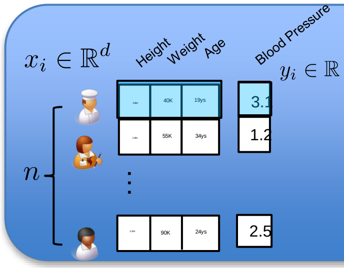
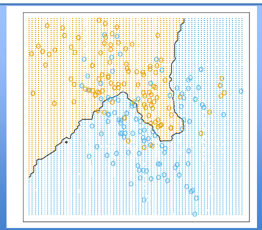
EECE5698
Parallel Processing for Data Analytics

Lecture 9: Regression and Statistical Learning

Road Map: What have We Learned So far?

Coming up:
Statistics &
Machine Learning

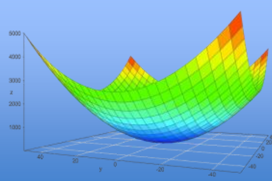


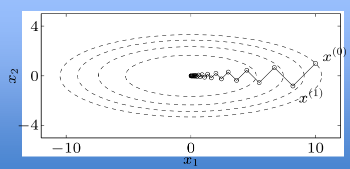


Regression, Classification
Regularization, Cross Validation
…

Convex Optimization

$$\arg\min_{x \in \mathbb{R}^d} f(x)$$





Descent Methods
Gradient Descent
Newton Method
…

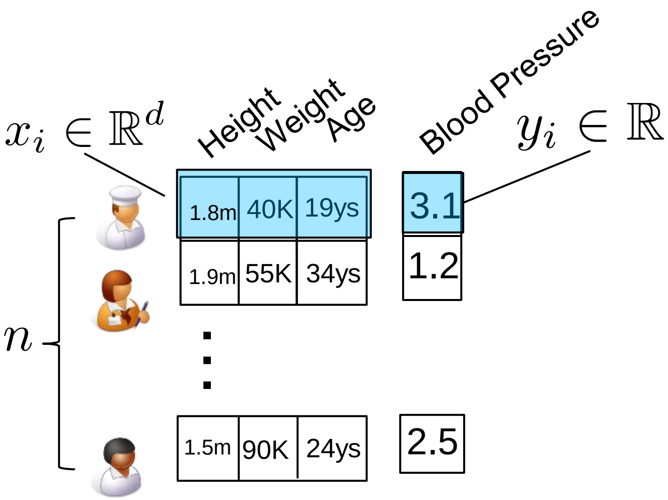
Parallel Processing

Apache Spark + python™

`map reduce reduceByKey join …`

Northeastern

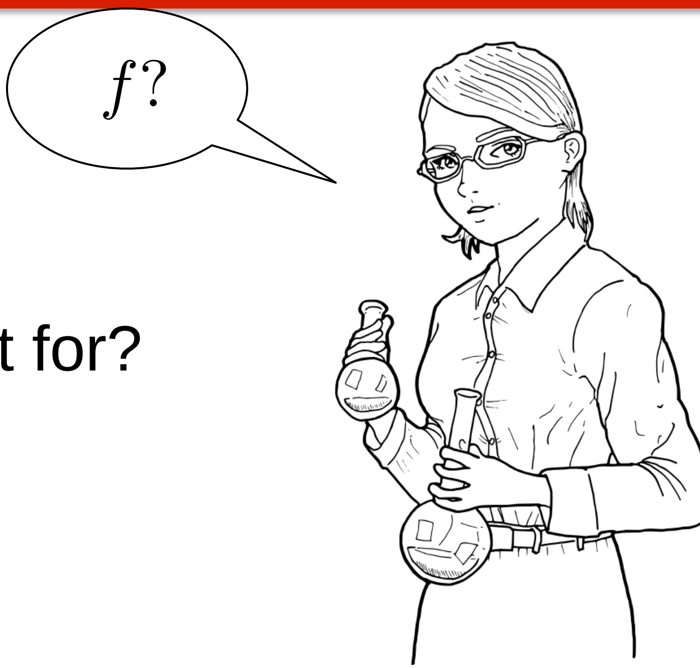
Regression



$$y_i \approx f(x_i), \quad i = 1, \ldots, n$$

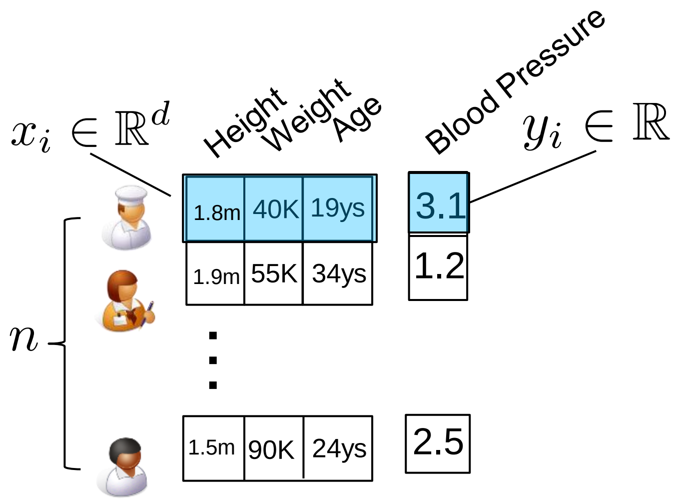
for some $f : \mathbb{R}^d \to \mathbb{R}$

What for?



- Prediction: If $x =$ | 1.2m | 70K | 20ys | then $y = ?$
- Correlation: If Weight ↑ then y ↑

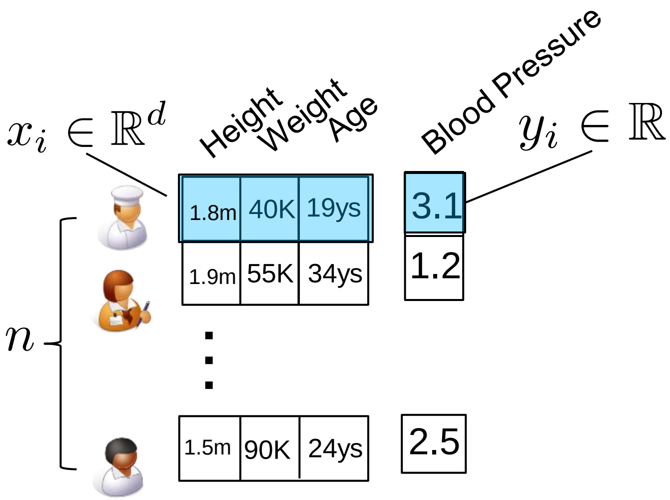
Regression: Terminology



$$y_i \approx f(x_i), \quad i = 1, \ldots, n$$

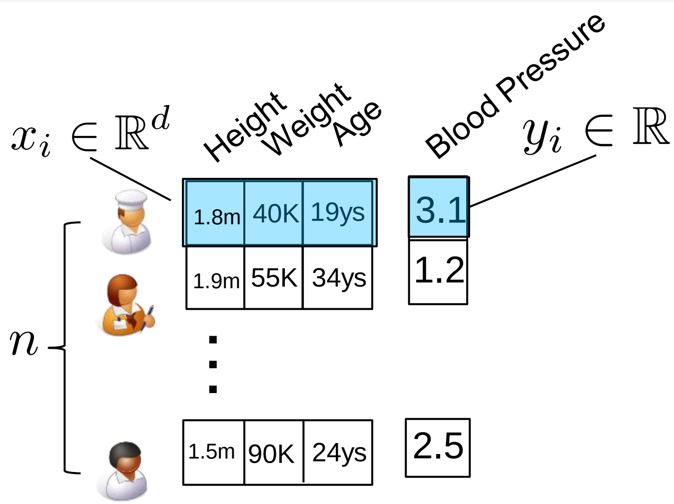
- $x_i \in \mathbb{R}^d$: features, independent variables, covariates, inputs,…
- $y_i \in \mathbb{R}$: label, dependent variable, outcome, response, output,…

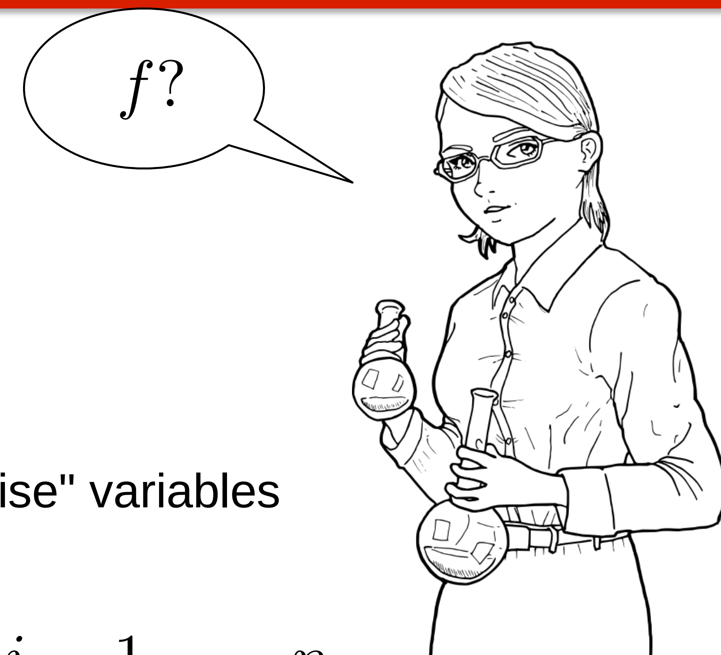
Regression: Noiseless Setting



$$y_i = f(x_i), \qquad i = 1, \ldots, n$$

Regression: Noisy setting





random "noise" variables

$$y_i = f(x_i) + \varepsilon_i, \qquad i = 1, \dots, n.$$

where ε_i are **independent and identically distributed** (i.i.d), and

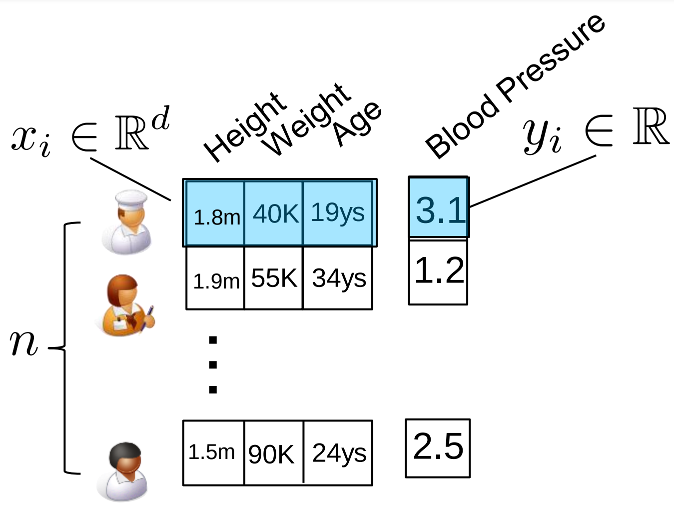
$$\mathbb{E}[\varepsilon_i] = 0 \qquad \mathbb{E}[\varepsilon_i^2] = \sigma^2 < \infty$$

Note: This implies that $y_i, i = 1, \dots, n,$ are **independent** random variables, where

$$\mathbb{E}[y_i] = f(x_i) \qquad \mathtt{Var}[y_i] = \mathbb{E}\left[(y_i - \mathbb{E}[y_i])^2\right] = \sigma^2$$

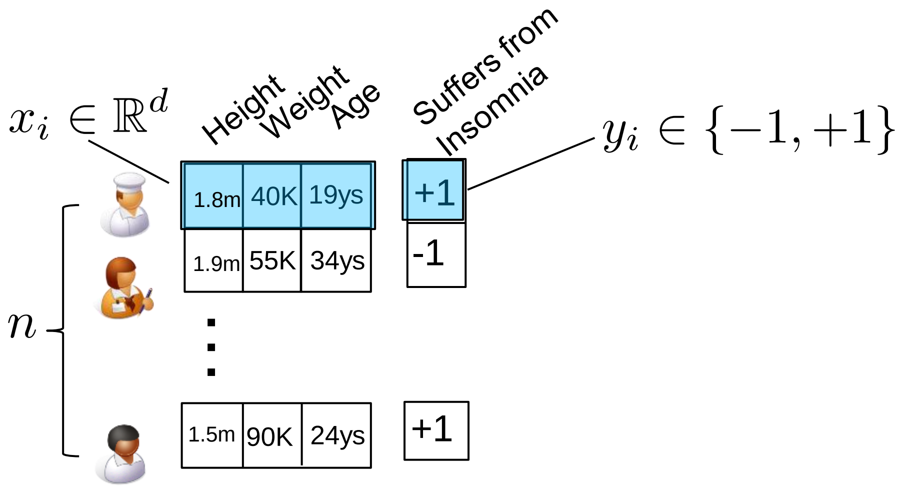
Northeastern

Regression vs. Classification



- ❑ Standard regression: $y_i \in \mathbb{R}$

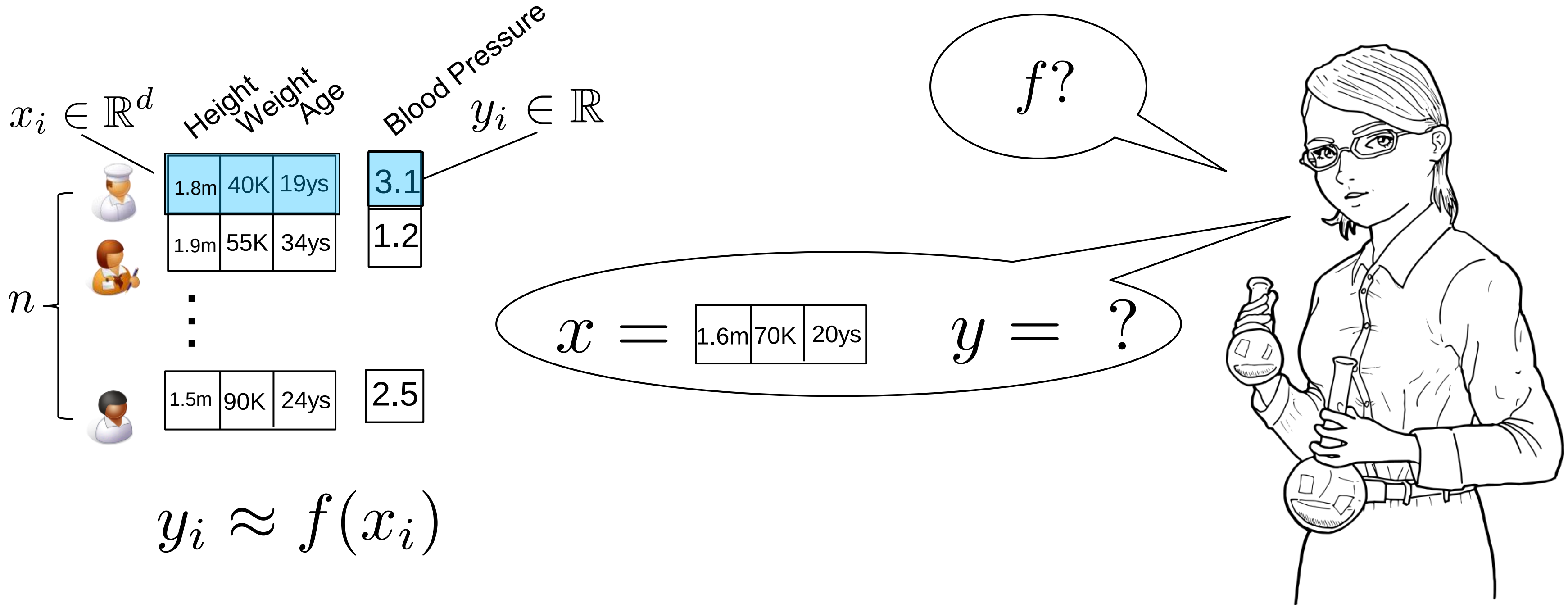
Regression vs. Classification



- Standard regression: $y_i \in \mathbb{R}$
- Classification: y_i are **discrete/categorical**, e.g.:
  - $y_i \in \{-1, +1\}$ (binary)
  - $y_i \in \{\texttt{red}, \texttt{blue}, \texttt{green}\}$

