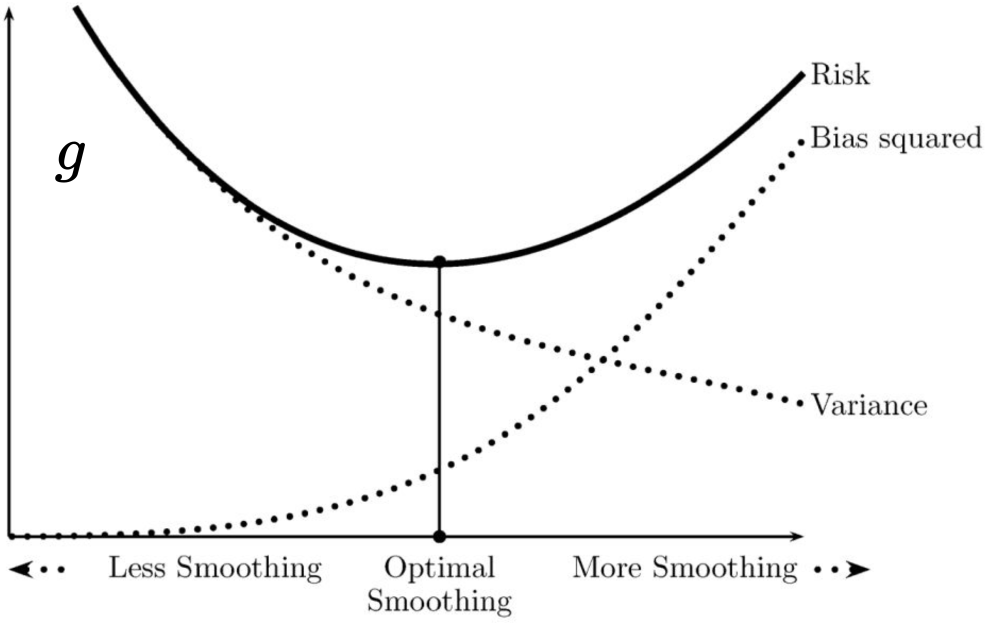


FIGURE 20.2. The Bias-Variance trade-off. The bias increases and the variance decreases with the amount of smoothing. The optimal amount of smoothing, indicated by the vertical line, minimizes the risk = bias^2 + variance.

(Wasserman, 2005, AoS)

Bias, Variance Tradeoff

$$R(g, \hat{g}_n) = \int b^2(x)dx + \int v(x)dx$$
$$b(x) = \mathbb{E}(\hat{g}_n(x)) - g(x)$$
$$v(x) = \mathbb{V}(\hat{g}_n(x)) = \mathbb{E}((\hat{g}_n(x)) - \mathbb{E}(\hat{g}_n(x))^2)$$

