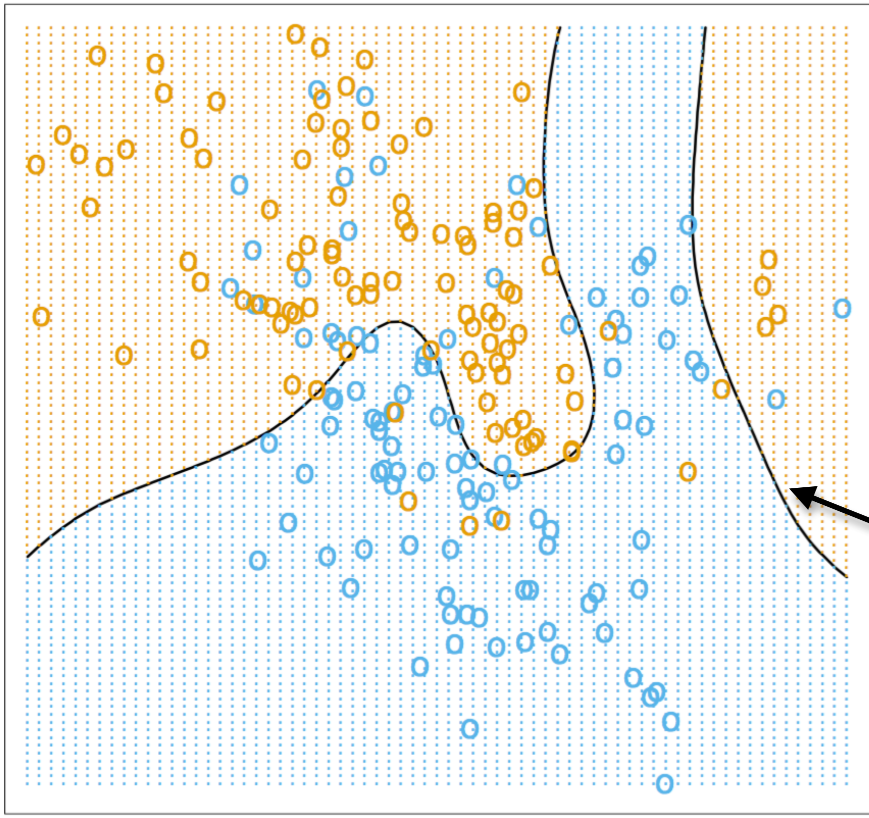
Nearest Neighbor

Some data, Bayes Classifier



Training data:

True label: +1

True label: -1

Optimal "Bayes" classifier:

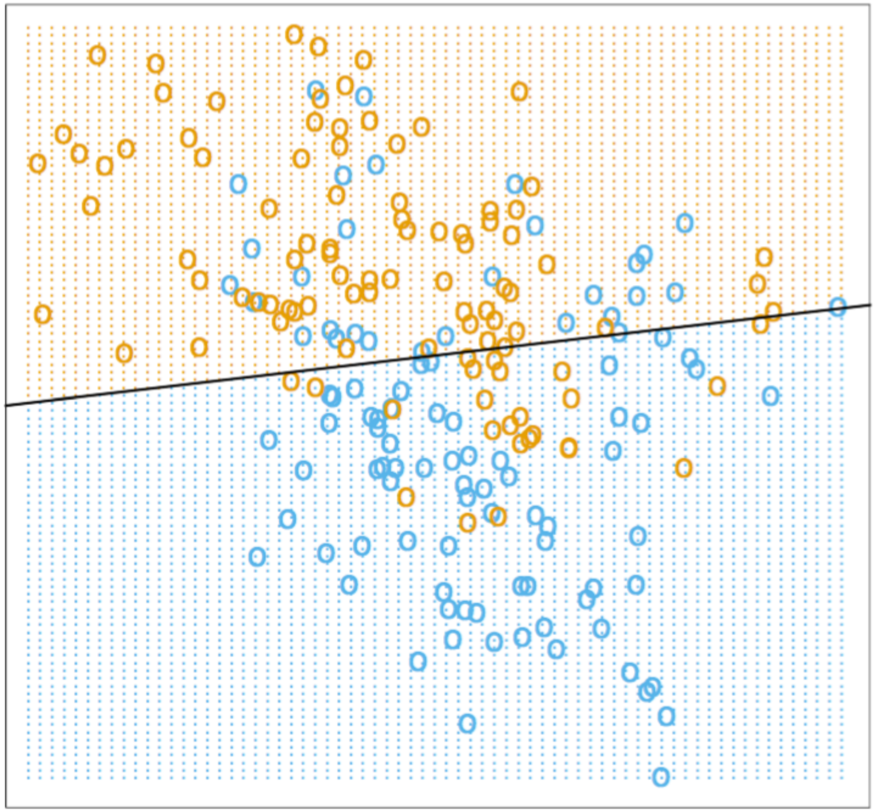
$$\mathbb{P}(Y = 1 | X = x) = \frac{1}{2}$$

Predicted label: +1

Predicted label: -1

Figures stolen from Hastie et al

Linear Decision Boundary



Training data:

- True label: +1
- True label: -1

Learned:

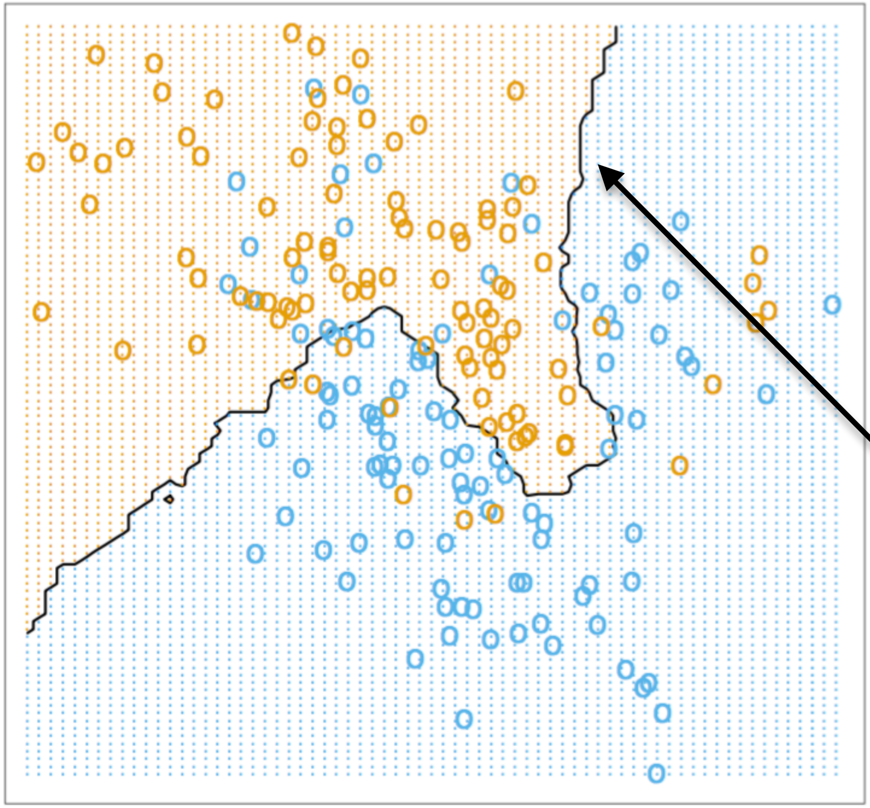
Linear Decision boundary

$$x^T w + b = 0$$

- Predicted label: +1
- Predicted label: -1

Figures stolen from Hastie et al

15 Nearest Neighbor Boundary



Training data:

True label: +1

True label: -1

Learned:

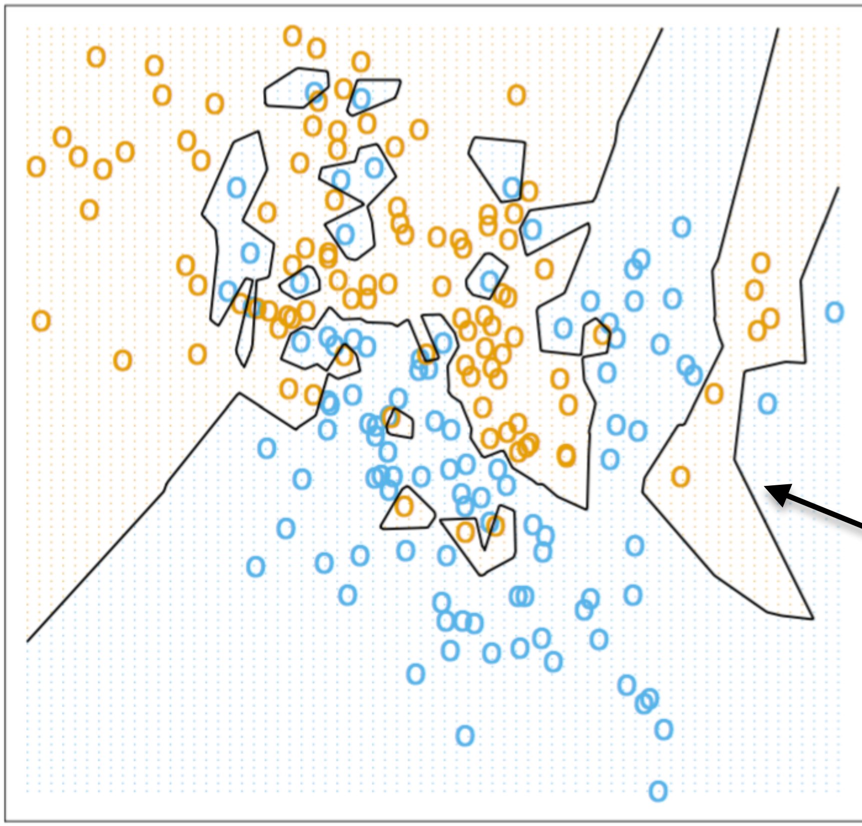
15 nearest neighbor decision boundary (majority vote)

Predicted label: +1

Predicted label: -1

Figures stolen from Hastie et al

1 Nearest Neighbor Boundary



Training data:

True label: +1

True label: -1

Learned:

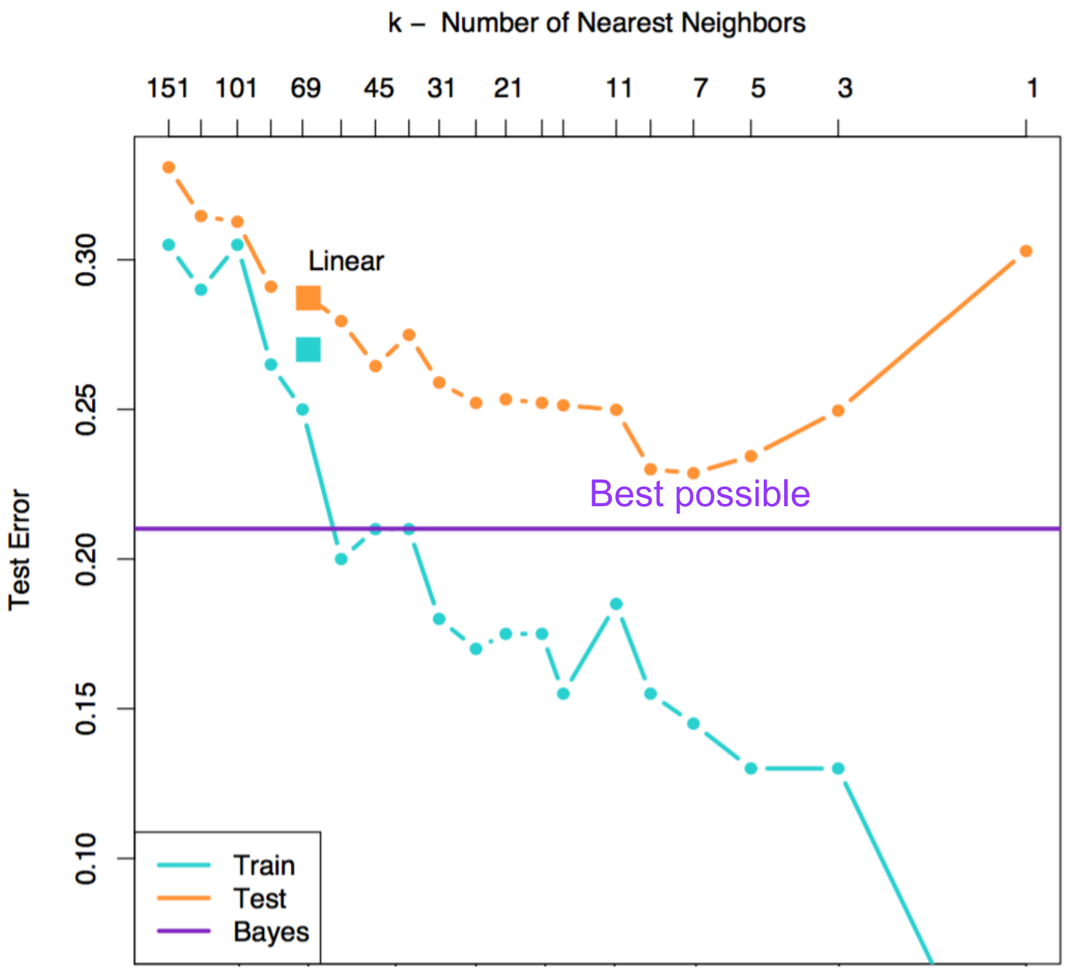
1 nearest neighbor decision boundary (majority vote)

Predicted label: +1

Predicted label: -1

Figures stolen from Hastie et al

k-Nearest Neighbor Error



Bias-Variance tradeoff

As k->infinity?

Bias:

Variance:

As k->1?