Northeastern

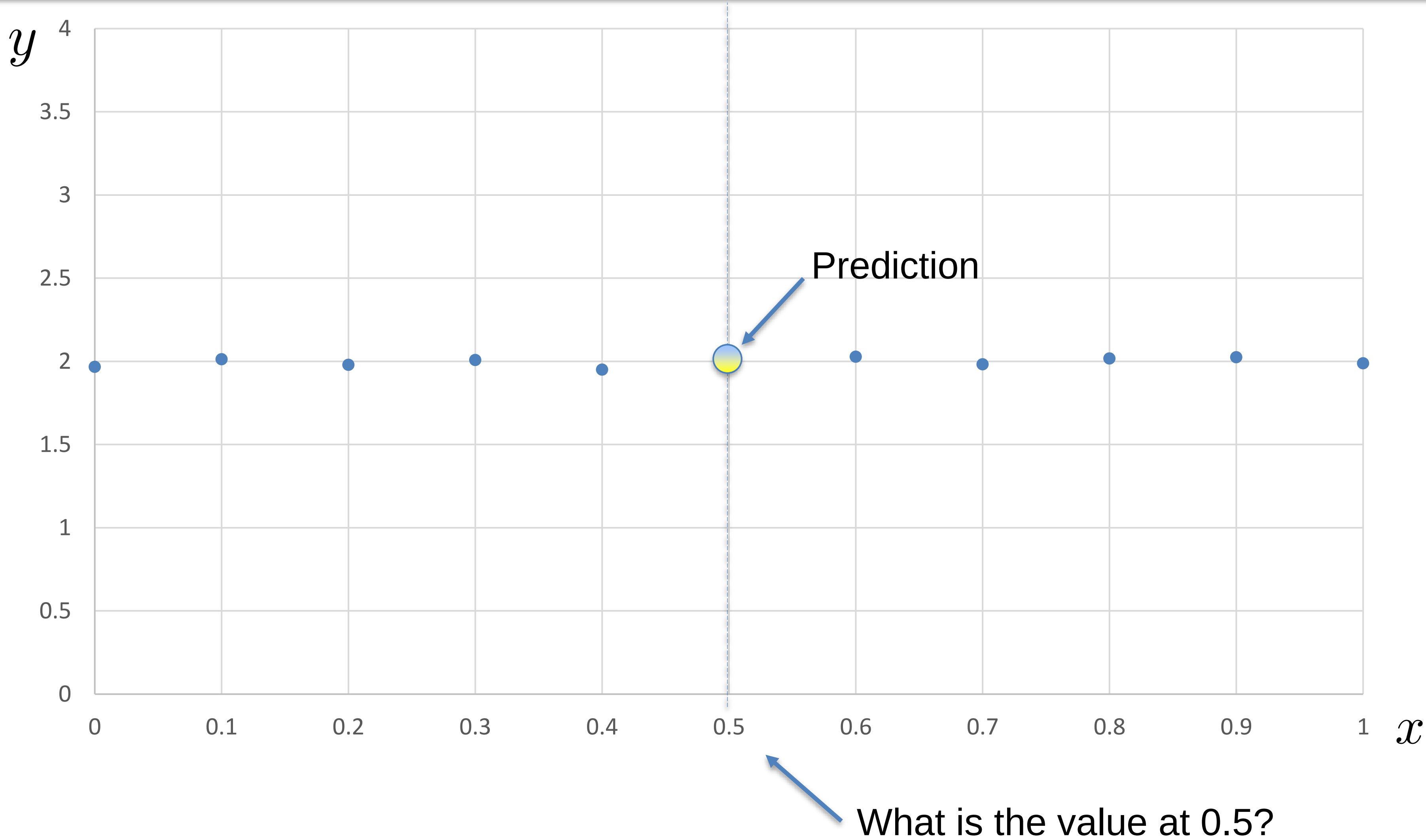
How would you solve this problem?



$$y_i \approx f(x_i)$$

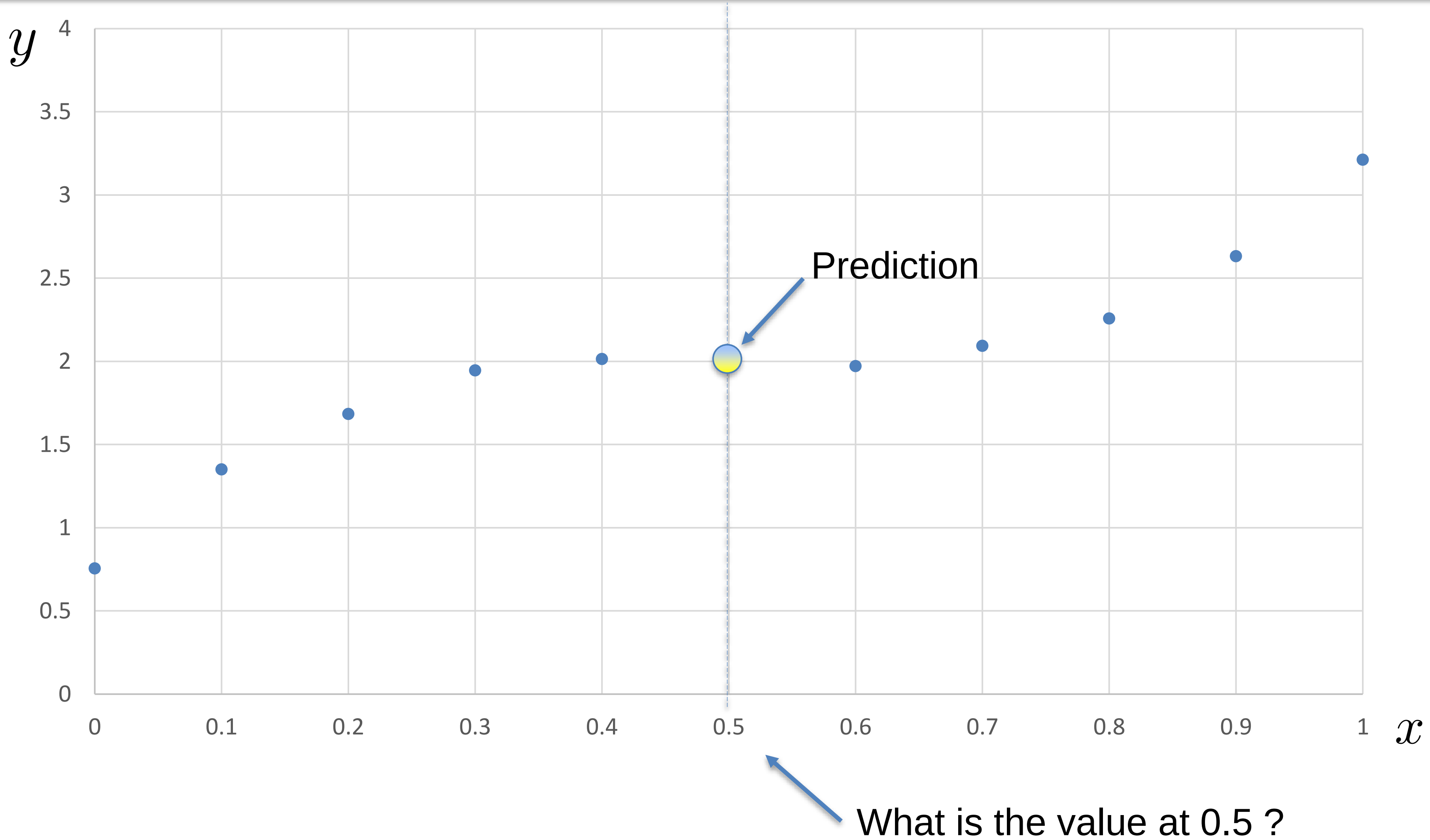
…you need to start making some assumptions on f!

Example 1



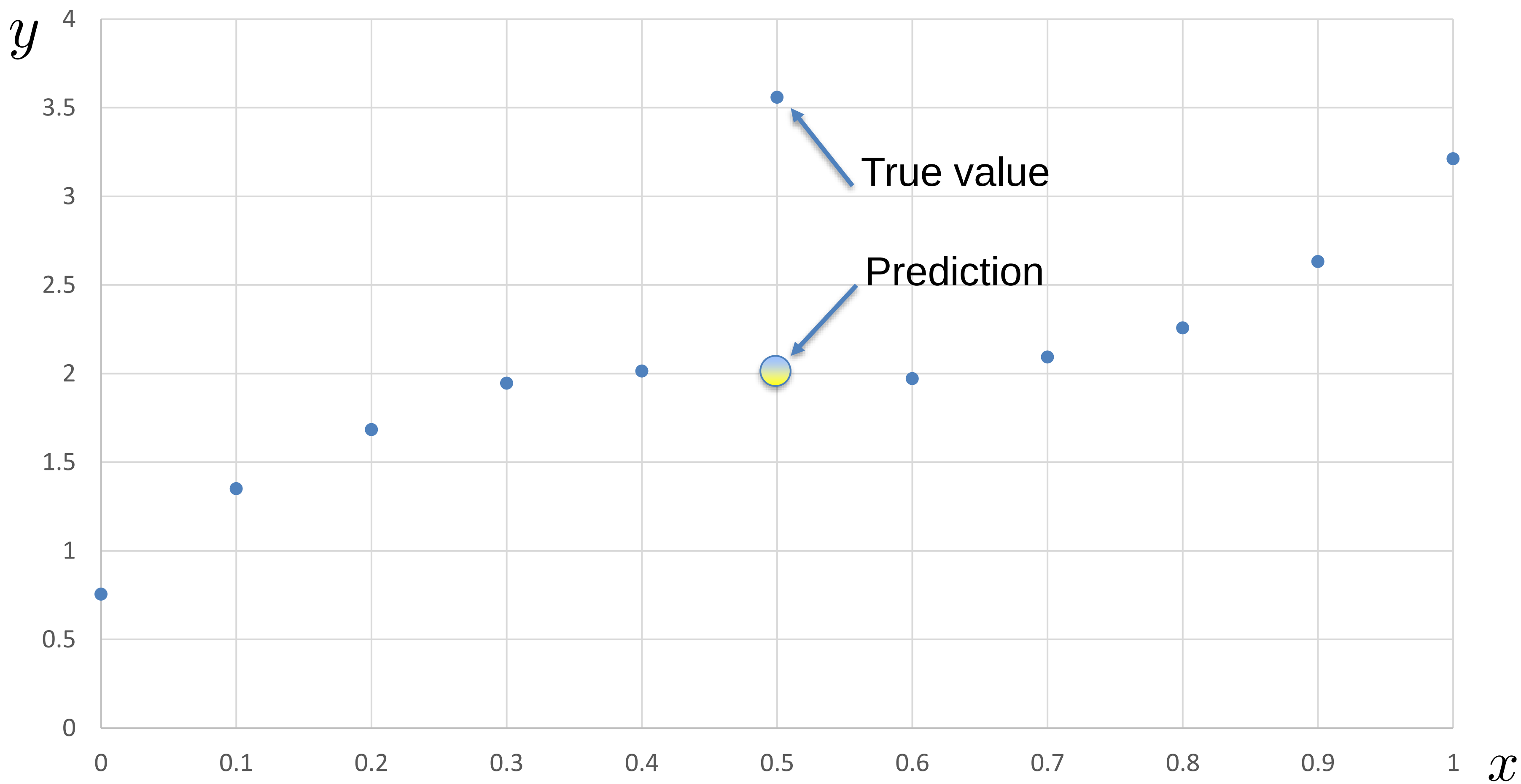
y
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x
Prediction
What is the value at 0.5?

Example 2



y
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x
Prediction
What is the value at 0.5 ?

Example 2



y
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x
True value
Prediction

Assumption: Continuity!

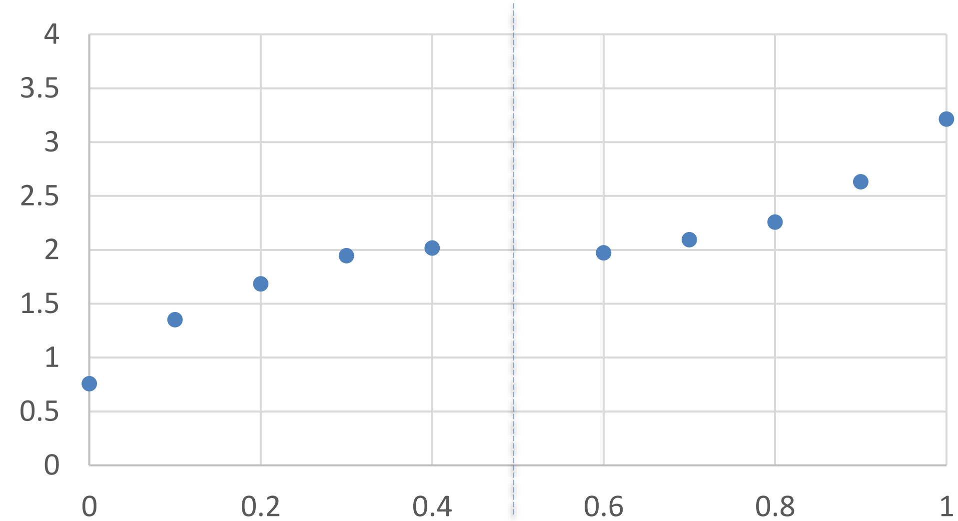
- **Assumption**: Function $f : \mathbb{R}^d \to \mathbb{R}$ is **continuous**

  If $\lim_{k\to\infty} x_k = x$ then $\lim_{k\to\infty} f(x_k) = f(x)$

- Values of f at points near x tell you something about $f(x)$!