Risk = Bias^2 + Variance

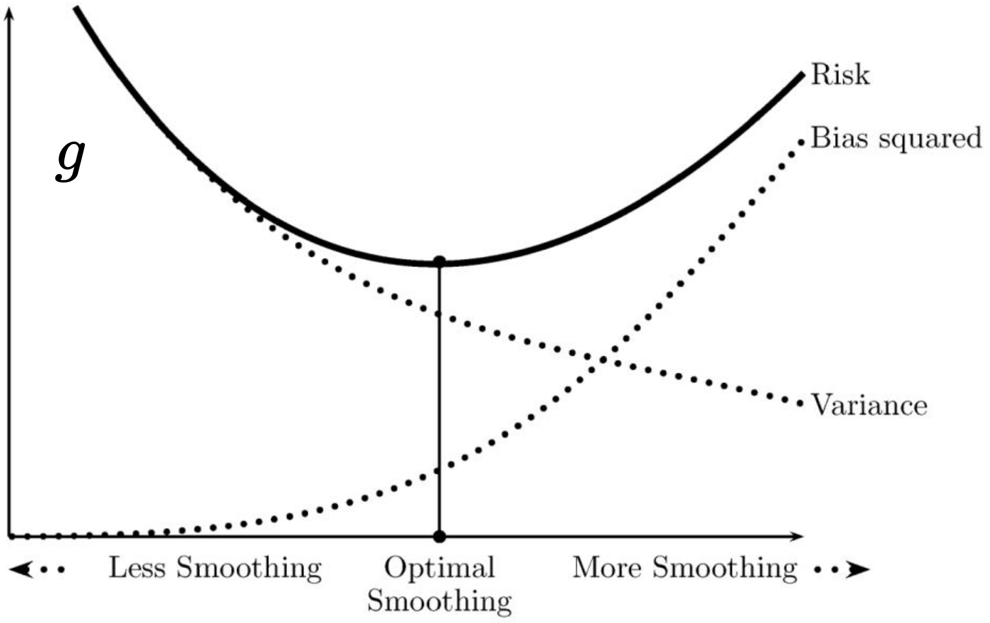


FIGURE 20.2. The Bias-Variance trade-off. The bias increases and the variance decreases with the amount of smoothing. The optimal amount of smoothing, indicated by the vertical line, minimizes the risk = bias^2 + variance.

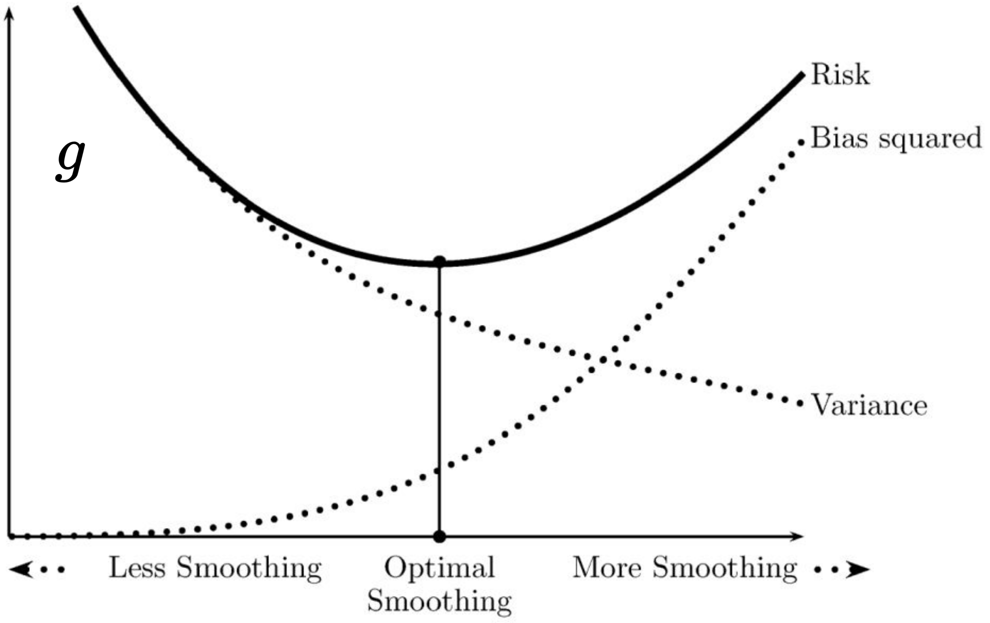
(Wasserman, 2005, AoS)

Nonparametric Regression

$$R(g, \hat{g}_n) = \int b^2(x)dx + \int v(x)dx$$
$$b(x) = \mathbb{E}(\hat{g}_n(x)) - g(x)$$
$$v(x) = \mathbb{V}(\hat{g}_n(x)) = \mathbb{E}((\hat{g}_n(x)) - \mathbb{E}(\hat{g}_n(x))^2)$$

(Wasserman, 2005, AoS)

Risk = Bias² + Variance



Conceptually:

As a model produces less variance (i.e. by penalizing the beta coefficients), it increases bias.

Nonparametric Regression

Regression: $Y_i = r(x_i) + \epsilon_i$

Nadaraya-Watson kernel estimator:

$$\hat{r} = \sum_{i=1}^{n} w_i(x) Y_i$$

where weights are given by (*K* is a kernel):

$$w_i(x) = \frac{K(\frac{x - x_i}{h})}{\sum_{j=1}^{n} K(\frac{x - x_j}{h})}$$

(Wasserman)