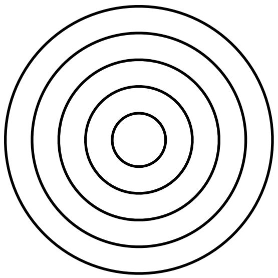
Bias:

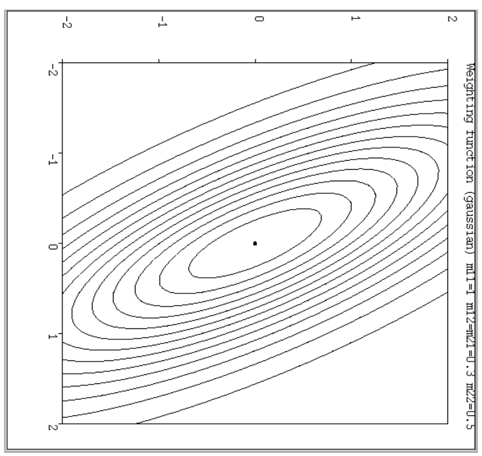
Variance:

Figures stolen from Hastie et al

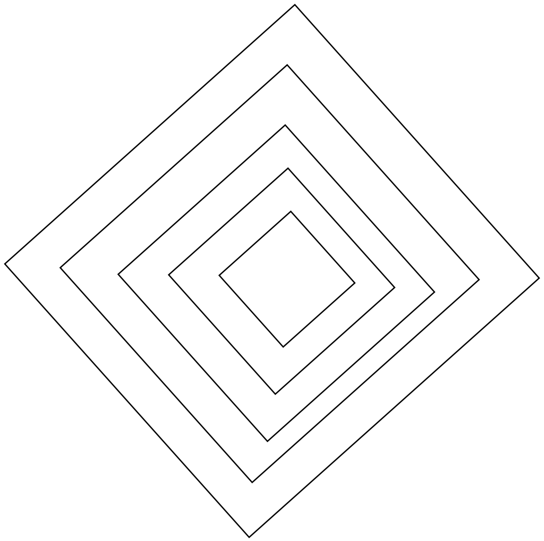
Notable distance metrics (and their level sets)

L_2 norm

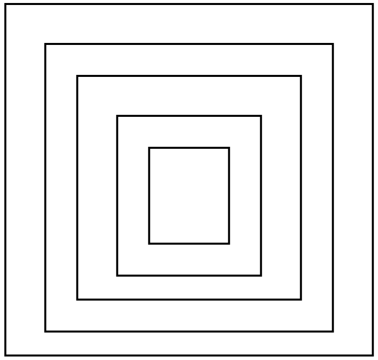




Mahalanobis



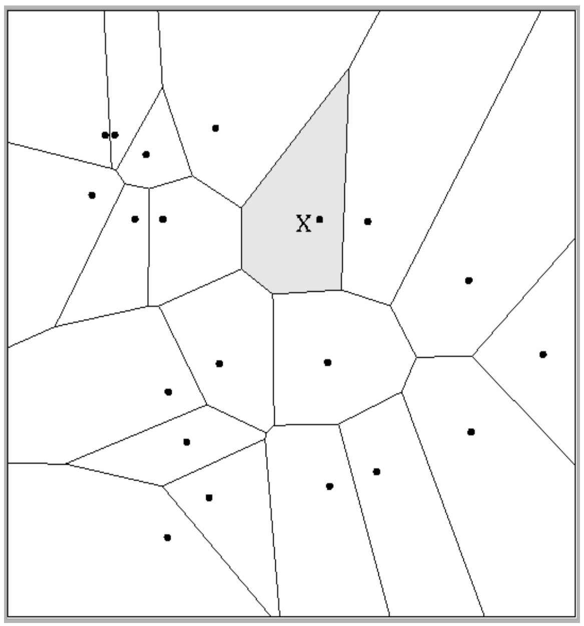
L_1 norm (taxi-cab)



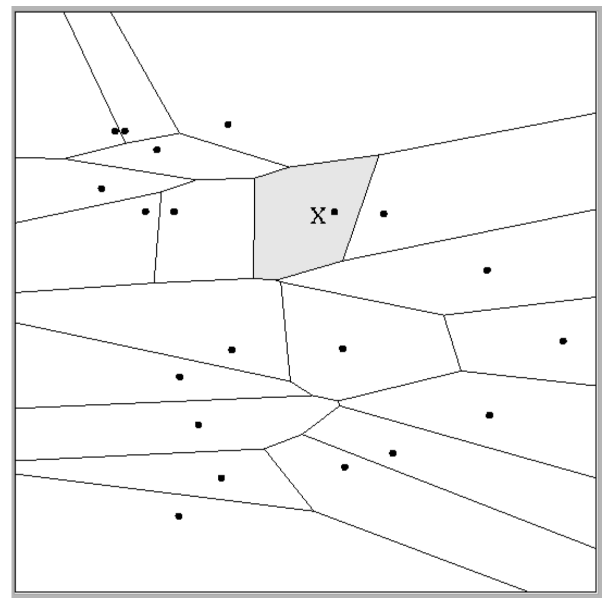
L-infinity *(max) norm*

1 nearest neighbor

One can draw the nearest-neighbor regions in input space.



$Dist(\mathbf{x}^i, \mathbf{x}^j) = (x^i_1 - x^j_1)^2 + (x^i_2 - x^j_2)^2$



$Dist(\mathbf{x}^i, \mathbf{x}^j) = (x^i_1 - x^j_1)^2 + (3x^i_2 - 3x^j_2)^2$

The relative scalings in the distance metric affect region shapes

1 nearest neighbor guarantee - classification

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d, \quad y_i \in \{0,1\} \qquad (x_i, y_i) \overset{iid}{\sim} P_{XY}$$

Theorem[Cover, Hart, 1967] If P_X is supported everywhere in $\mathbb{R}^d$ and $P(Y = 1|X = x)$ is smooth everywhere, then as $n \to \infty$ the 1-NN classification rule has error at most twice the Bayes error rate.

1 nearest neighbor guarantee - classification

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d, \quad y_i \in \{0,1\} \qquad (x_i, y_i) \overset{iid}{\sim} P_{XY}$$

Theorem[Cover, Hart, 1967] If P_X is supported everywhere in $\mathbb{R}^d$ and $P(Y = 1|X = x)$ is smooth everywhere, then as $n \to \infty$ the 1-NN classification rule has error at most twice the Bayes error rate.