K-Nearest Neighbor

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$

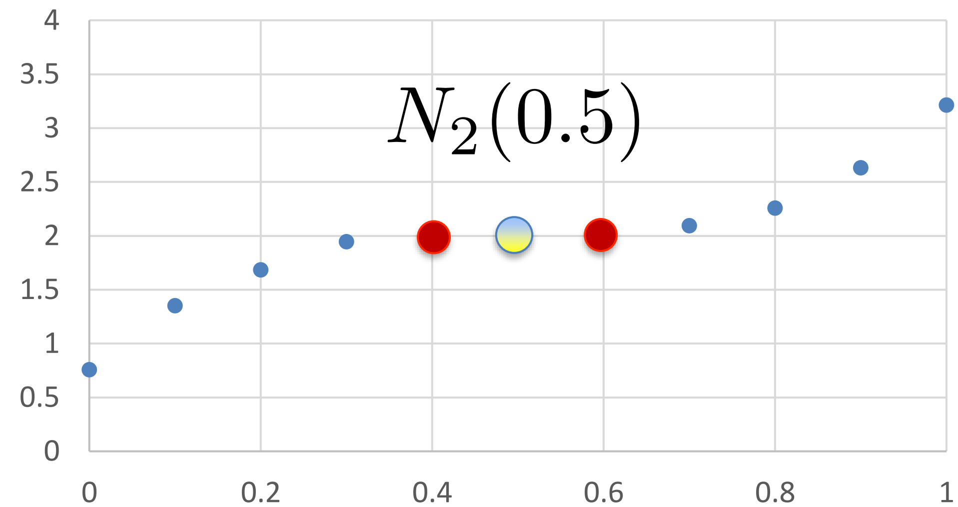


where $N_k(x)$ is the set of the k nearest neighbors of x

Northeastern

K-Nearest Neighbor

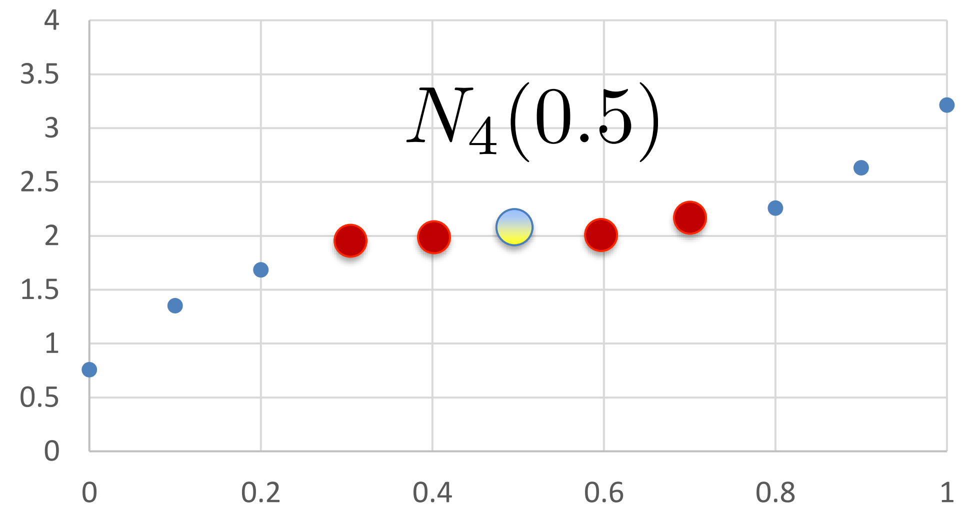
$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$



where $N_k(x)$ is the set of the k nearest neighbors of x

K-Nearest Neighbor

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$



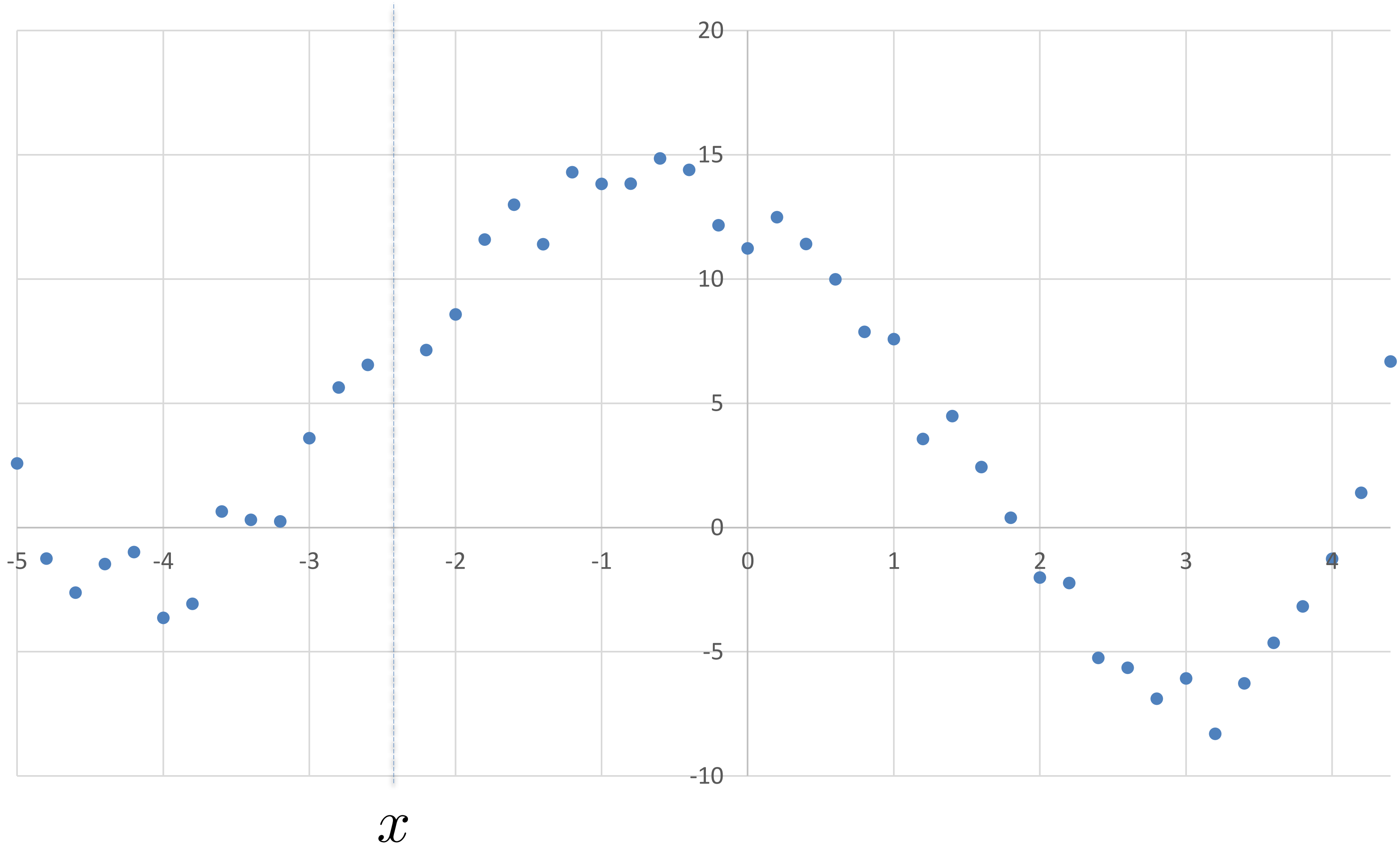
where $N_k(x)$ is the set of the k nearest neighbors of x

- ❑ $|N_k(x)| = k$
- ❑ For all $i \in N_k(x)$ and $j \notin N_k(x)$

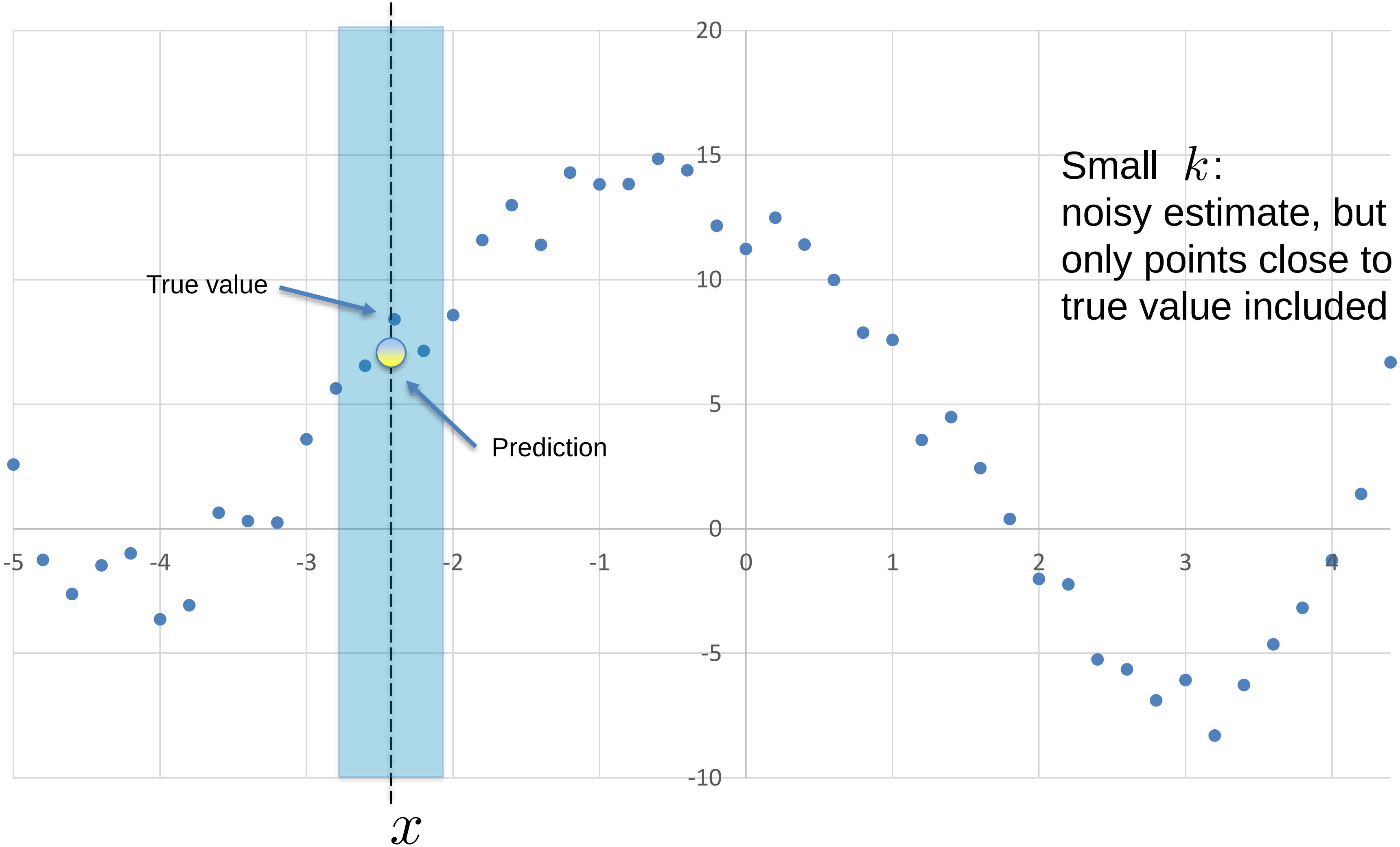
$$\|x - x_i\| \leq \|x - x_j\|$$

Northeastern

How big should k be?



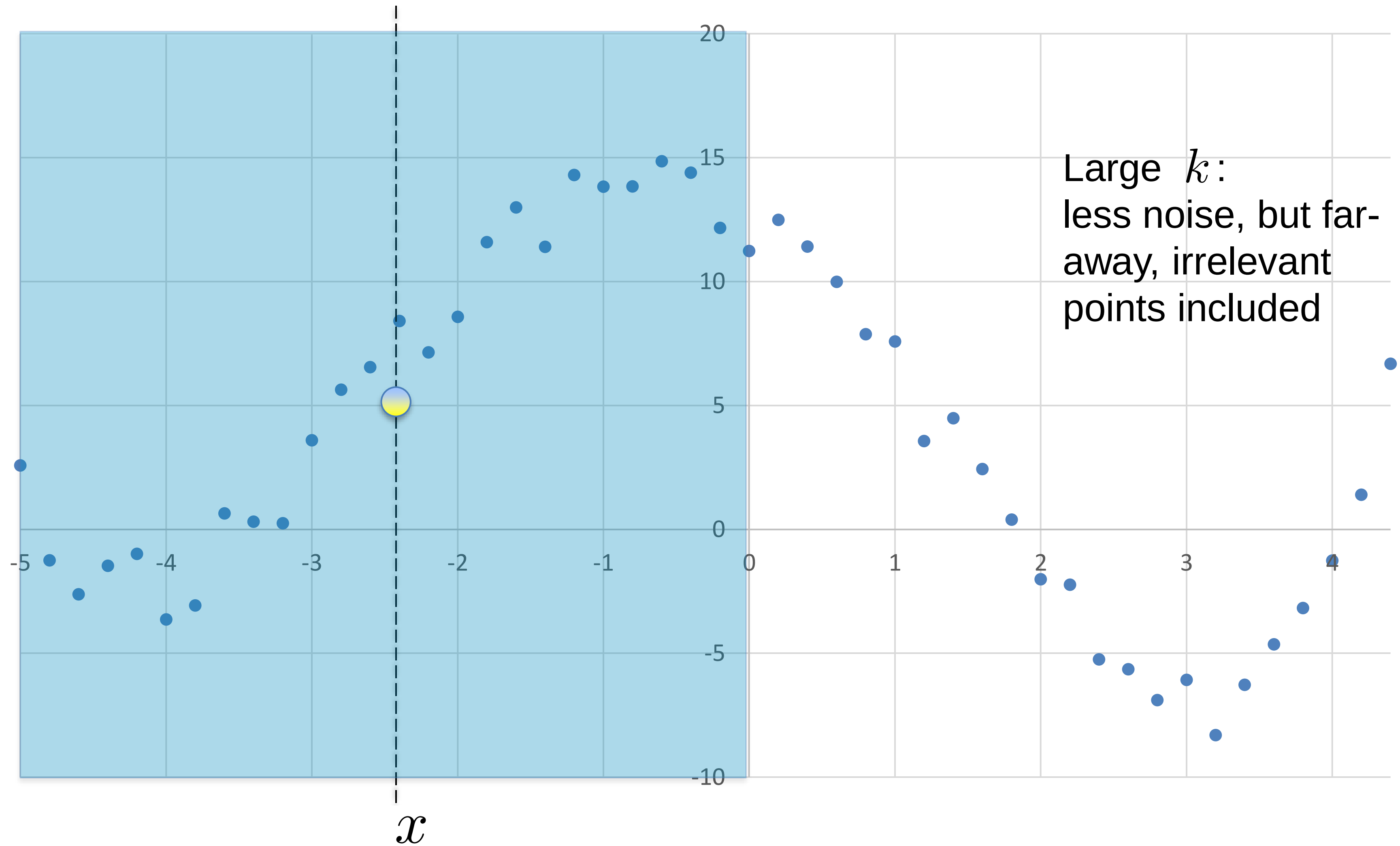
How big should k be?



Small k:
noisy estimate, but only points close to true value included

Northeastern

How big should k be?



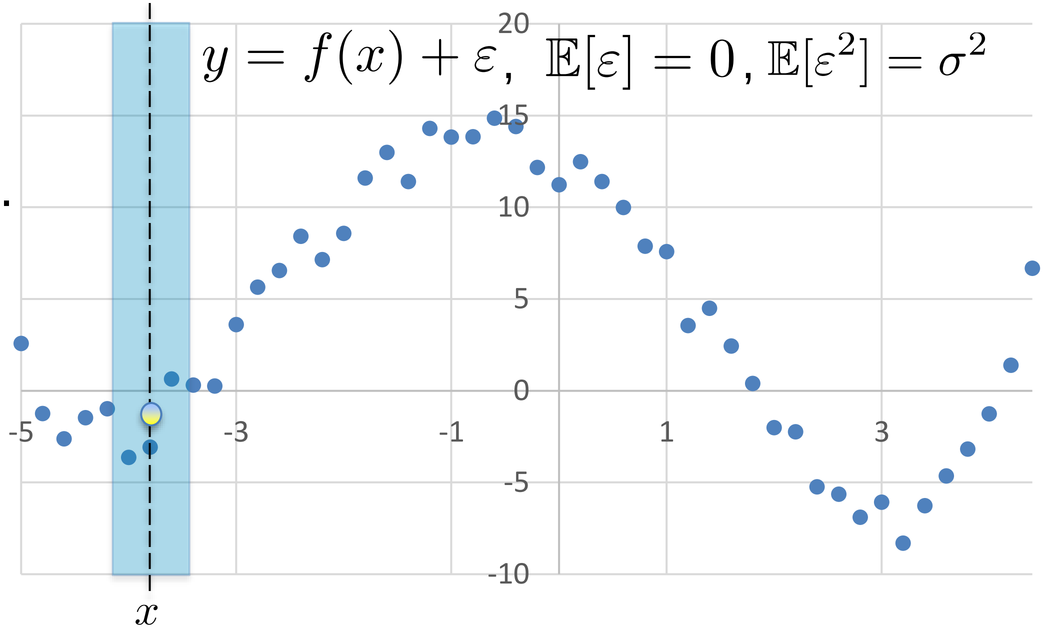
Large k:
less noise, but far-away, irrelevant points included

Bias vs. Variance Trade-off

$$y_i = f(x_i) + \varepsilon_i, \qquad i = 1, \ldots, n.$$

$$\varepsilon_i \text{ i.i.d., } \mathbb{E}[\varepsilon_i] = 0\,, \mathbb{E}[\varepsilon_i^2] = \sigma^2 < \infty\,.$$

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$



Expected Prediction Error (EPE):

$$\mathbb{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \mathbb{E}\left[(y - \mathbb{E}[y])^2\right] + \left(\mathbb{E}[y] - \mathbb{E}[\hat{f}(x)]\right)^2 + \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x)] - \hat{f}(x)\right)^2\right]$$