Nonparametric Regression

Regression: $Y_i = r(x_i) + \epsilon_i$

Nadaraya-Watson kernel estimator:

$$\hat{r} = \sum_{i=1}^{n} w_i(x) Y_i$$

where weights are given by (*K* is a kernel):

$$w_i(x) = \frac{K(\frac{x-x_i}{h})}{\sum_{j=1}^{n} K(\frac{x-x_j}{h})}$$

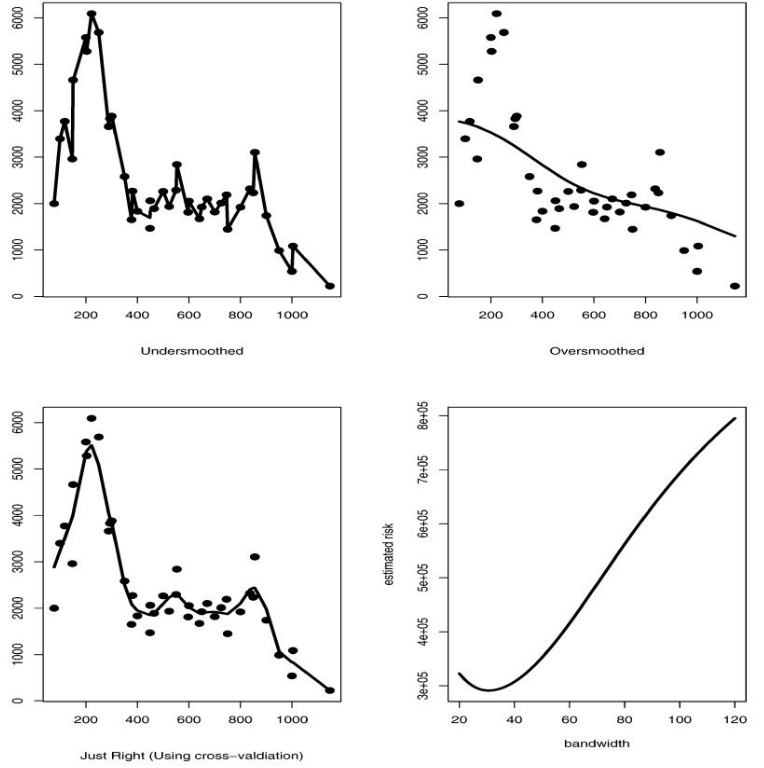


FIGURE 20.8. Regression analysis of the CMB data. The first fit is undersmoothed, the second is oversmoothed, and the third is based on cross-validation. The last panel shows the estimated risk versus the bandwidth of the smoother. The data are from BOOMERaNG, Maxima, and DASI.

(Wasserman)

Nonparametric Regression

Regression: $Y_i = r(x_i) + \epsilon_i$

Nadaraya-Watson kernel estimator:

$$\hat{r} = \sum_{i=1}^{n} w_i(x) Y_i$$

where weights are given by (*K* is a kernel):