As $x_a \to x_b$ we have $\mathbb{P}(Y_a = 1|X_a = x_a) \to \mathbb{P}(Y_b = 1|X_b = x_b)$

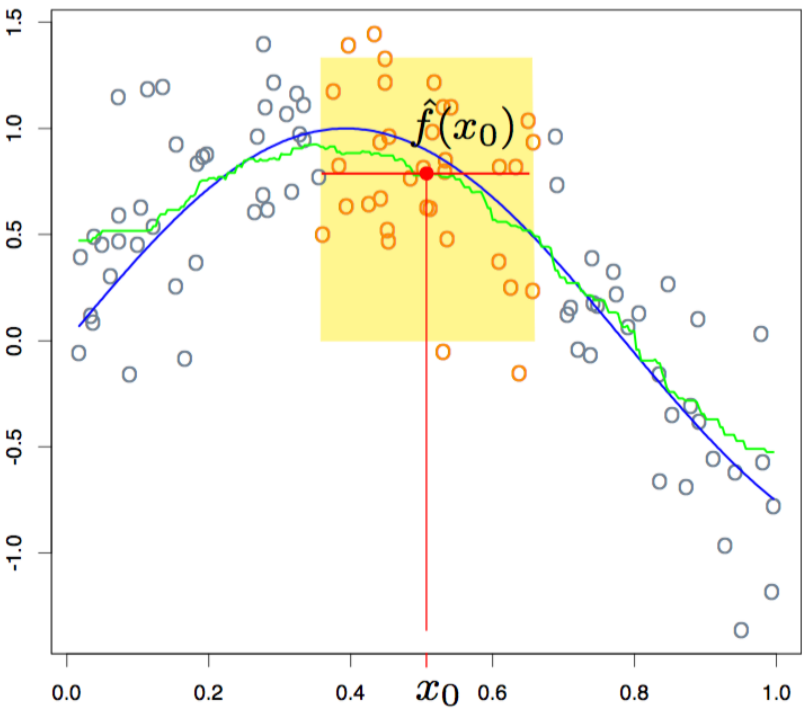
If $p_* = \max_{y=0,1} \mathbb{P}(Y_b = y|X_b = x_b)$ then the Bayes Error $= 1 - p_*$

1-nearest neighbor error $=$

$$\lim_{n\to\infty} \mathbb{P}(\widehat{f}_{1NN}(x_a) \neq Y_a|X_a = x_a) = \mathbb{P}(Y_b \neq Y_a|X_a = x_b, X_b = x_b)$$
$$= \mathbb{P}(Y_b = 1|X_b = x_b)\mathbb{P}(Y_a = 0|X_a = x_b) + \mathbb{P}(Y_b = 0|X_b = x_b)\mathbb{P}(Y_a = 1|X_a = x_b)$$
$$= 2p_*(1 - p_*) \leq 2(1 - p_*)$$

Nearest neighbor regression

$$\{(x_i, y_i)\})_{i=1}^{n}$$

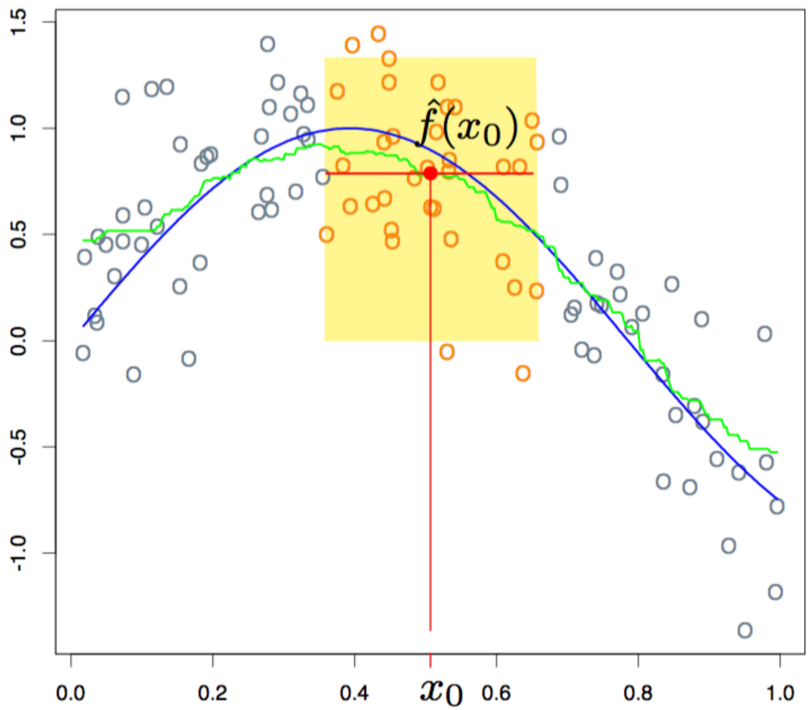


$$\mathcal{N}_k(x_0) = k\text{-nearest neighbors of } x_0$$

$$\widehat{f}(x_0) = \sum_{x_i \in \mathcal{N}_k(x_0)} \frac{1}{k}\, y_i$$

Nearest neighbor regression

$$\{(x_i, y_i)\}_{i=1}^n$$

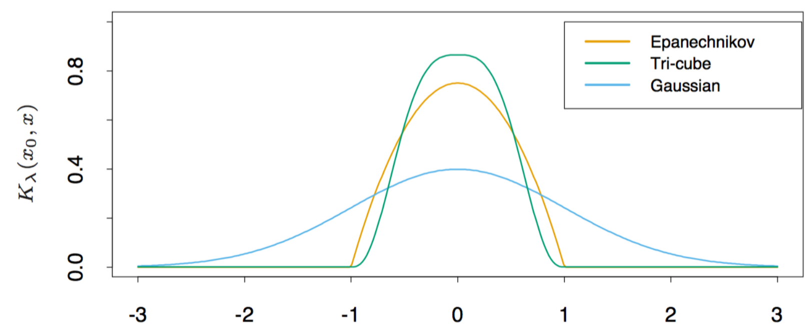


$\mathcal{N}_k(x_0) = k$-nearest neighbors of x_0

$$\widehat{f}(x_0) = \sum_{x_i \in \mathcal{N}_k(x_0)} \frac{1}{k} y_i$$

Why are far-away neighbors weighted same as close neighbors!