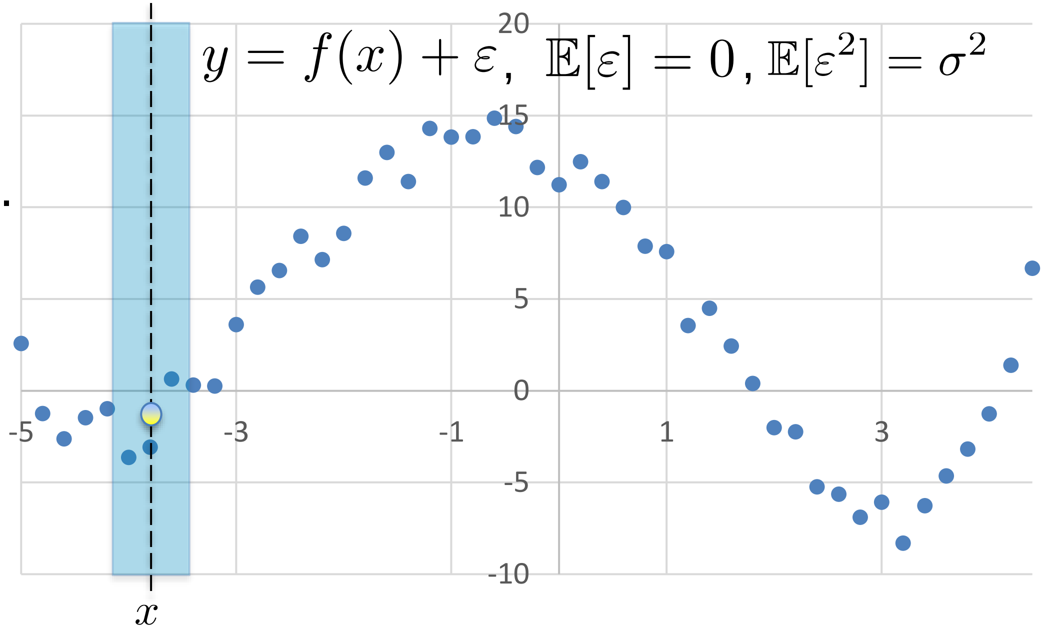
Northeastern

Bias vs. Variance Trade-off

$$y_i = f(x_i) + \varepsilon_i, \qquad i = 1, \ldots, n.$$

ε_i i.i.d., $\mathbb{E}[\varepsilon_i] = 0\,, \mathbb{E}[\varepsilon_i^2] = \sigma^2 < \infty.$

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$



Expected Prediction Error (EPE):

$$\mathbb{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \underbrace{\mathbb{E}\left[(y - \mathbb{E}[y])^2\right]}_{\text{inherent noise}} + \left(\mathbb{E}[y] - \mathbb{E}[\hat{f}(x)]\right)^2 + \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x)] - \hat{f}(x)\right)^2\right]$$

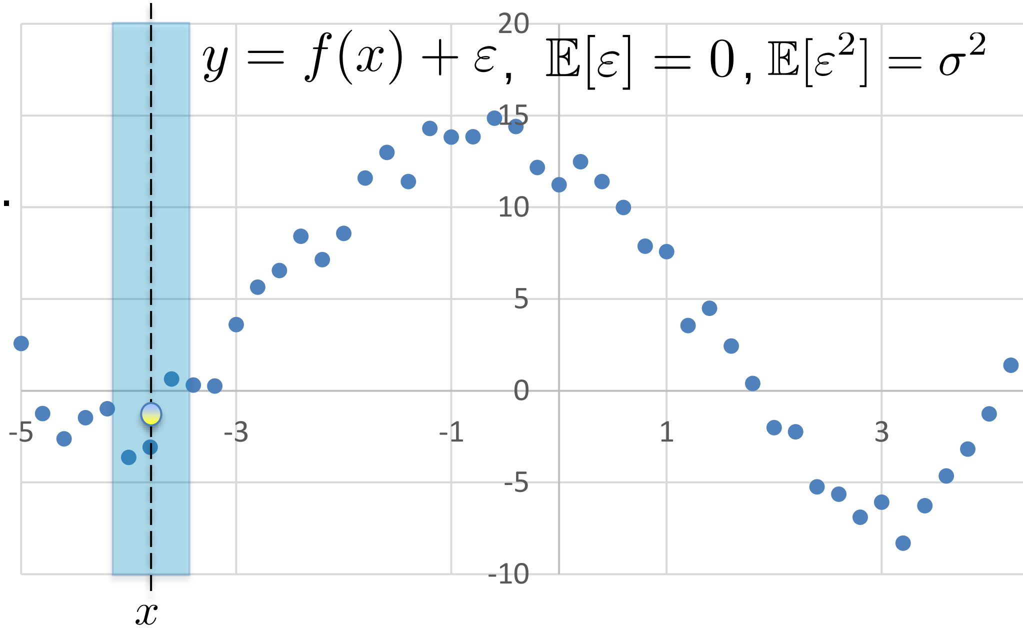
Northeastern

Bias vs. Variance Trade-off

$$y_i = f(x_i) + \varepsilon_i, \qquad i = 1, \ldots, n.$$

$$\varepsilon_i \text{ i.i.d., } \mathbb{E}[\varepsilon_i] = 0\,, \mathbb{E}[\varepsilon_i^2] = \sigma^2 < \infty\,.$$

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$



Expected Prediction Error (EPE):

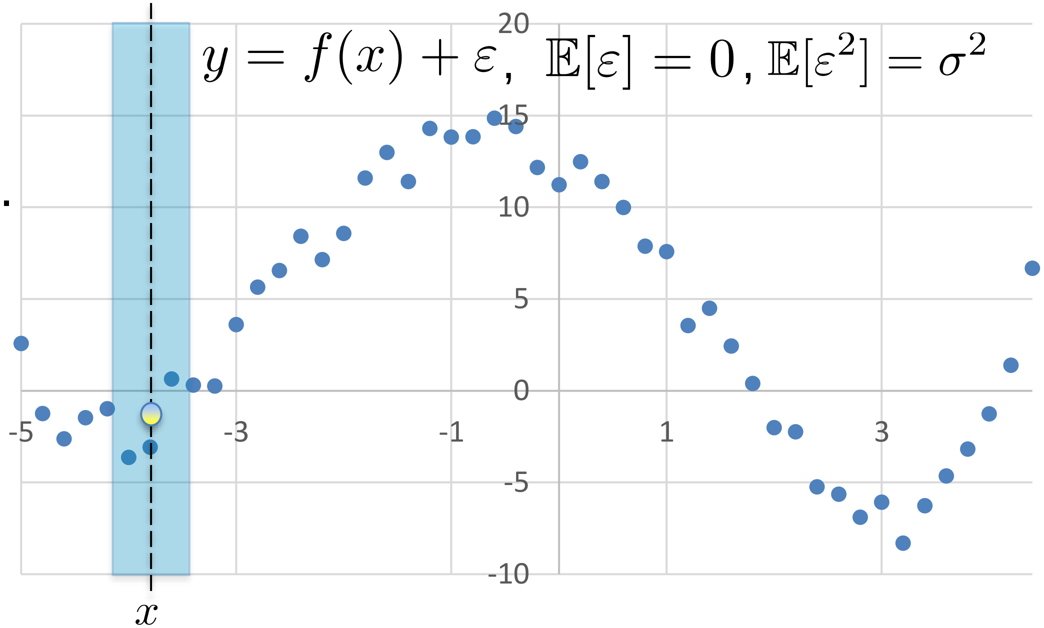
$$\mathbb{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \mathbb{E}\left[(y - \mathbb{E}[y])^2\right] + \underbrace{\left(\mathbb{E}[y] - \mathbb{E}[\hat{f}(x)]\right)}_{\text{estimator } \textbf{bias}}^2 + \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x)] - \hat{f}(x)\right)^2\right]$$

Bias vs. Variance Trade-off

$$y_i = f(x_i) + \varepsilon_i, \qquad i = 1, \ldots, n.$$

ε_i i.i.d., $\mathbb{E}[\varepsilon_i] = 0\,, \mathbb{E}[\varepsilon_i^2] = \sigma^2 < \infty.$

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$



Expected Prediction Error (EPE):

$$\mathbb{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \mathbb{E}\left[(y - \mathbb{E}[y])^2\right] + \left(\mathbb{E}[y] - \mathbb{E}[\hat{f}(x)]\right)^2 + \underbrace{\mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x)] - \hat{f}(x)\right)^2\right]}_{\text{estimator } \textbf{variance}}$$

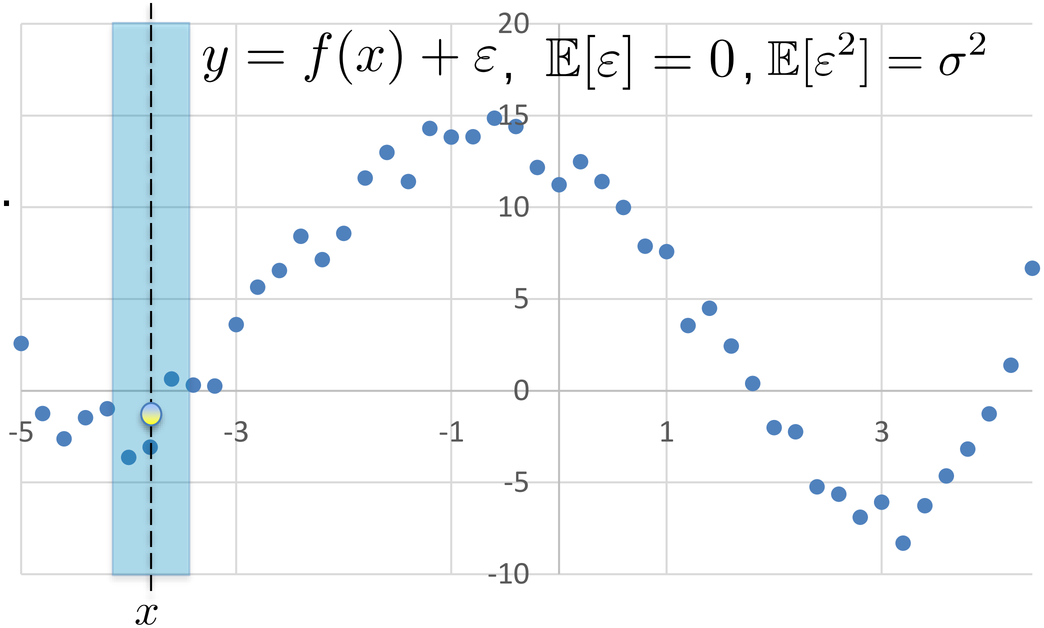
Northeastern

Bias vs. Variance Trade-off

$$y_i = f(x_i) + \varepsilon_i, \qquad i = 1, \ldots, n.$$

ε_i i.i.d., $\mathbb{E}[\varepsilon_i] = 0\,, \mathbb{E}[\varepsilon_i^2] = \sigma^2 < \infty.$

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$



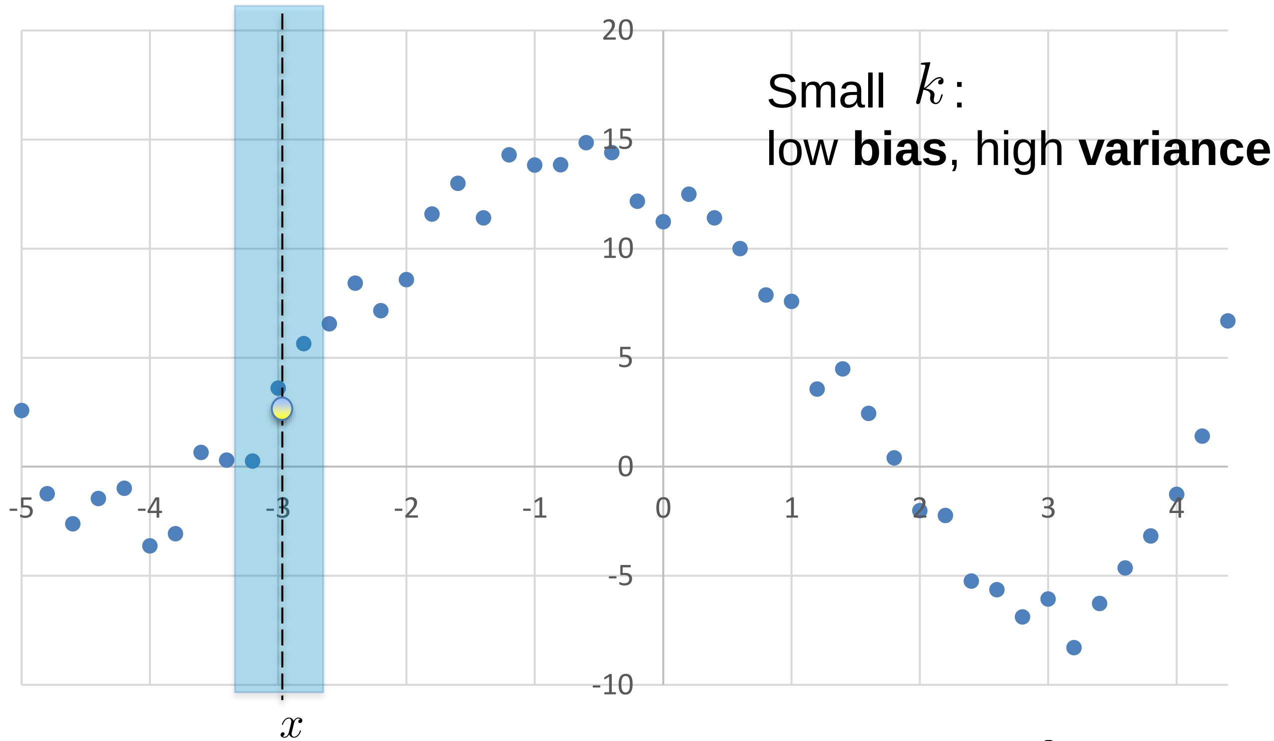
Expected Prediction Error (EPE):

$$\mathbb{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \mathbb{E}\left[(y - \mathbb{E}[y])^2\right] + \left(\mathbb{E}[y] - \mathbb{E}[\hat{f}(x)]\right)^2 + \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x)] - \hat{f}(x)\right)^2\right]$$

$$= \sigma^2 + \left(f(x) - \frac{1}{k} \sum_{i \in N_k(x)} f(x_i)\right)^2 + \frac{\sigma^2}{k}$$