$$w_i(x) = \frac{K(\frac{x - x_i}{h})}{\sum_{j=1}^{n} K(\frac{x - x_j}{h})}$$

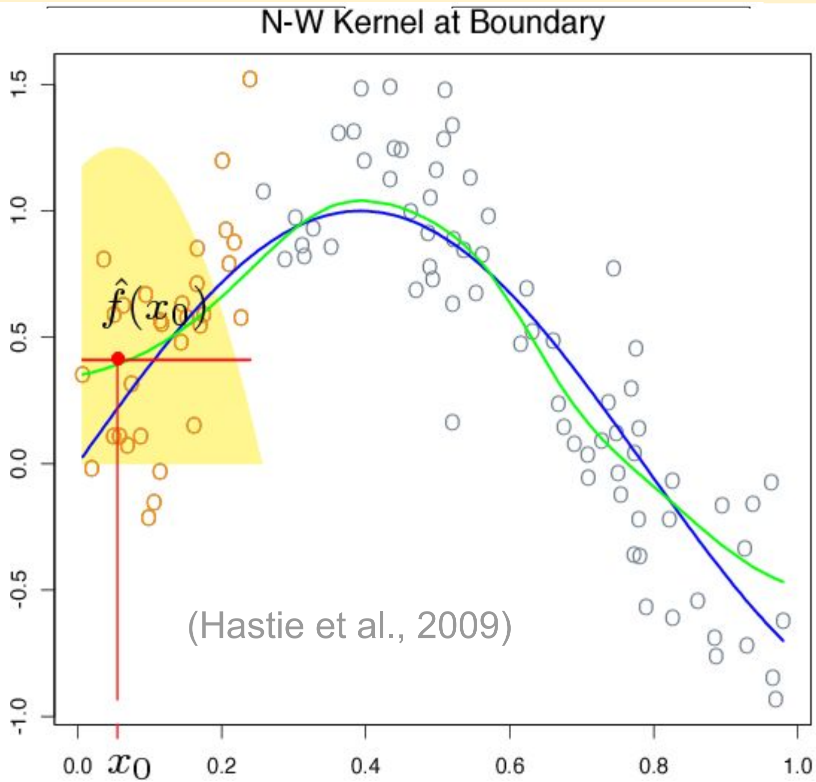


FIGURE 20.8. Regression analysis of the CMB data. The first fit is undersmoothed, the second is oversmoothed, and the third is based on cross-validation. The last panel shows the estimated risk versus the bandwidth of the smoother. The data are from BOOMERaNG, Maxima, and DASI.

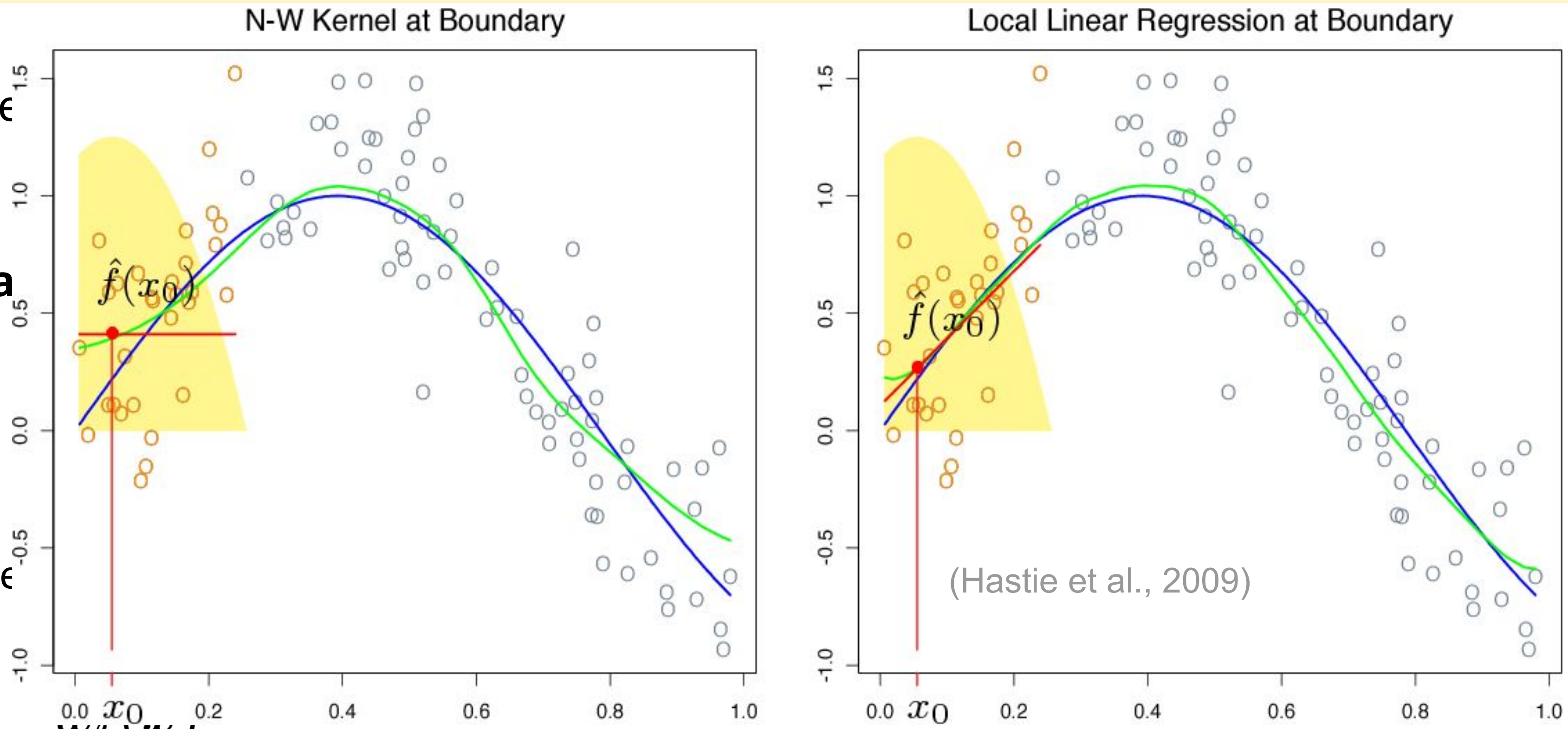
(Wasserman)