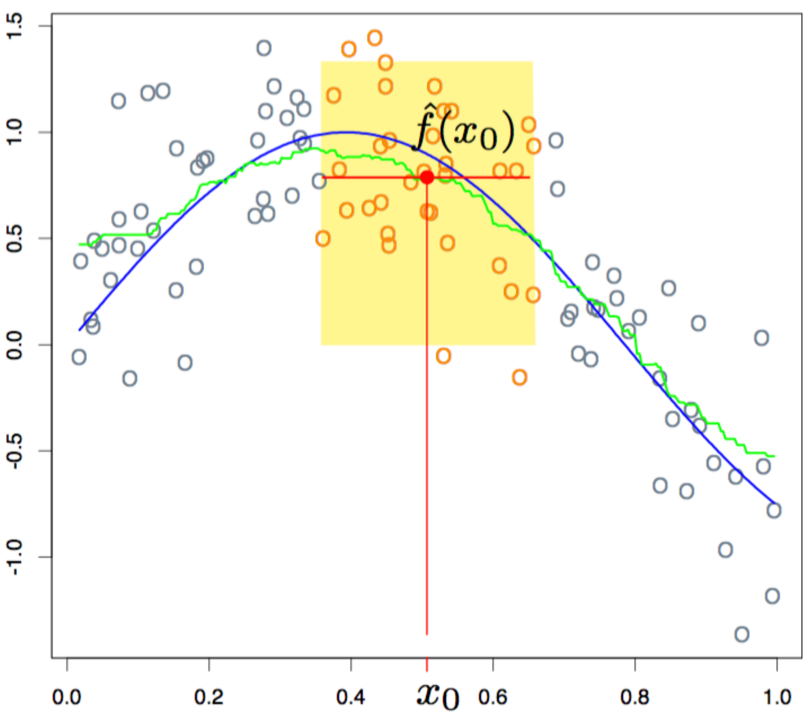
Kernel smoothing: $K(x, y)$

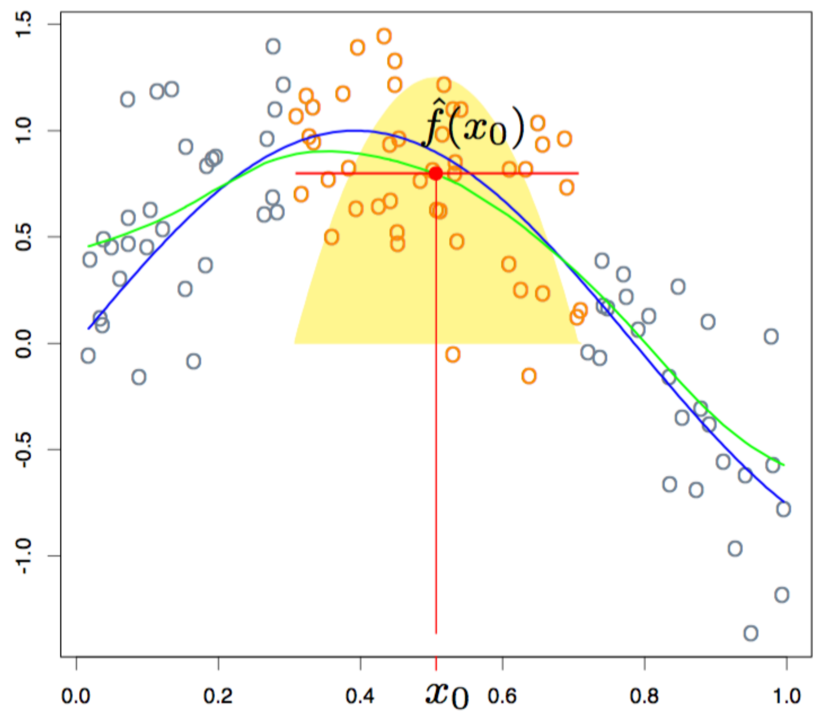


$$\widehat{f}(x_0) = \frac{\sum_{i=1}^n K(x_0, x_i) y_i}{\sum_{i=1}^n K(x_0, x_i)}$$

Nearest neighbor regression

$$\{(x_i, y_i)\}_{i=1}^n$$





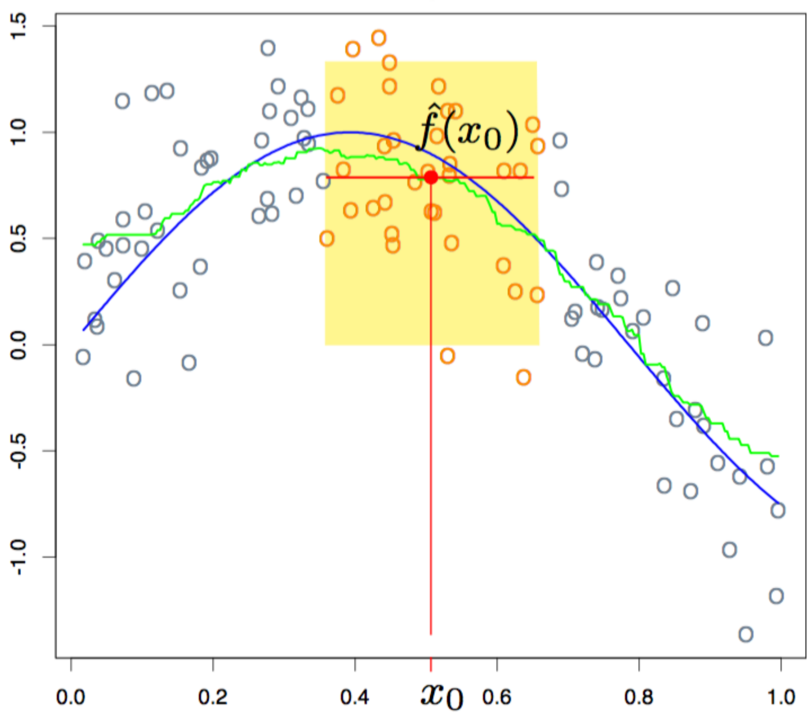
$\mathcal{N}_k(x_0) = k$-nearest neighbors of x_0

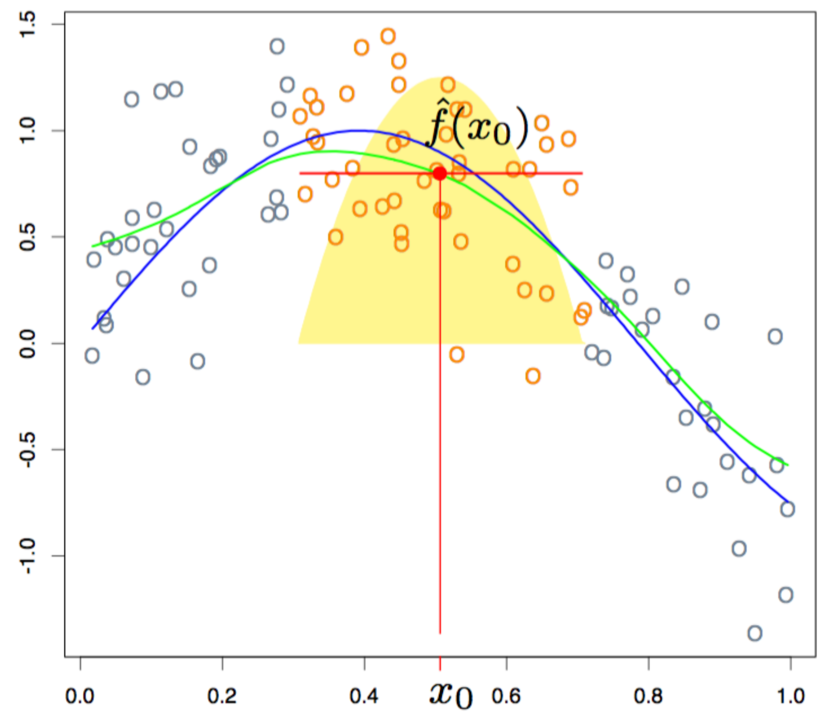
$$\widehat{f}(x_0) = \sum_{x_i \in \mathcal{N}_k(x_0)} \frac{1}{k}\, y_i$$

$$\widehat{f}(x_0) = \frac{\sum_{i=1}^n K(x_0, x_i) y_i}{\sum_{i=1}^n K(x_0, x_i)}$$

Nearest neighbor regression

$$\{(x_i, y_i)\}_{i=1}^n$$





$\mathcal{N}_k(x_0) = k$-nearest neighbors of x_0

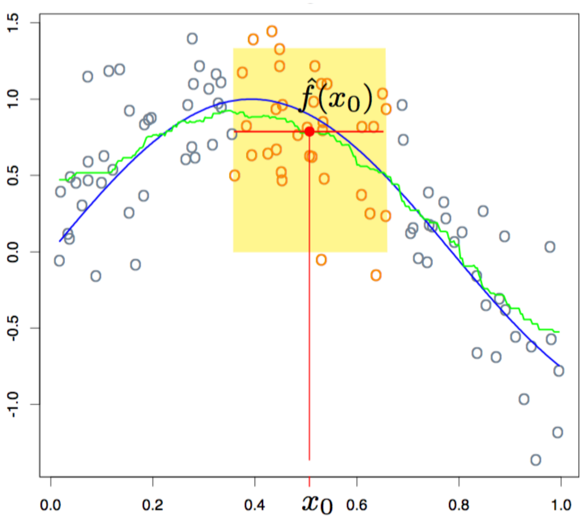
Why just average them?