Northeastern

Bias vs. Variance Trade-off

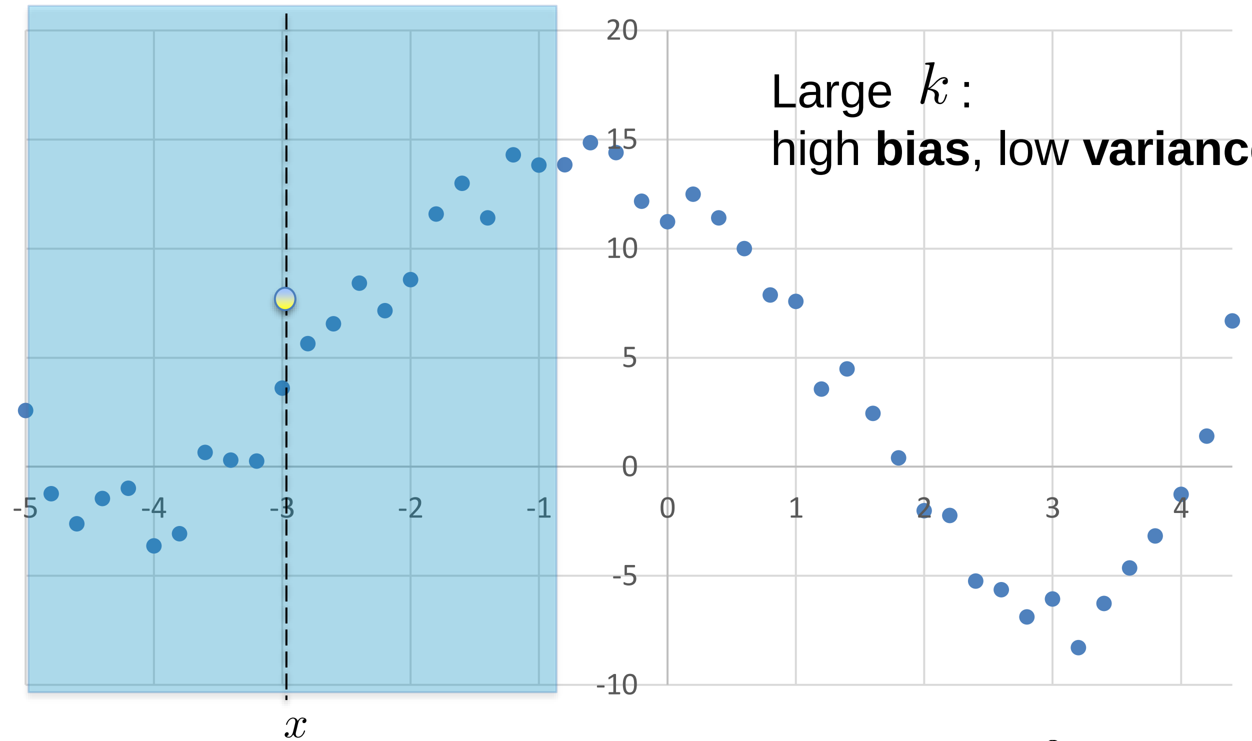


EPE: $\mathbb{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \sigma^2 + \underbrace{\left(f(x) - \frac{1}{k}\sum_{i \in N_k(x)} f(x_i)\right)^2}_{\text{estimator bias}} + \underbrace{\frac{\sigma^2}{k}}_{\text{estimator variance}}$

estimator **bias** estimator **variance**

Northeastern

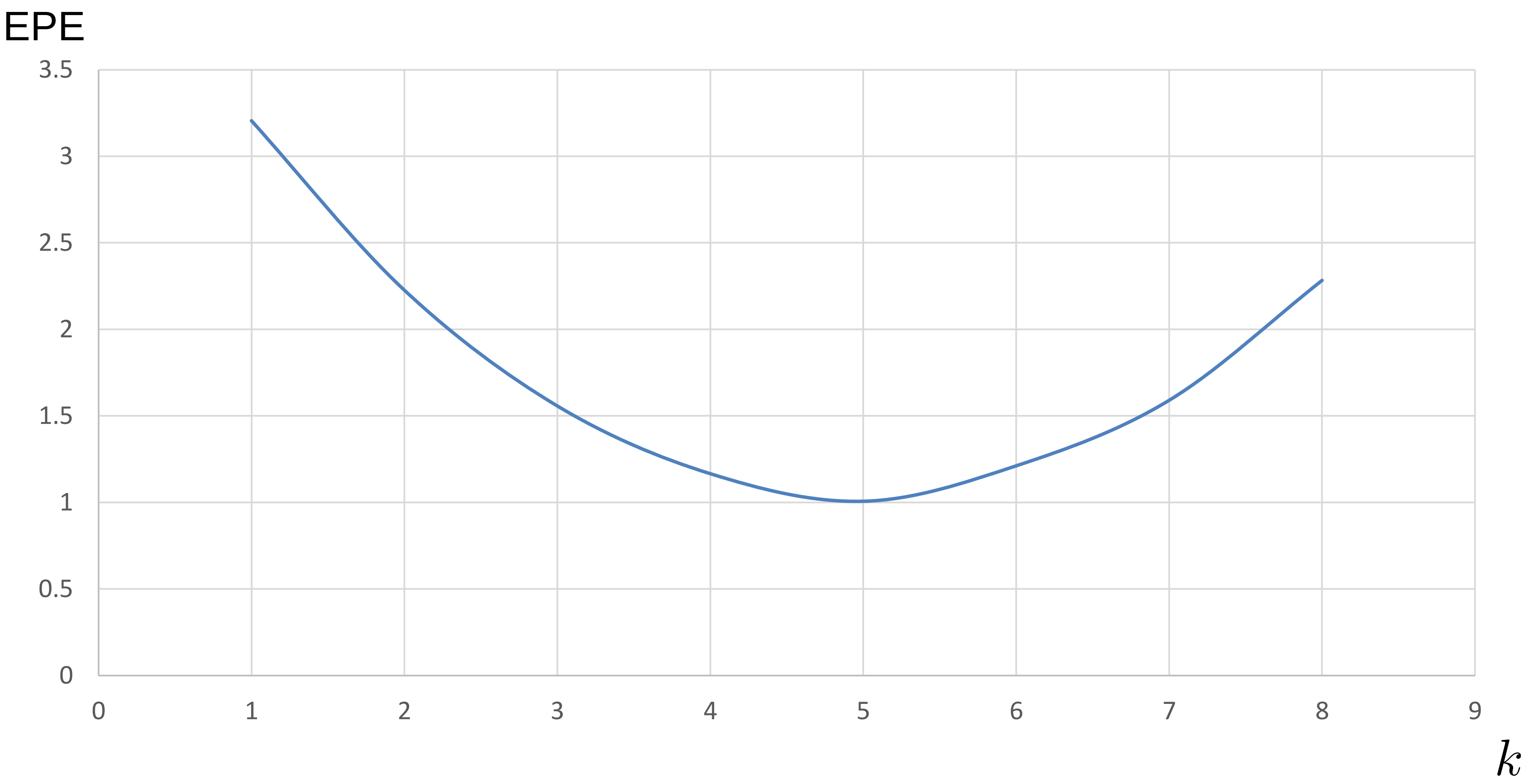
Bias vs. Variance Trade-off



Large k:
high **bias**, low **variance**

EPE: $\mathbb{E}\left[\left(y - \hat{f}(x)\right)^2\right] = \sigma^2 + \underbrace{\left(f(x) - \frac{1}{k}\sum_{i \in N_k(x)} f(x_i)\right)^2}_{\text{estimator bias}} + \underbrace{\frac{\sigma^2}{k}}_{\text{estimator variance}}$

Bias vs. Variance Tradeoff

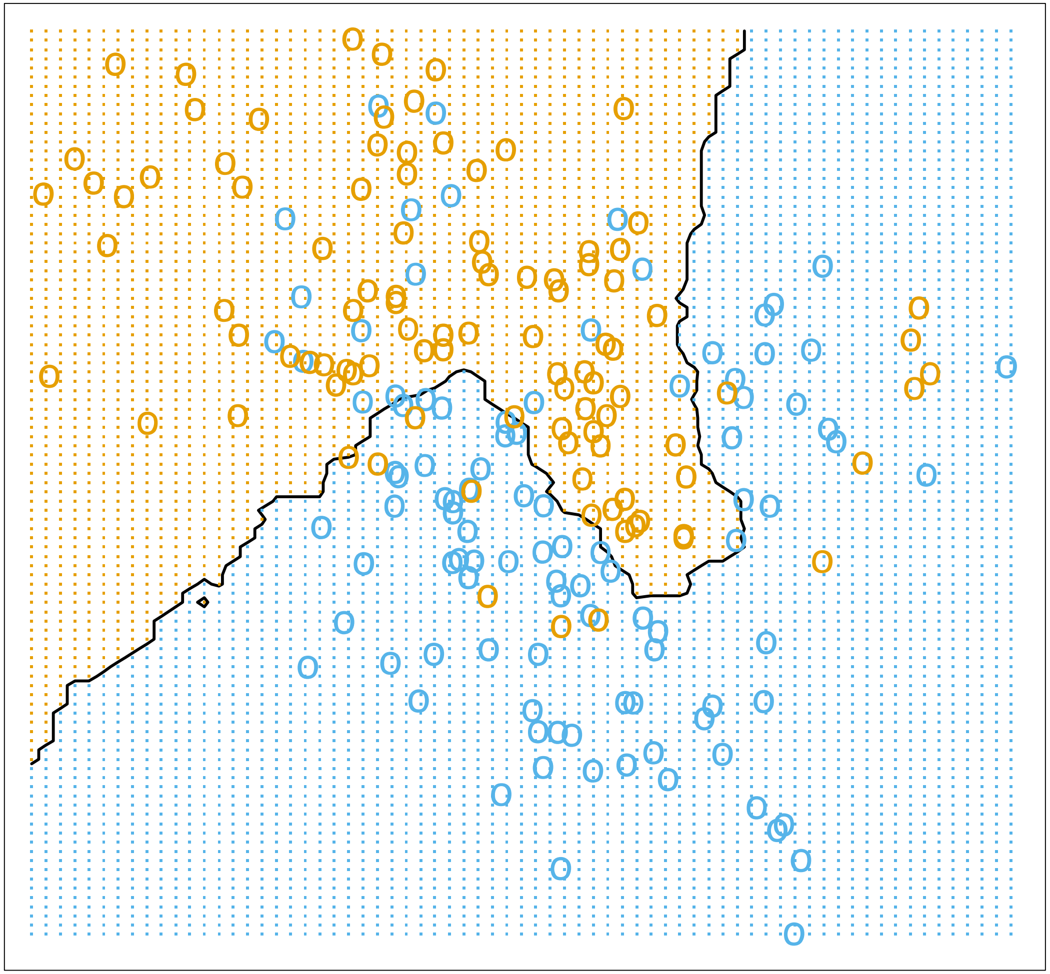


Northeastern

kNN for Classification

- Labels in discrete set C
- Majority vote:

$$\hat{f}(x) = \arg\max_{c \in C} |\{i \in N_k(x) : y_i = c\}|$$

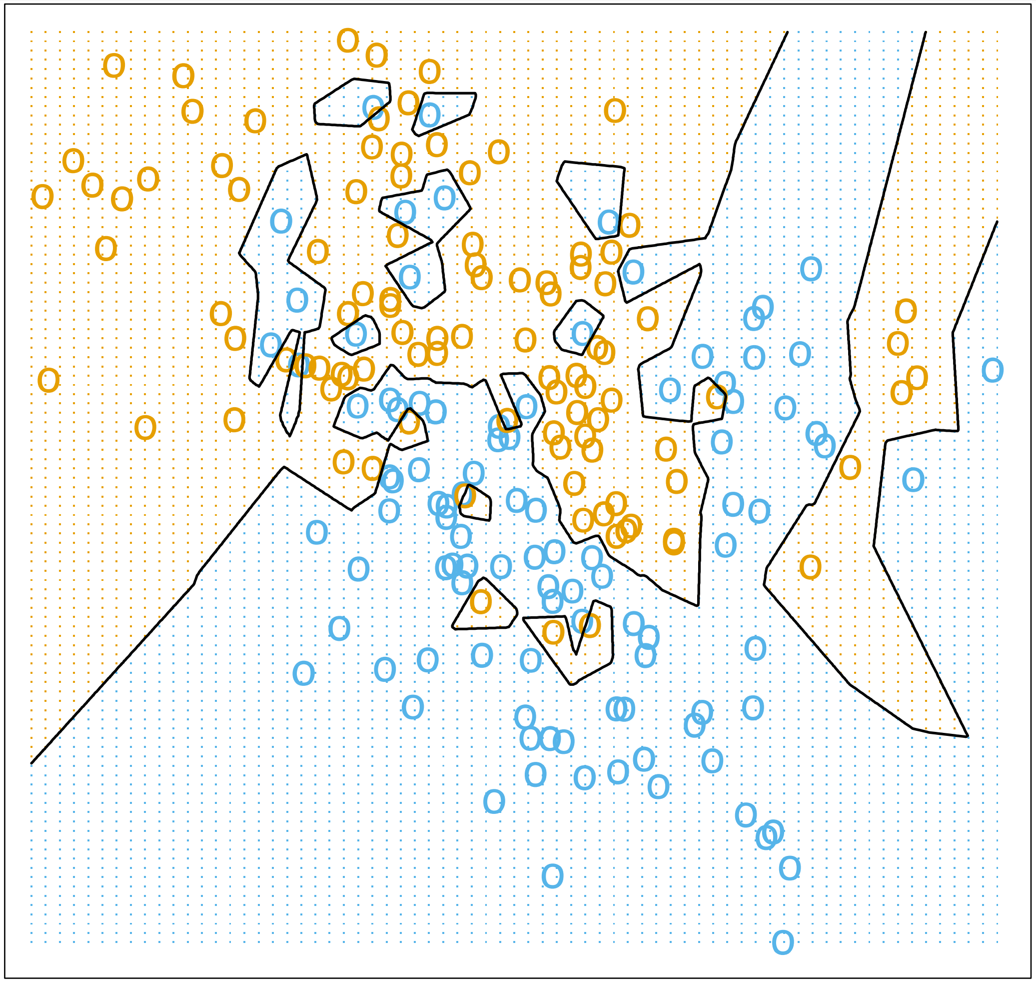


15-NN

kNN for Classification

- ❑ Labels in discrete set C
- ❑ Majority vote:

$$\hat{f}(x) = \arg\max_{c \in C} |\{i \in N_k(x) : y_i = c\}|$$

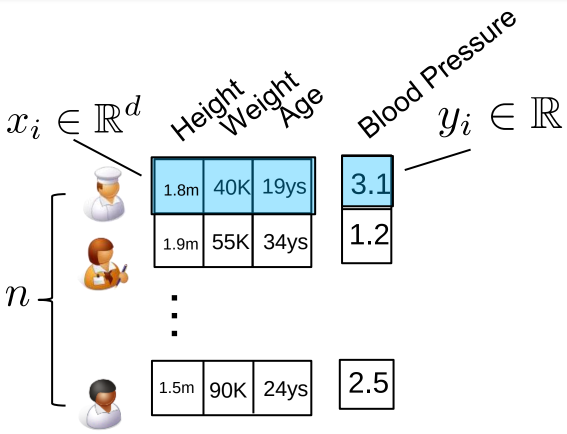


Northeastern

Why k-NN?

- ❑Almost no statistical assumption other than continuity (though smoothness helps)
- ❑Very simple to code!
- ❑Works well in many cases!

Implementation via Map-Reduce



$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$

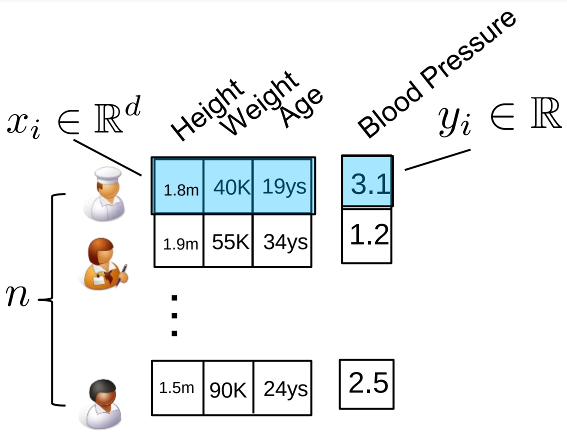
```
rdd = sc.parallelize(
        [(x1,y1), (x2,y2), ...,   (xn,yn)])

x = np.array([0.1,0.4,-2.0])
k = 10


k_nearest = rdd.map( lambda (xi,yi):
            (np.linalg.norm(xi-x),yi)
           ).takeOrdered(k, key=lambda (dist,y):dist)

y_hat = 1./k*sum([y for (dist,y) in k_nearest])
```

Implementation in Practice



$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$

- Use specifically designed **data structure** for nearest-neighbor queries
  - Cover trees
  - Locality Sensitive Hashing

- Cost per query:

$$O(k \cdot \text{poly}\log(n))$$

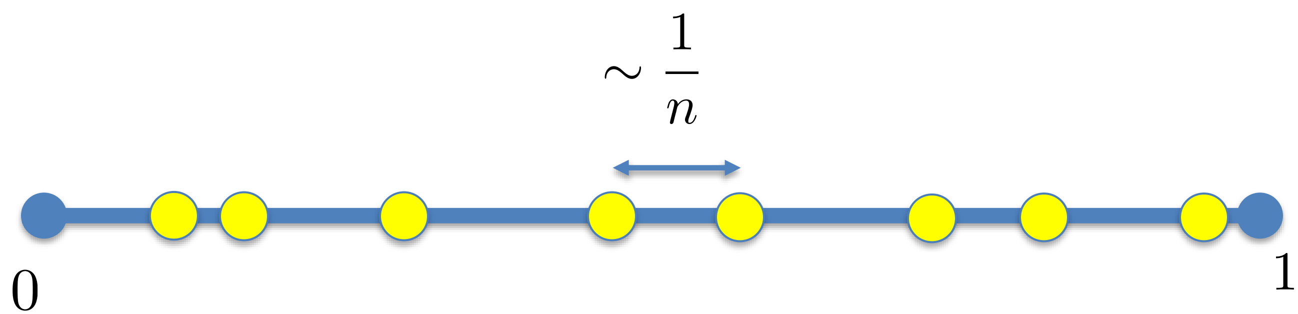
Why **not** k-NN?