Nonparametric Regression

Regre

Nada

where



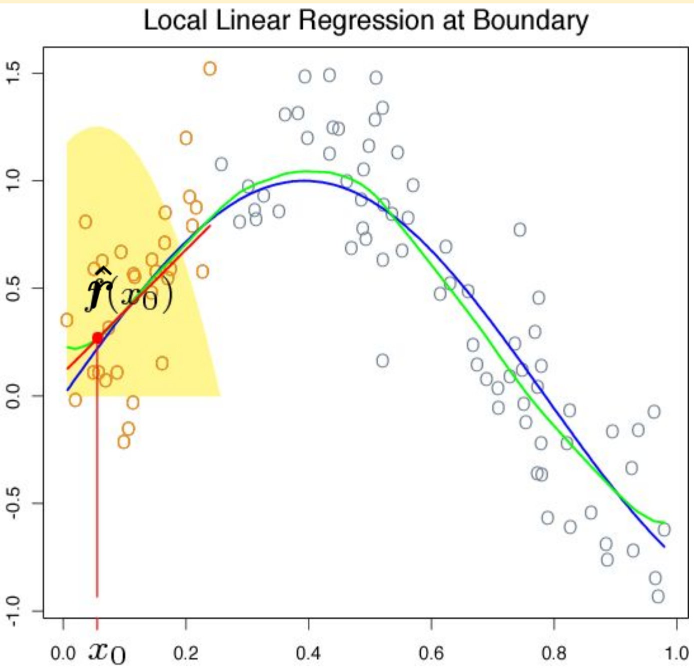
$\sum_{j=1}^{n} K(\frac{x-x_i}{h})$

Nonparametric Regression

Regression: $$Y_i = r(x_i) + \epsilon_i$$

Local Linear Regression:

$$\hat{r}(x_0) = \hat{\beta}_0(x_0) + \hat{\beta}_1(x_0)x_0$$



(Hastie et al., 2009)

Nonparametric Regression

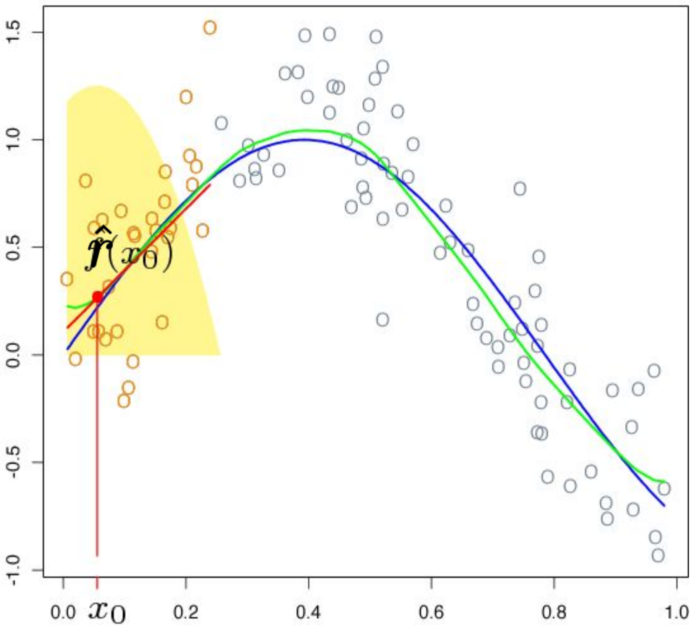
Regression: $$Y_i = r(x_i) + \epsilon_i$$

Local Linear Regression:

$$\hat{r}(x_0) = \hat{\beta}_0(x_0) + \hat{\beta}_1(x_0)x_0$$

local regression coefficients
x_0: a range of points around a given x_i;
e.g. the 30 nearest neighbors

Local Linear Regression at Boundary



(Hastie et al., 2009)