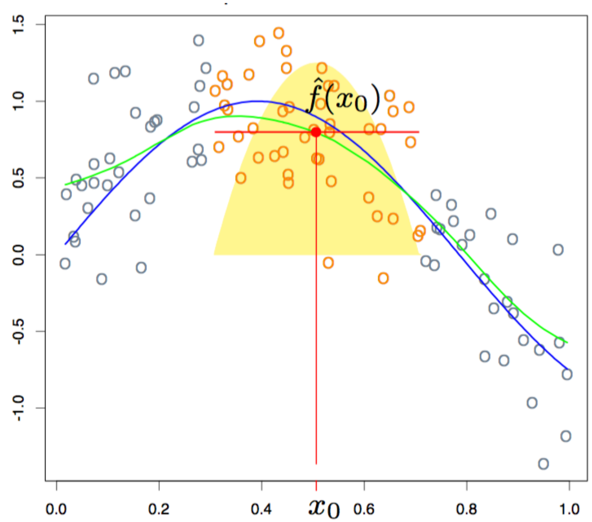
$$\widehat{f}(x_0) = \sum_{x_i \in \mathcal{N}_k(x_0)} \frac{1}{k} y_i$$

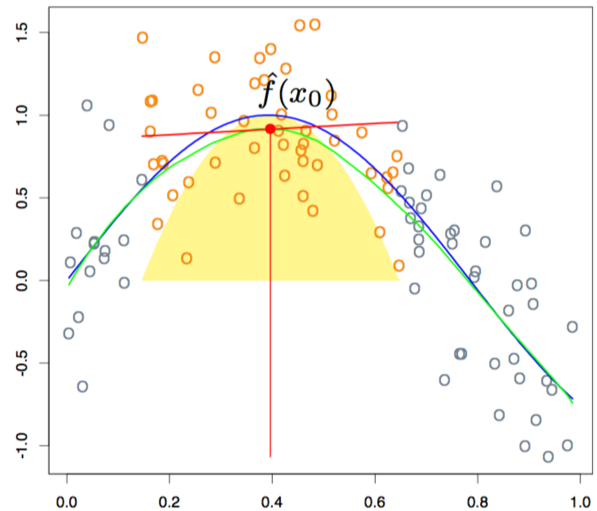
$$\widehat{f}(x_0) = \frac{\sum_{i=1}^n K(x_0, x_i) y_i}{\sum_{i=1}^n K(x_0, x_i)}$$

Nearest neighbor regression

$$\{(x_i, y_i)\}_{i=1}^n$$







$\mathcal{N}_k(x_0) = k\text{-nearest neighbors of } x_0$

$$\widehat{f}(x_0) = \sum_{x_i \in \mathcal{N}_k(x_0)} \frac{1}{k} \, y_i$$

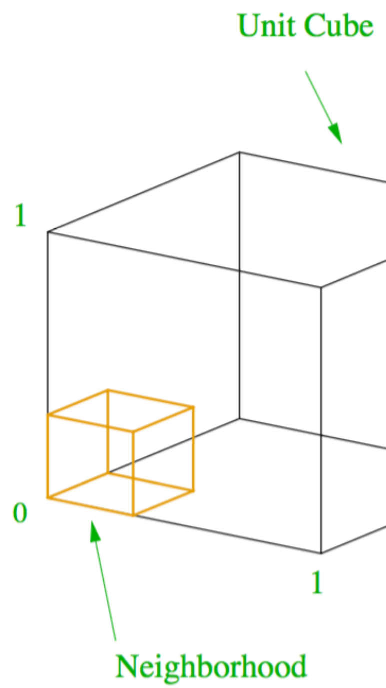
$$\widehat{f}(x_0) = \frac{\sum_{i=1}^n K(x_0, x_i) y_i}{\sum_{i=1}^n K(x_0, x_i)}$$

$$\widehat{f}(x_0) = b(x_0) + w(x_0)^T x_0$$

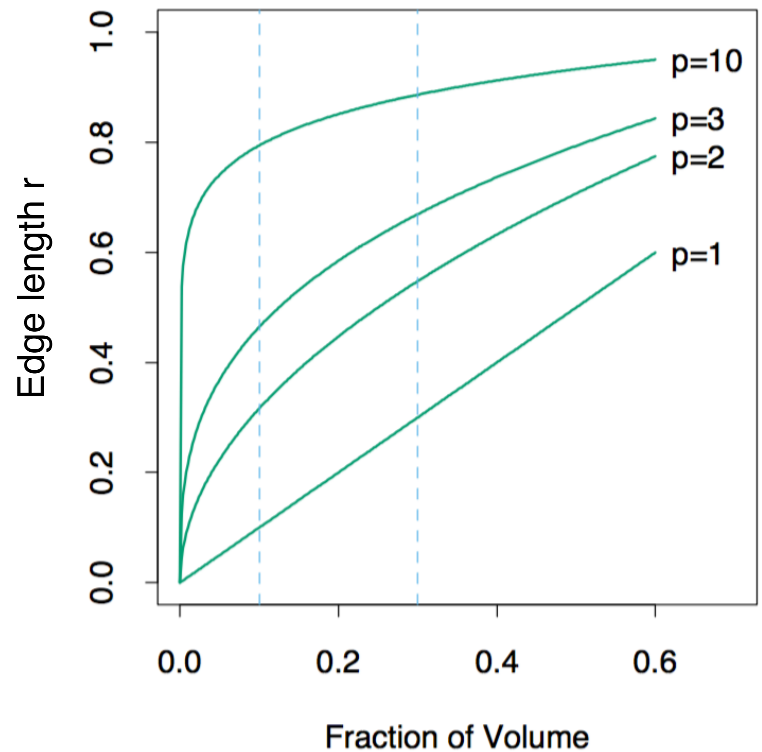
$$w(x_0), b(x_0) = \arg\min_{w,b} \sum_{i=1}^n K(x_0, x_i)(y_i - (b + w^T x_i))^2$$