… the curse of dimensionality

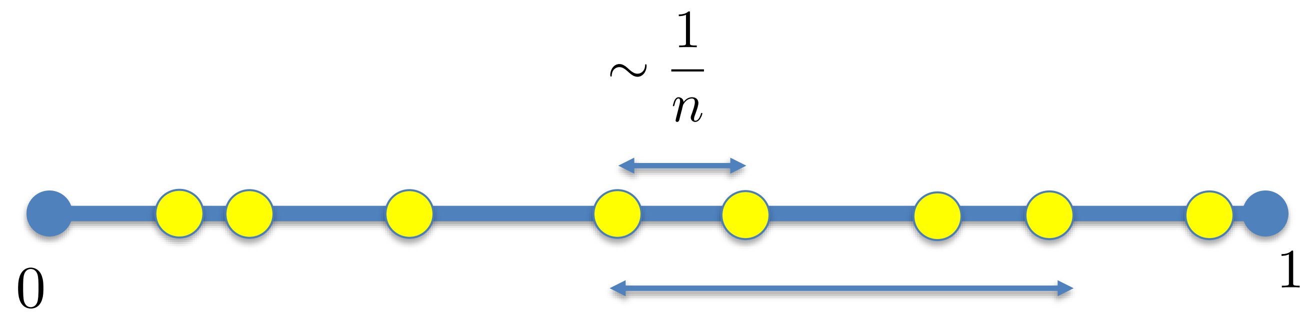
Curse of Dimensionality

- Suppose that $d = 1$, and that you have a dataset of n samples, where
$$x_i \in [0, 1], \quad i = 1, \ldots, n.$$

- Suppose that points are distributed u.a.r. over $[0, 1]$:



Curse of Dimensionality



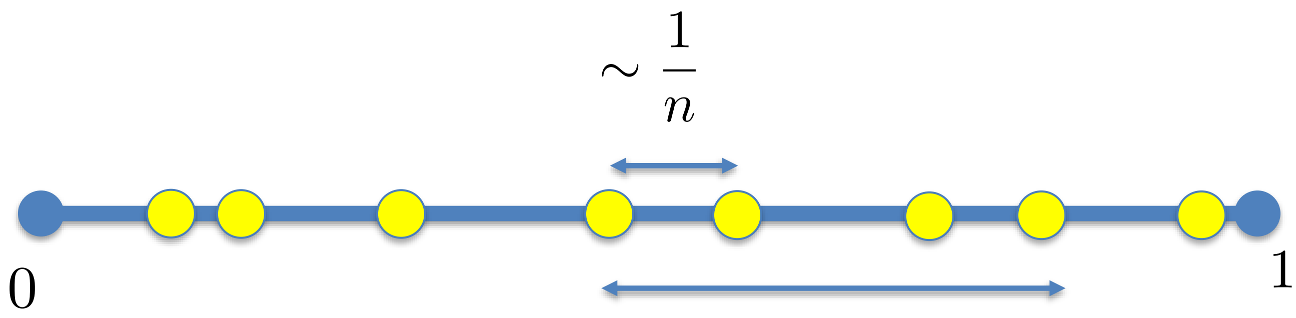
k -th neighbor at distance at most $\sim \frac{k}{n}$

As you increase the number of samples n, k-NN becomes **less biased!**

If you increase k slowly enough with n, e.g., $k = \log n$, $k = \sqrt{n}$
both bias **and** variance will **go to zero!**

Northeastern

Eliminating Bias & Variance as k Increases with n



k -th neighbor at distance at most $\sim \frac{k}{n}$

Let $k = \sqrt{n}$.

Then, for al $i \in N_k(x)$, $|x_i - x| \lesssim \frac{k}{n} = \frac{1}{\sqrt{n}} \to 0$ as $n \to \infty$.

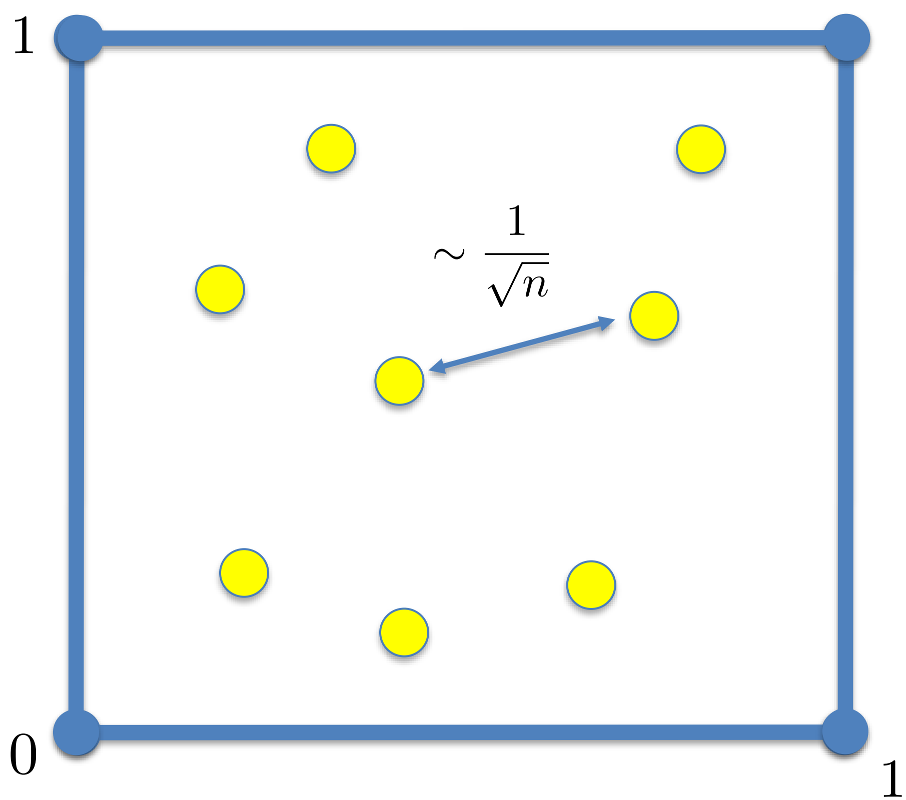
Hence, for all $i \in N_k(x)$, $x_i \to x$, so by continuity $\hat{f}(x) \to f(x)$, so bias $(\hat{f}(x) - f(x))^2 \to 0$ as $n \to \infty$.

On the other hand, as $k = \sqrt{n}$, variance $\frac{\sigma^2}{k} \to 0$ as $n \to \infty$.

Curse of Dimensionality

- What if $d = 2$?

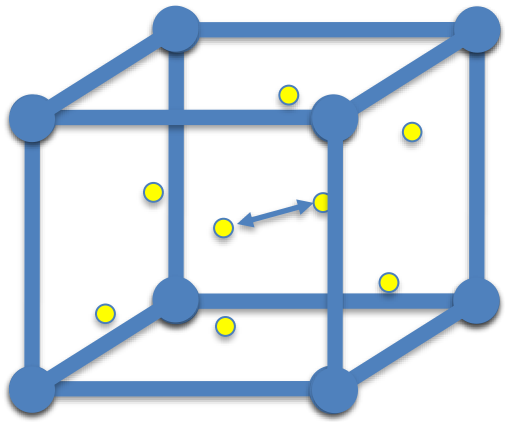
$$x_i \in [0,1]^2, \quad i = 1, \ldots, n.$$



Curse of Dimensionality

- What if $d = 3$?

$$x_i \in [0,1]^3, \quad i = 1, \ldots, n.$$

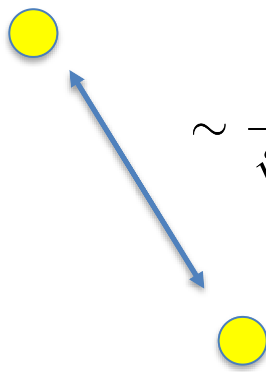


$$\sim \frac{1}{\sqrt[3]{n}}$$

Curse of Dimensionality

- For arbitrary d

$$x_i \in [0,1]^d, \quad i = 1, \ldots, n.$$



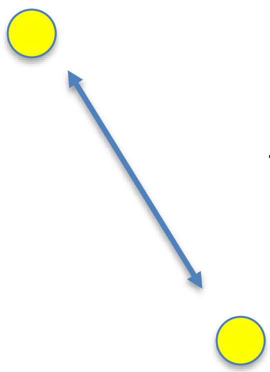
$$\sim \frac{1}{\sqrt[d]{n}}$$

$$n = 100, d = 100$$

$$\frac{1}{\sqrt[d]{n}} \approx 0.954992586021436$$

- Extremely **low density**
- Points are lie on **opposite boundaries**!

Curse of Dimensionality

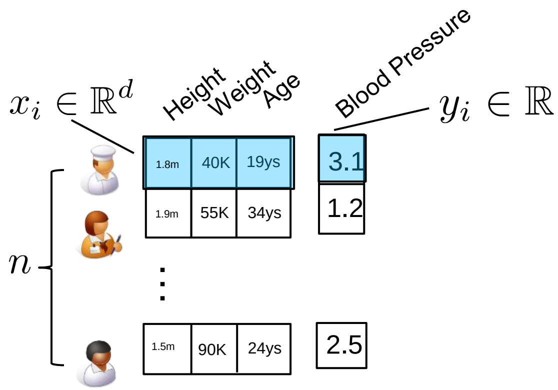


$$\frac{1}{\sqrt[d]{n}} \leq \varepsilon \qquad \Leftrightarrow \qquad n \geq \left(\frac{1}{\epsilon}\right)^d$$

$$\left.\begin{array}{l} \varepsilon = 0.1 \\ d = 100 \end{array}\right\} \Rightarrow n \geq 10^{100}$$

- **Curse of Dimensionality**: To maintain an unbiased estimate with k-NN, the size of the dataset needs to grow **exponentially** with the dimension size!!!

Summary

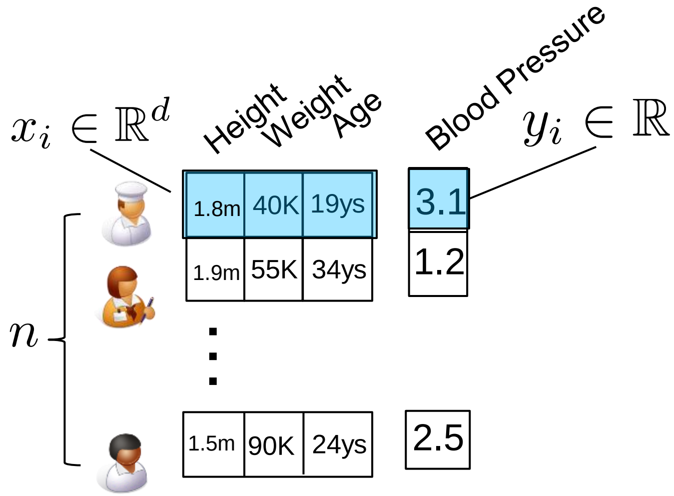


$$y_i \approx f(x_i), \quad i = 1, \ldots, n$$

- To regress f from data, we *need* to make *some assumption* on f …
- The assumption of *continuity* led us to k-NN…
- k-NN suffers from curse of dimensionality…
- Now what?

Add more assumptions!!!!

Regression



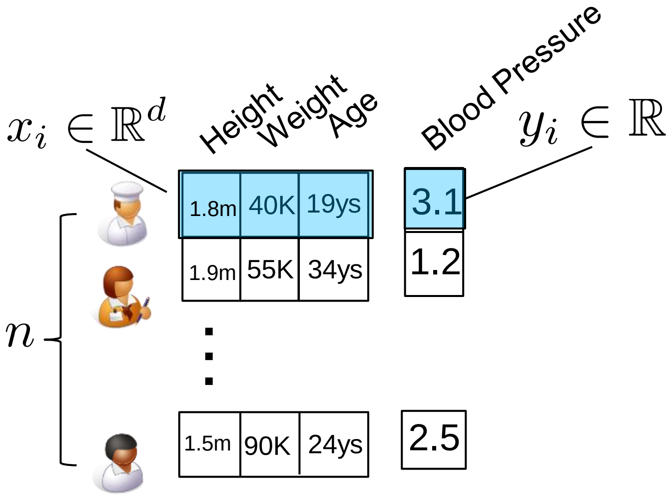
random "noise" variables

$$y_i = f(x_i) + \varepsilon_i, \qquad i = 1, \ldots, n.$$

where ε_i are **independent and identically distributed** (i.i.d), and

$$\mathbb{E}[\varepsilon_i] = 0 \qquad \mathbb{E}[\varepsilon_i^2] = \sigma^2 < \infty$$

Linear Regression



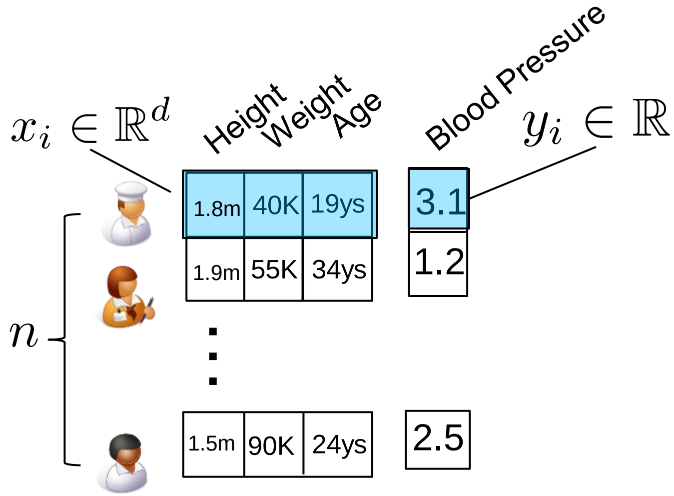
$$y_i \approx f(x_i) = \beta^\top x_i = \sum_{k=1}^{d} \beta_k x_{ik}$$

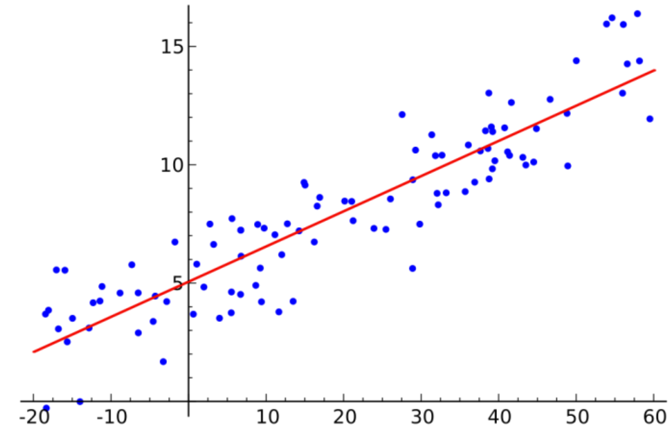
Assumption: There exists $\beta \in \mathbb{R}^d$ such that:

$$y_i = \langle \beta, x_i \rangle + \varepsilon_i, \quad i = 1, \ldots, n$$

where ε_i are i.i.d., $\mathbb{E}[\varepsilon_i] = 0$, $\mathbb{E}[\varepsilon_i^2] = \sigma^2 < \infty$