Nonparametric Regression

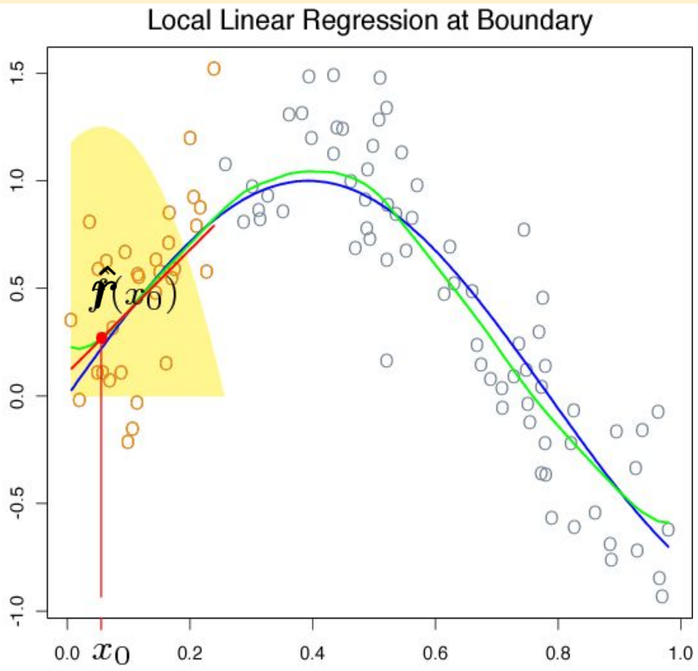
Regression: $$Y_i = r(x_i) + \epsilon_i$$

Local Linear Regression:

$$\hat{r}(x_0) = \hat{\beta}_0(x_0) + \hat{\beta}_1(x_0)x_0$$

Objective: Minimize weighted least squares:

$$\sum_{i=1}^{n} K_\lambda(x_0, x_i)[y_i - \beta_0(x_0) - \beta_1(x_0)x_i]^2$$



(Hastie et al., 2009)

Nonparametric Regression

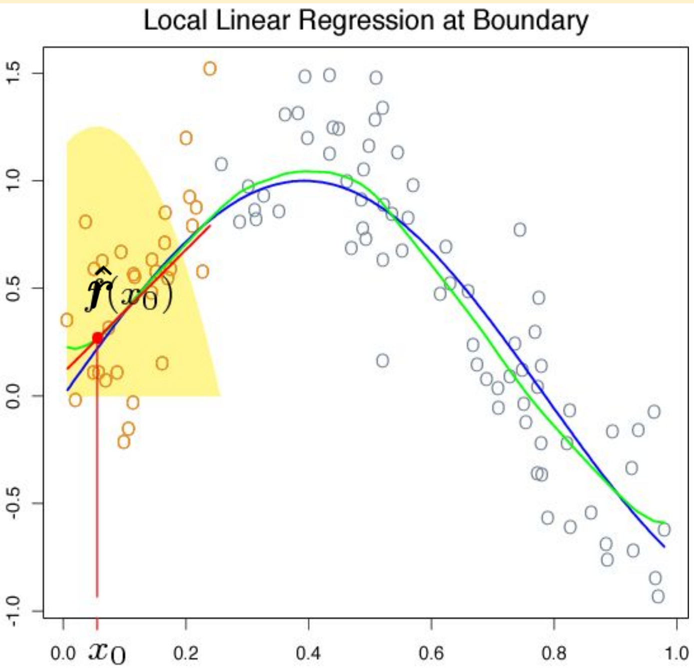
Regression: $$Y_i = r(x_i) + \epsilon_i$$

Local Linear Regression:

$$\hat{r}(x_0) = \hat{\beta}_0(x_0) + \hat{\beta}_1(x_0)x_0$$

Objective: Minimize weighted least squares:

$$\sum_{i=1}^{n} K_\lambda(x_0, x_i)[y_i - \beta_0(x_0) - \beta_1(x_0)x_i]^2$$



Full solution for local linear regression with normal equations for weighted least squares:

$$\hat{r}(x_0) = b(x_0)^T(\beta^T W(x_0)\beta)^{-1}\beta^T W(x_0)y$$

(Hastie et al., 2009)

Nonparametric Regression

Regression: $$Y_i = r(x_i) + \epsilon_i$$

Local Linear Regression:

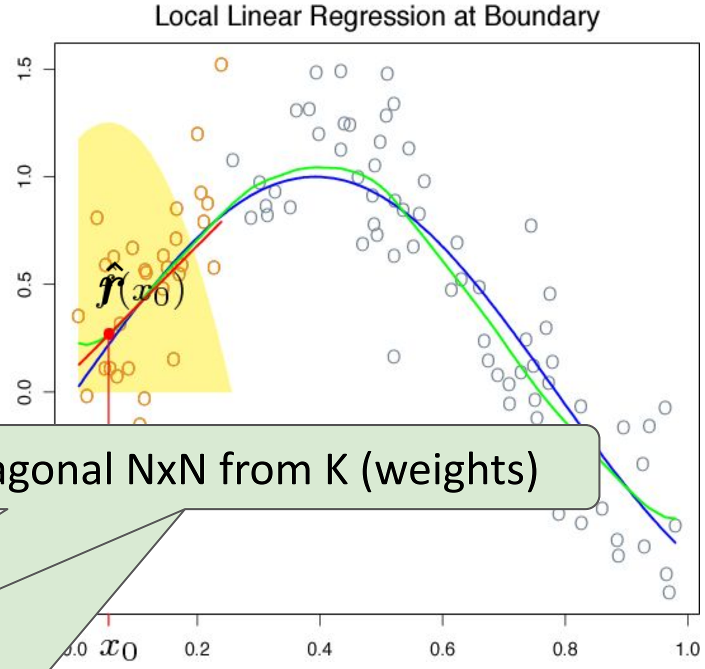
$$\hat{r}(x_0) = \hat{\beta}_0(x_0) + \hat{\beta}_1(x_0)x_0$$

Objective: Minimiz … s:

$$\sum_{i=1}^{n} K_\lambda(x_0, x_i)[y_i \ldots _0(x_0) - \beta_1(x_0)x_i]^2$$

row for x_0

diagonal NxN from K (weights)

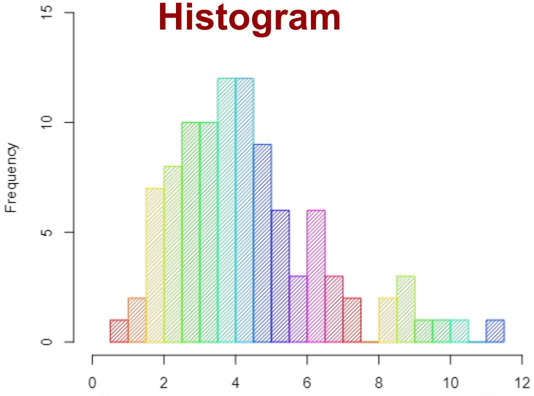


Full solution for local linear regression with normal equations for weighted least squares:

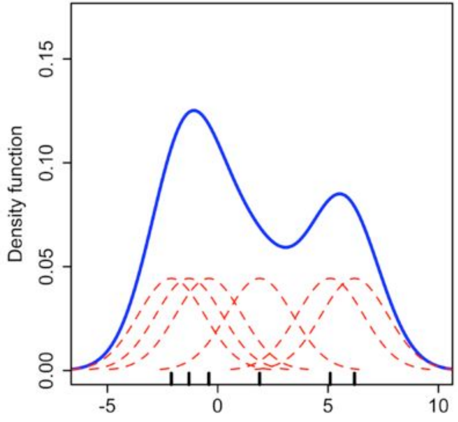
$$\hat{r}(x_0) = b(x_0)^T(\beta^T W(x_0)\beta)^{-1}\beta^T W(x_0)y$$

(Hastie et al., 2009)

Other Nonparametric Methods



Histogram



Kernel Density Estimation

```
#compute the bootstrap:

iters = 5000 #number of iterations
all_means = [] #will store the sample mean per iteration
#run the simulation loops:
for i in range(iters):
  resample = np.random.choice(sample, size=n, replace=True)
  resample_mean = resample.mean()
  all_means.append(resample_mean)

#sort the resampled means from least to greatest:
sorted_means = sorted(all_means)
#pick the upper and lower values for 95% CI:
lower = sorted_means[int(0.025*iters)]
upper = sorted_means[-int(0.025*iters)]
print("95 CI based on the bootstrap: [%.3f, %.3f]" % (lower, upper) )
```

```
95 CI based on the bootstrap: [19.239, 19.626]
```

The Bootstrap