Local Linear Regression

Curse of dimensionality Ex. 1



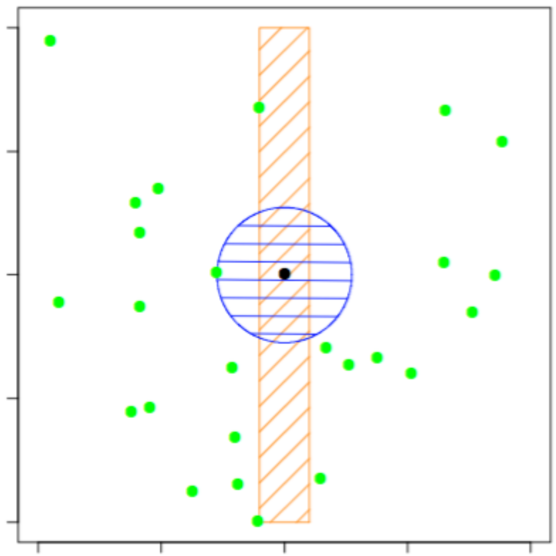
side length r

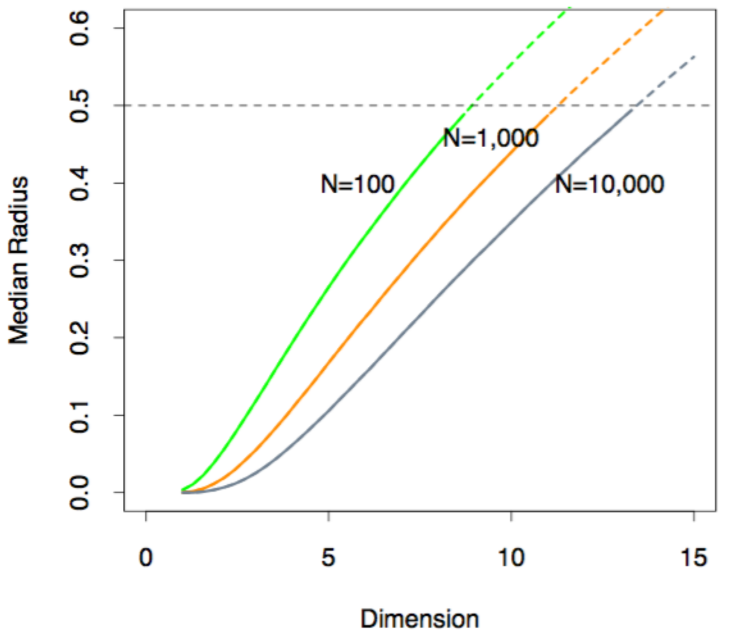


X is uniformly distributed over $[0,1]^p$. What is $\mathbb{P}(X \in [0,r]^p)$?

Curse of dimensionality Ex. 2

$\{X_i\}_{i=1}^n$ are uniformly distributed over $[-.5, .5]^p$.





What is the median distance from a point at origin to its 1NN?

Nearest Neighbor Overview

- Very simple to explain and implement
- No training! But finding nearest neighbors in large dataset at test can be computationally demanding (kD-trees help)
- You can use other forms of distance (not just Euclidean)
- Smoothing with Kernels and local linear regression can improve performance (at the cost of higher variance)
- With a lot of data, “local methods” have strong, simple theoretical guarantees.
- Without a lot of data, neighborhoods aren’t “local” and methods suffer.

Bootstrap

UNIVERSITY *of* WASHINGTON

Limitations of CV

- An 80/20 split throws out a relatively large amount of data if only have, say, 20 examples.
- Test error is informative, but how accurate is this number? (e.g., 3/5 heads vs. 30/50)
- How do I get confidence intervals on statistics like the median or variance of a distribution?
- Instead of the error for the entire dataset, what if I want to study the error for a particular example x?

The Bootstrap: Developed by Efron in 1979.

Bootstrap: basic idea

Given dataset drawn iid samples with CDF F_Z:

$$\mathcal{D} = \{z_1, \ldots, z_n\} \overset{i.i.d.}{\sim} F_Z$$

We compute a *statistic* of the data to get: $\widehat{\theta} = t(\mathcal{D})$

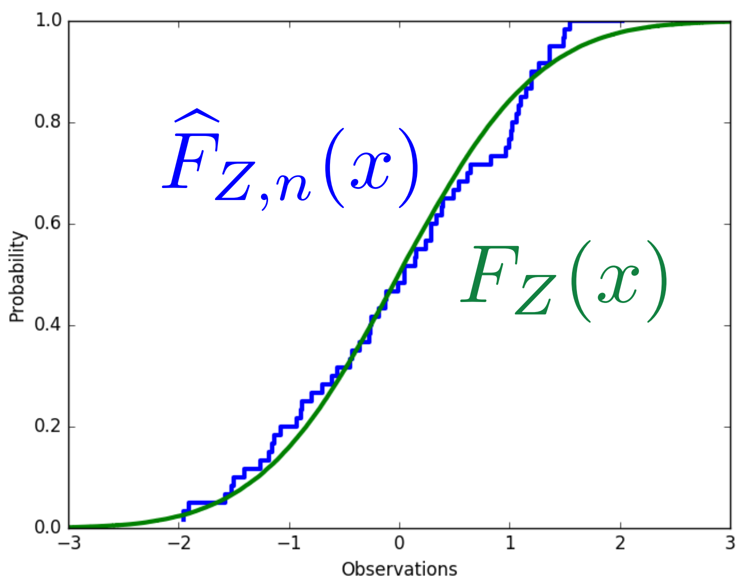
What is the distribution of $\widehat{\theta} = t(\mathcal{D})$?

Bootstrap: basic idea

Given dataset drawn iid samples with CDF F_Z:

$$\mathcal{D} = \{z_1, \ldots, z_n\} \overset{i.i.d.}{\sim} F_Z$$

We compute a *statistic* of the data to get: $\widehat{\theta} = t(\mathcal{D})$



$$F_Z(x) = \mathbb{P}(Z \le x)$$

$$\widehat{F}_{Z,n}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{z_i \le x\}$$

$$|\widehat{F}_{Z,n}(x) - F_Z(x)| \overset{n\to\infty}{\to} 0 \quad \text{a.s.}$$