Affine Can Be Written as Linear





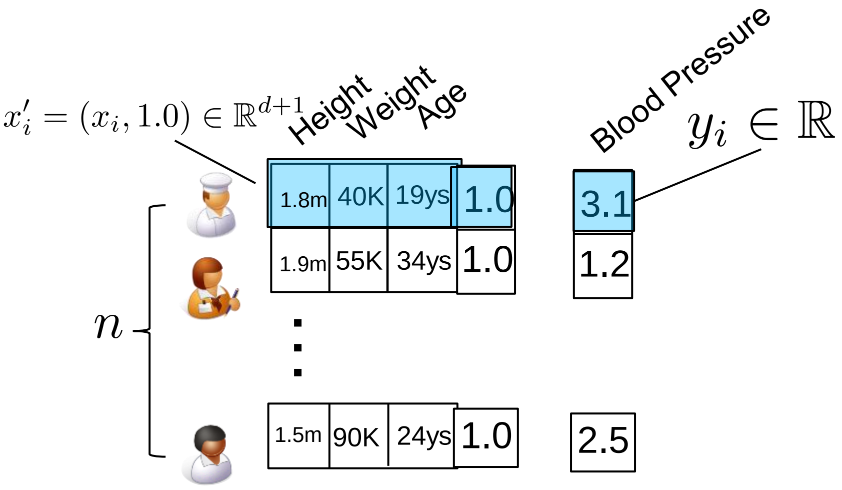
bias, offset

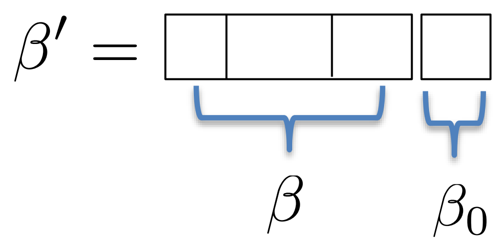
$$f(x_i) = \beta^\top x_i + \beta_0 \text{, where } \beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}$$

$$= \beta'^\top x'_i, \qquad \text{where } \beta' = (\beta, \beta_0) \in \mathbb{R}^{d+1}$$

$$x'_i = (x_i, 1.0) \in \mathbb{R}^{d+1}$$

Affine Can Be Written as Linear



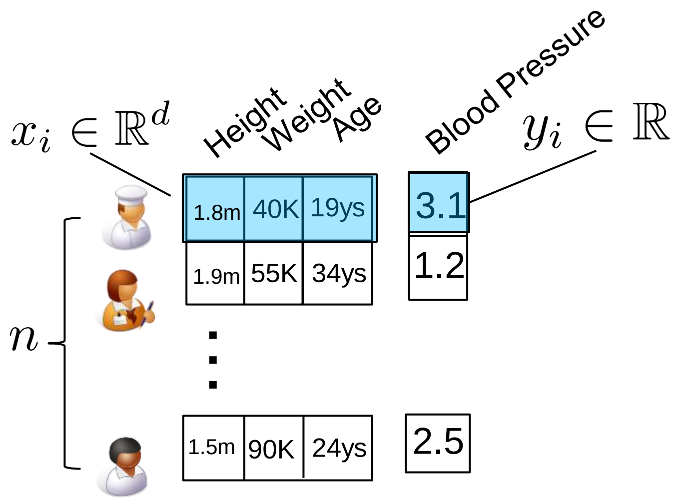


$$f(x_i) = \beta^\top x_i + \beta_0 \text{, where } \beta \in \mathbb{R}^d,\ \beta_0 \in \mathbb{R}$$

$$= \beta'^\top x_i', \qquad \text{where } \beta' = (\beta, \beta_0) \in \mathbb{R}^{d+1}$$

$$x_i' = (x_i, 1.0) \in \mathbb{R}^{d+1}$$

Operations that Preserve Linearity



- Affine in $\mathbb{R}^d$ is linear in $\mathbb{R}^{d+1}$
- Linear transforms on features (i.e., rescaling):
  - E.g., from kilograms to pounds
- Affine transforms in features (i.e., rescaling and shifting):
  - E.g., from Fº to Cº.

Binary Features

Benders

= 1.0

$x_i \in \mathbb{R}^d$

Bender	Weight	Age
1	40K	19ys
1	55K	34ys
⋮		
0	90K	24ys

n

Blood Pressure
3.1
1.2
2.5

$y_i \in \mathbb{R}$

Non-Benders

= 0.0

$x =$

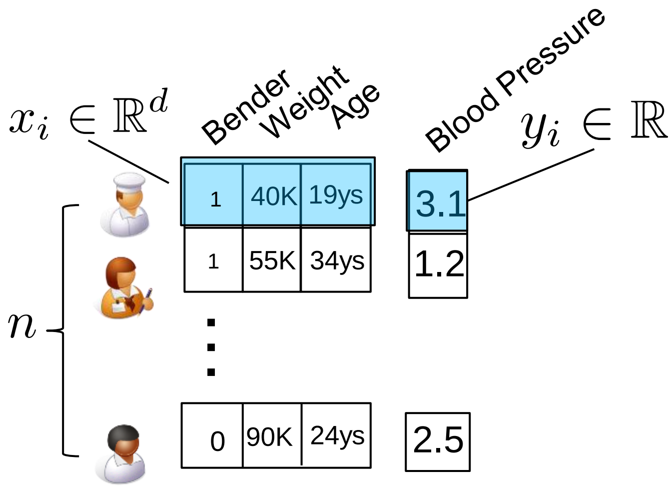
1.0	90K	24ys

$\beta =$

0.5	-3.1	1.2

class bias

Binary Features



Bender

= 1.0

Non-Bender

= 0.0

$$x = \begin{array}{c} \text{Bender} \\ \begin{bmatrix} 1.0 & 70\text{K} & 16\text{ys} \end{bmatrix} \end{array} \qquad x' = \begin{array}{c} \text{Bender} \\ \begin{bmatrix} 0.0 & 70\text{K} & 16\text{ys} \end{bmatrix} \end{array}$$

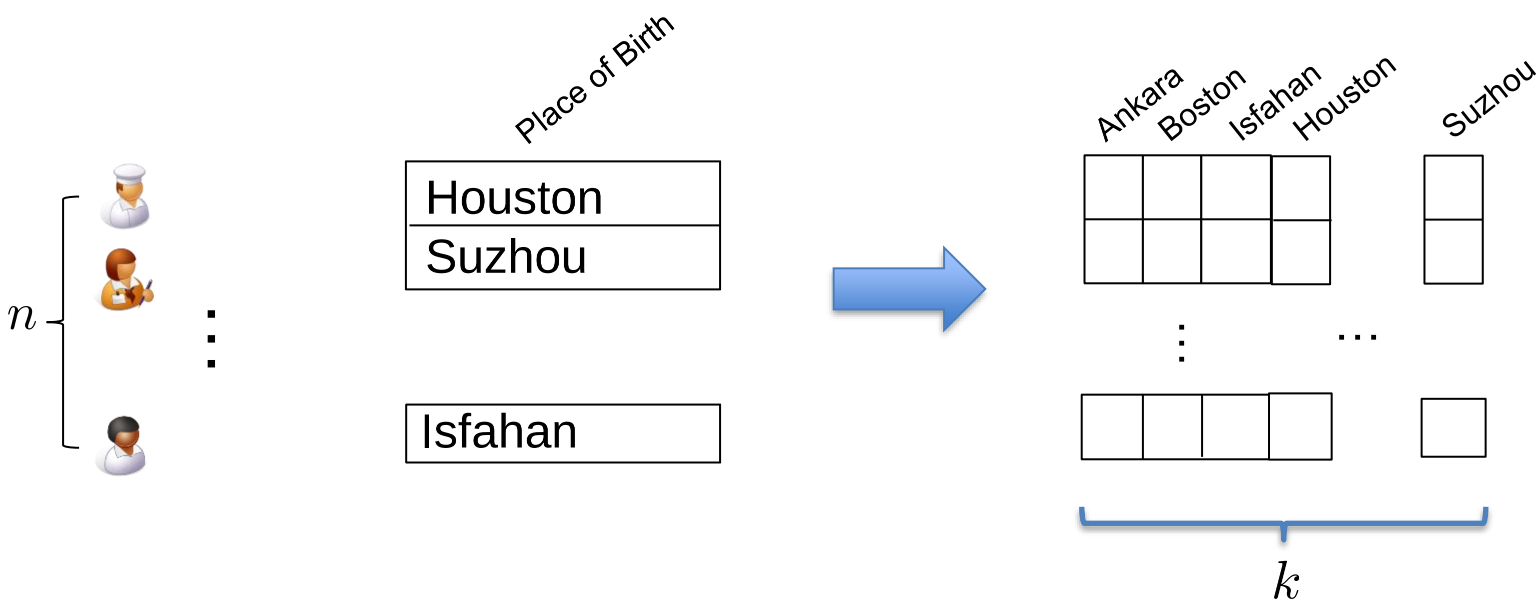
$$\beta = \begin{bmatrix} \boxed{0.5} & -3.1 & 1.2 \end{bmatrix}$$

class bias

$$\mathbb{E}[y - y'] = \beta^\top x - \beta^\top x' = 0.5$$

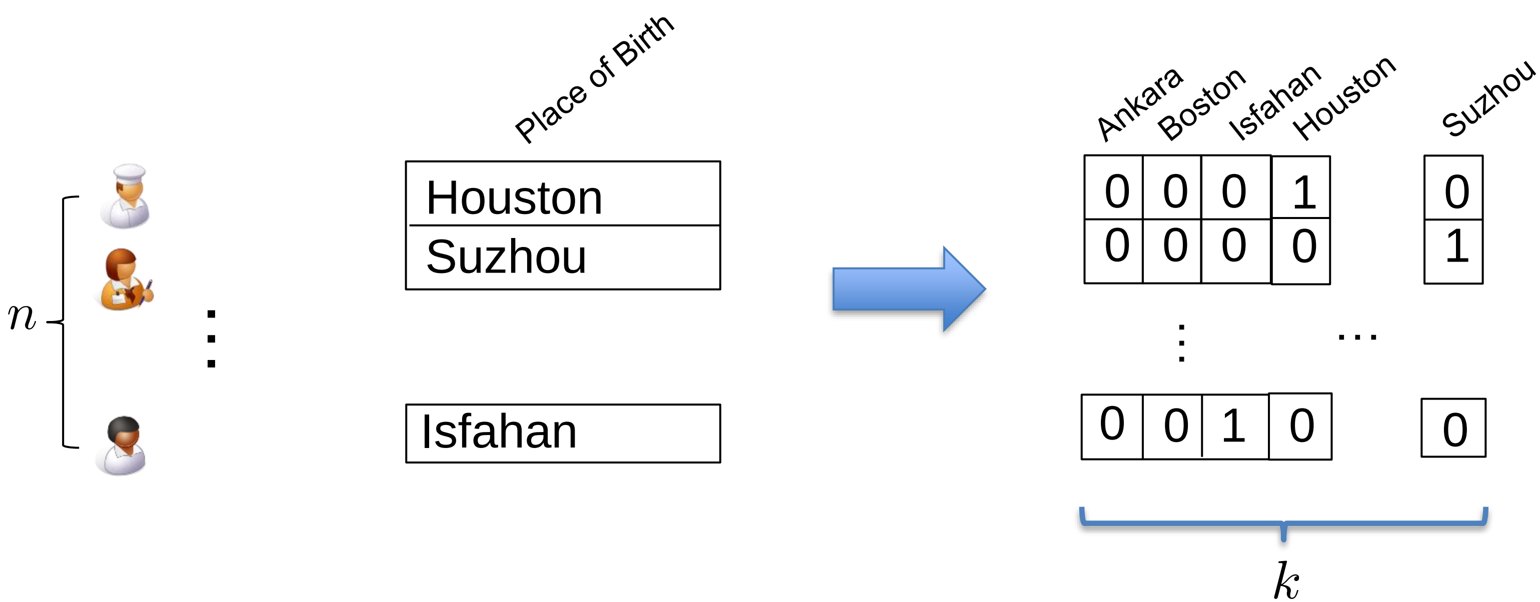
Northeastern

Categorial Features: Binarization



Categories (cities) = k

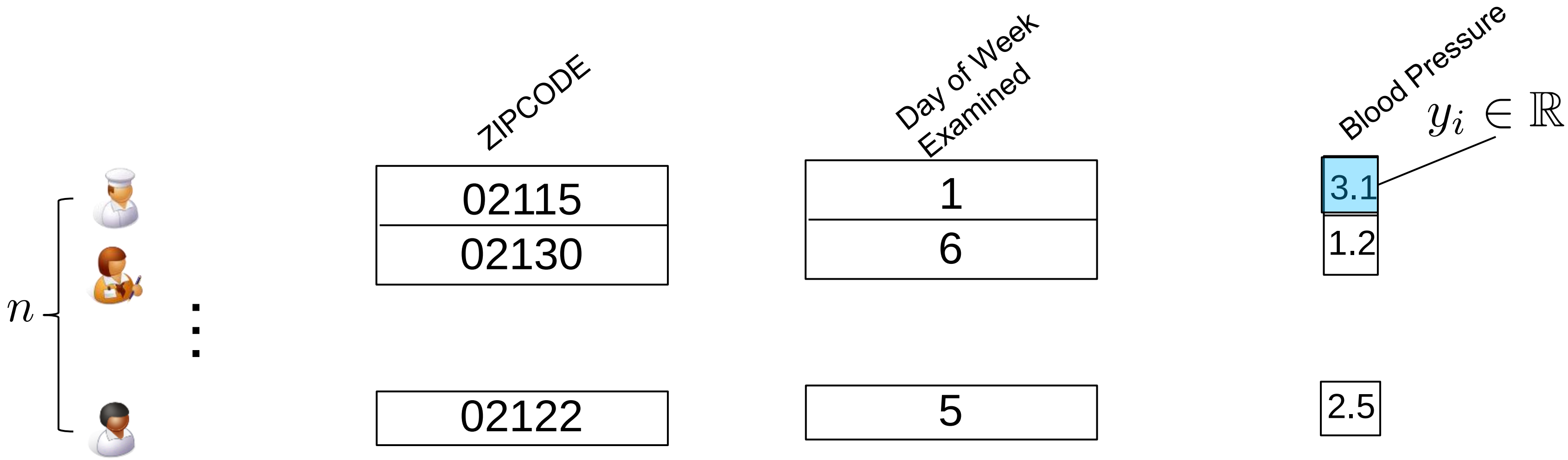
Categorial Features: Binarization



Categories (cities) = k

- ❑ Categorical features are very common: locations, genes, words in document
- ❑ Binarization leads to feature vectors that are **sparse:** most elements are 0!

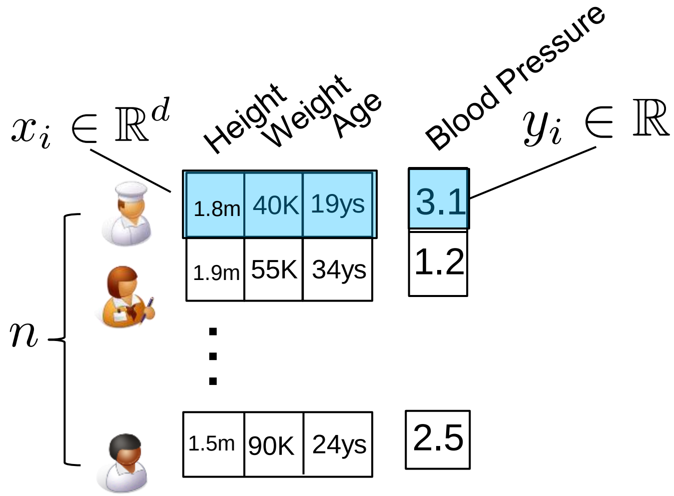
Numeric Features May Be Categorical!!!!!