Bootstrap: basic idea

Given dataset drawn iid samples with CDF F_Z:

$$\mathcal{D} = \{z_1, \ldots, z_n\} \stackrel{i.i.d.}{\sim} F_Z$$

We compute a *statistic* of the data to get: $\widehat{\theta} = t(\mathcal{D})$

For b=1,…,B define the *b*th ***bootstrapped*** dataset as drawing *n* samples **with replacement** from *D*

$$\mathcal{D}^{*b} = \{z_1^{*b}, \ldots, z_n^{*b}\} \stackrel{i.i.d.}{\sim} \widehat{F}_{Z,n}$$

and the *b*th bootstrapped statistic as: $\theta^{*b} = t(\mathcal{D}^{*b})$

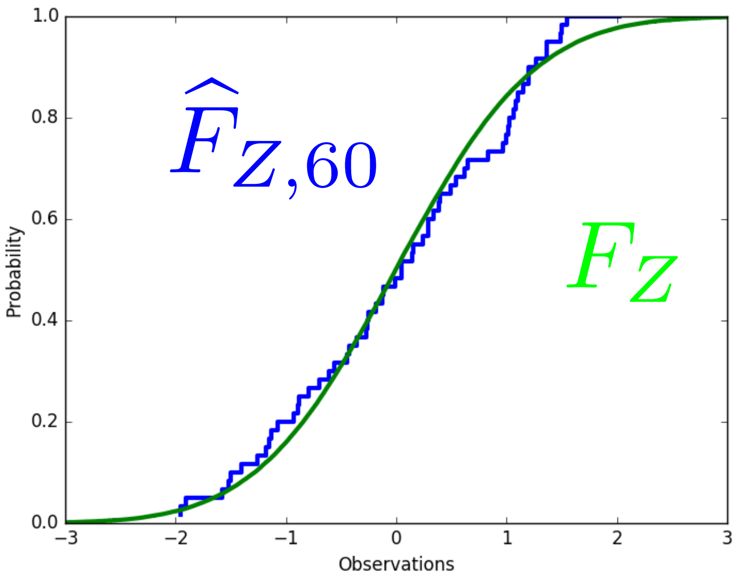
Bootstrap: basic idea

Given dataset drawn iid samples with CDF F_Z:

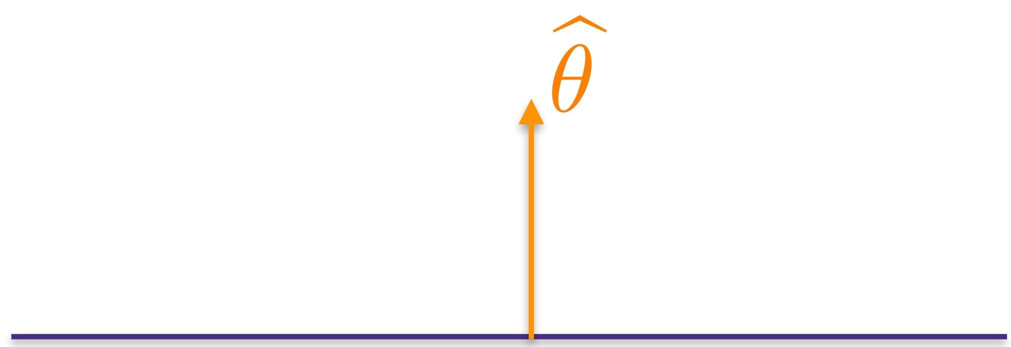
$$\mathcal{D} = \{z_1, \ldots, z_n\} \overset{i.i.d.}{\sim} F_Z$$

We compute a *statistic* of the data to get: $\widehat{\theta} = t(\mathcal{D})$

$$\mathcal{D}^{*b} = \{z_1^{*b}, \ldots, z_n^{*b}\} \overset{i.i.d.}{\sim} \widehat{F}_{Z,n} \qquad \theta^{*b} = t(\mathcal{D}^{*b})$$



$$|\widehat{F}_{Z,n}(x) - F_Z(x)| \overset{n\to\infty}{\to} 0 \quad \text{a.s.}$$

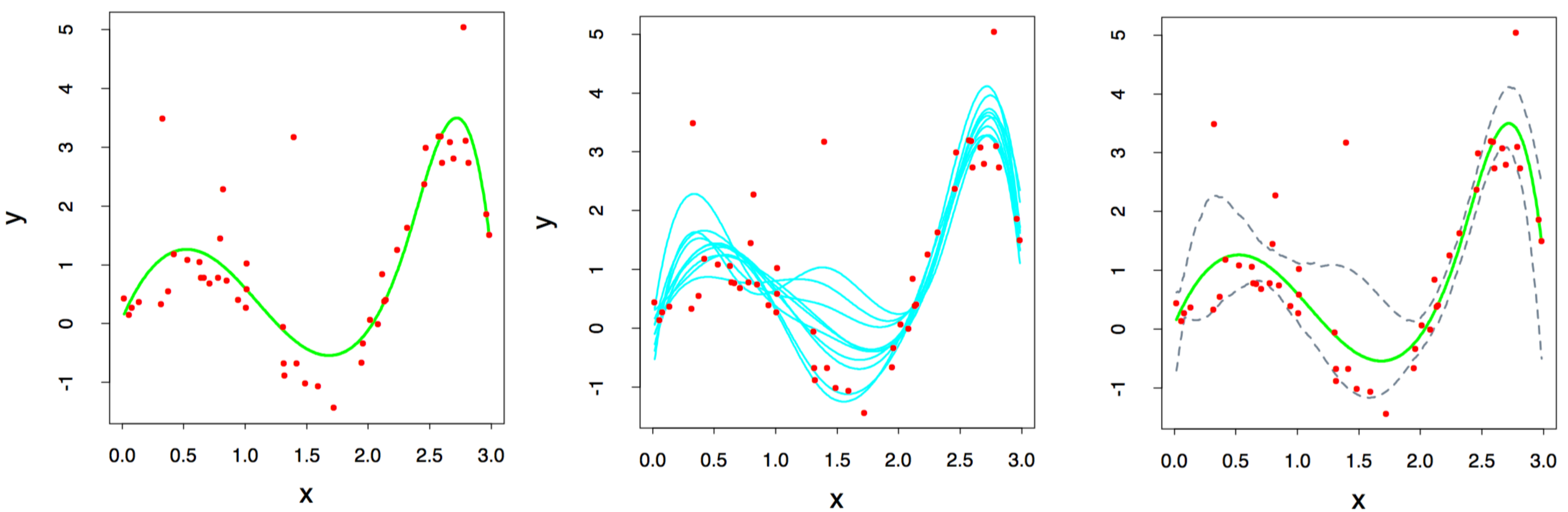


Applications

Common applications of the bootstrap:

- **Estimate parameters that escape simple analysis like the variance or median of an estimate**
- **Confidence intervals**
- **Estimates of error for a particular example:**

$\mathcal{D}$ $\widehat{\theta}$ θ^{*b} for $b = 1, \ldots, 10$ 95% confidence interval



Figures from Hastie et al

Takeaways

Advantages:

- **Bootstrap is very generally applicable. Build a confidence interval around *anything***
- **Very simple to use**
- **Appears to give meaningful results even when the amount of data is very small**
- **Very strong asymptotic theory (as num. examples goes to infinity)**

Takeaways

Advantages:

- **Bootstrap is very generally applicable. Build a confidence interval around *anything***
- **Very simple to use**
- **Appears to give meaningful results even when the amount of data is very small**
- **Very strong asymptotic theory (as num. examples goes to infinity)**

Disadvantages

- **Very few meaningful finite-sample guarantees**
- **Potentially computationally intensive**
- **Reliability relies on test statistic and rate of convergence of empirical CDF to true CDF, which is unknown**
- **Poor performance on "extreme statistics" (e.g., the max)**

Not perfect, but better than nothing.