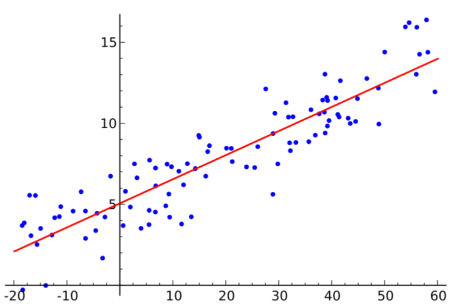
Mon: 1
Tue: 2
…
Sun: 7

Rule of thumb: if 2 does not mean "2 times" 1, treat it as categorical

Least Squares Estimator (LSE)



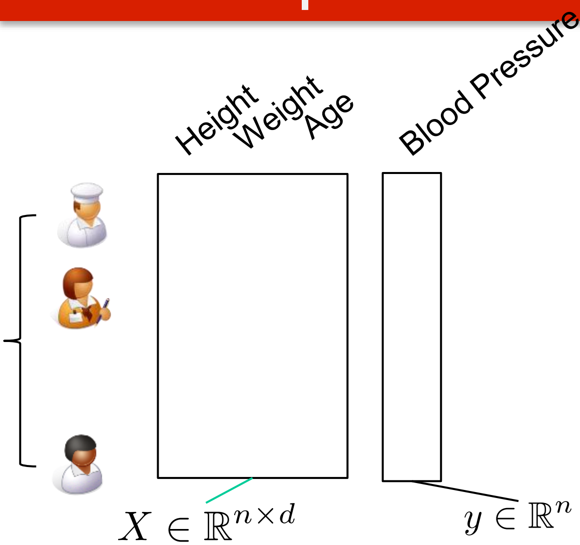
$$y_i \approx f(x_i) = \beta^\top x_i = \sum_{k=1}^{d} \beta_k x_{ik}$$



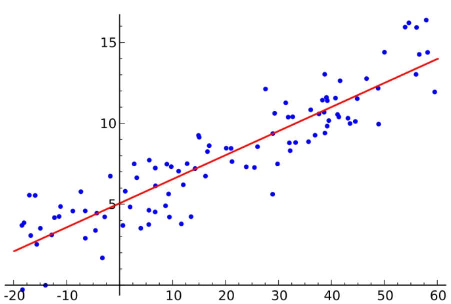
Estimate of β

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2$$

Least Squares Estimator (LSE)



$$y_i \approx f(x_i) = \beta^\top x_i = \sum_{k=1}^{d} \beta_k x_{ik}$$



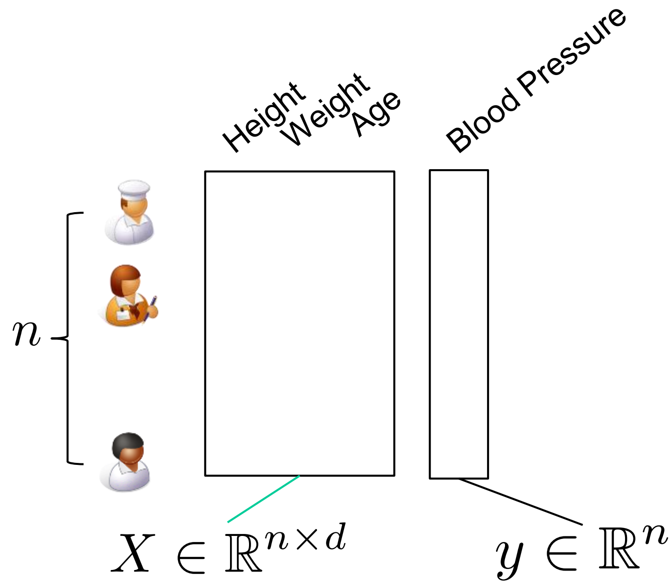
Why LSE?

Estimate of β

$$\begin{aligned}\hat\beta &= \arg\min_{\beta\in\mathbb{R}^d} \sum_{i=1}^{n} (y_i - \langle \beta, x_i\rangle)^2 \\ &= \arg\min_{\beta\in\mathbb{R}^d} \|X\beta - y\|_2^2 \\ &= (X^T X)^{-1} X^T y\end{aligned}$$

Northeastern

Reason #1: If Noise is Gaussian, LSE is an MLE!



$$y_i = \beta^\top x_i + \varepsilon_i, \quad i = 1,\ldots,n$$

ε_i i.i.d., $\mathbb{E}[\varepsilon_i] = 0, \mathbb{E}[\varepsilon_i^2] = \sigma^2 < \infty$

- Suppose, in addition, that

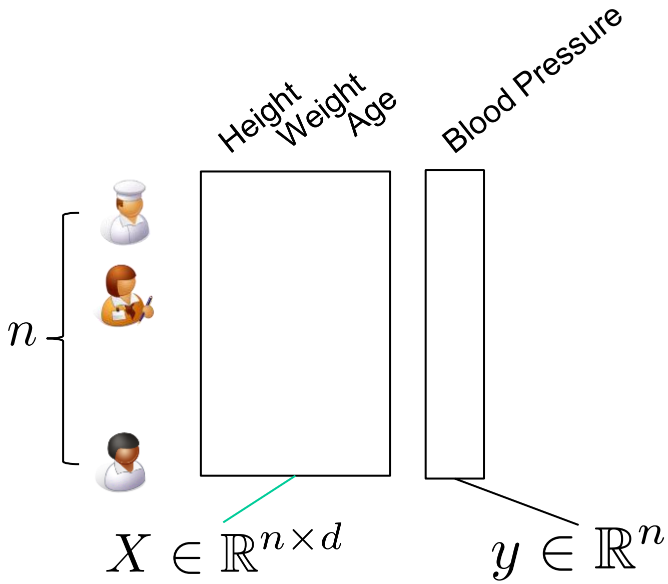
$$\varepsilon_i \sim N(0,\sigma^2)$$

Then, the negative log-likelihood of the labels is:

$$-\log\left(P\left(y|\beta,X\right)\right) = -\log\left(\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi}\sigma}e^{-(y_i-\beta^\top x_i)^2/2\sigma^2}\right)$$

$$= \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta^\top x_i)^2 + C$$

Reason #1: If Noise is Gaussian, LSE is an MLE!



$$y_i = \beta^\top x_i + \varepsilon_i, \quad i = 1, \ldots, n$$

ε_i i.i.d., $\mathbb{E}[\varepsilon_i] = 0, \mathbb{E}[\varepsilon_i^2] = \sigma^2 < \infty$

- Suppose, in addition, that

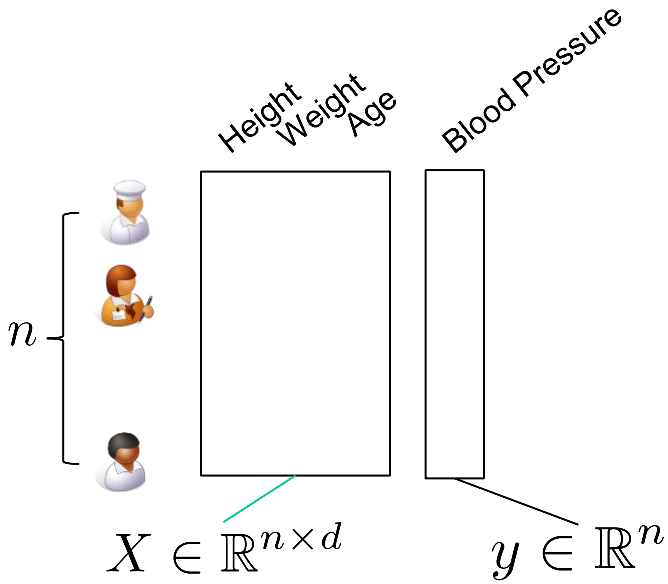
What if $\varepsilon_i \sim \cancel{N(0, \sigma^2)}$ **?**

Then, LSE is a **Maximum Likelihood Estimator**:

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2 = \arg\min_{\beta \in \mathbb{R}^d} \; -\log\left(P\left(y|\beta, X\right)\right)$$

$$= \arg\max_{\beta \in \mathbb{R}^d} \; P\left(y|\beta, X\right)$$

Additional Properties of LSE

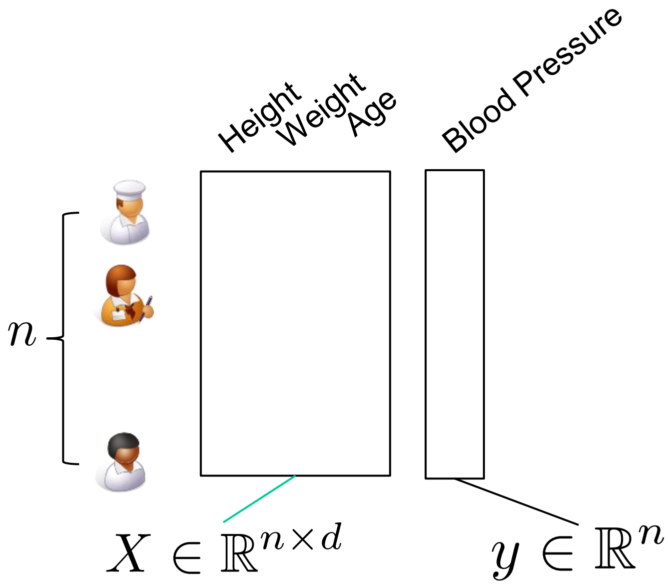


$$y_i = \beta^\top x_i + \varepsilon_i, \quad i = 1, \ldots, n$$

$$\begin{aligned}\hat{\beta} &= \arg\min_{\beta\in\mathbb{R}^d} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2 \\ &= (X^T X)^{-1} X^T y\end{aligned}$$

- Expectation: $\mathbb{E}[\hat{\beta}] = \beta$, i.e., LSE is **unbiased**.
- Covariance: $\text{Cov}(\hat{\beta}) = \mathbb{E}[\hat{\beta} - \beta)(\hat{\beta} - \beta)^\top] = \sigma^2 (X^\top X)^{-1} \succeq 0$
- Estimator (and covariance) is **undefined** if $\text{rank}(X) < d$!

Expected Prediction Error (EPE):



$$y_i = \beta^\top x_i + \varepsilon_i, \quad i = 1, \ldots, n$$

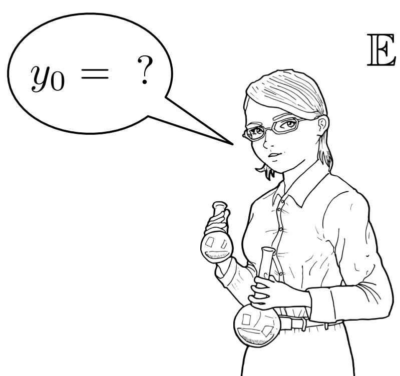
$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2$$
$$= (X^T X)^{-1} X^T y$$

Estimate: $\hat{y}_0 = \hat{\beta}^\top x_0$

Expected Prediction Error:

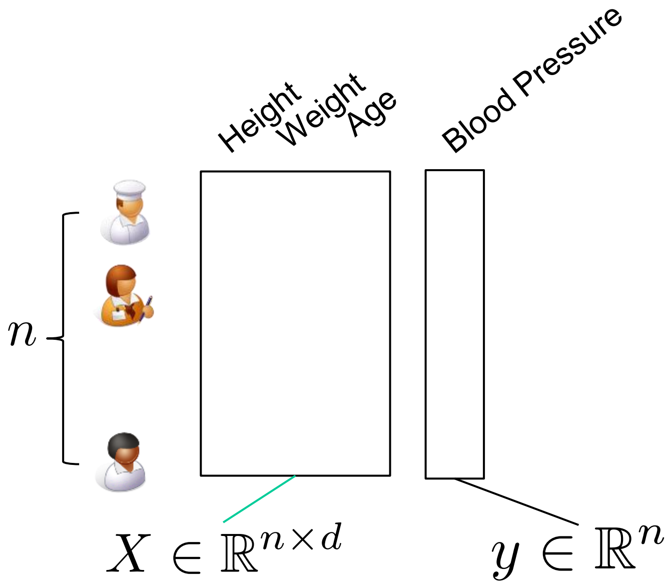
$$x_0 = \boxed{1.8\text{m}}\boxed{90\text{K}}\boxed{24\text{ys}}$$

$$y_0 = \beta^\top x_0 + \varepsilon_0$$



$$\begin{aligned} \mathbb{E}[(y_0 - \hat{y}_0)^2] &= \mathbb{E}[(y_0 - \beta^\top x_0)^2] + \mathbb{E}[(\beta^\top x_0 - \hat{\beta}^\top x_0)^2] \\ &= \sigma^2 + x_0{}^\top \mathbb{E}[(\beta - \hat{\beta})(\beta - \hat{\beta})^\top] x_0 \\ &= \underbrace{\sigma^2}_{\text{inherent noise}} + \underbrace{x_0{}^\top \mathsf{Cov}(\hat{\beta}) x_0}_{\text{variance in direction } x_0} \end{aligned}$$

A Few Observations on Covariance



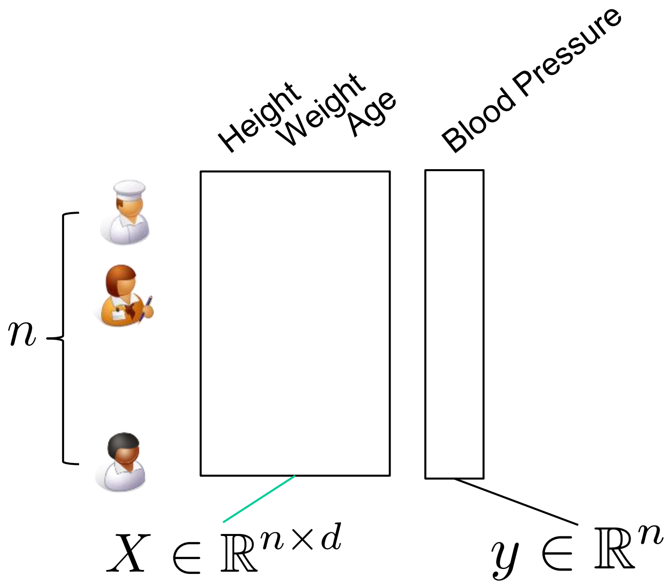
$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2$$

$$\text{Cov}(\hat{\beta}) = \mathbb{E}[\hat{\beta} - \beta)(\hat{\beta} - \beta)^\top] = \sigma^2 (X^\top X)^{-1} \succeq 0$$

$$\text{EPE: } \mathbb{E}[(y_0 - \hat{y}_0)^2] = \sigma^2 + x_0^\top \text{Cov}(\hat{\beta}) x_0$$

- 1-dimension: covariance = variance
- Variance in a specific direction: $\mathtt{Var}[\langle \hat{\beta}, x \rangle] = x^T \mathtt{Cov}(\hat{\beta}) x \geq 0$
- **Eigenvalues** of $\mathtt{Cov}(\hat{\beta})$ summarize **variability in all directions.**

Why LSE if Noise is Non-Gaussian?

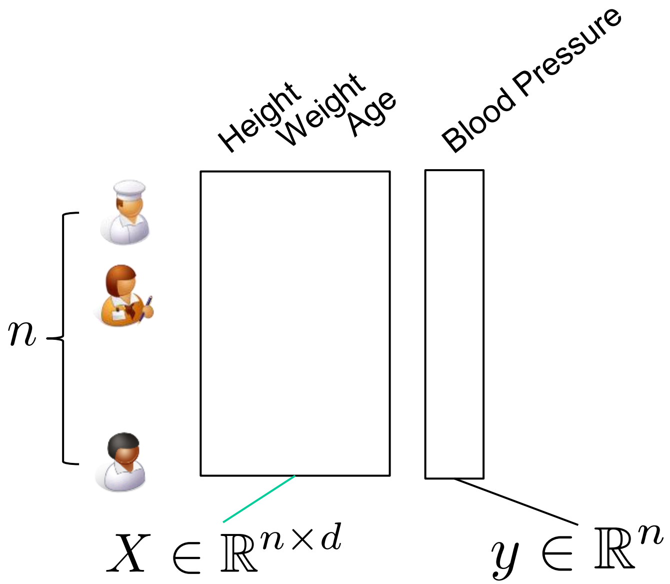


$$\hat{\beta} = \arg\min_{\beta\in\mathbb{R}^d} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2 = (X^T X)^{-1} X^T y$$

- An estimator $\hat{\beta}$ of β is called **unbiased** if $\mathbb{E}[\hat{\beta}] = \beta$, for all $\beta \in \mathbb{R}^d$.
- An estimator $\hat{\beta}$ of β is called **linear** if $\hat{\beta} = D(X)y$

➡ LSE is both unbiased and linear

Gauss-Markov Theorem



$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (y_i - \langle \beta, x_i \rangle)^2$$
$$= (X^T X)^{-1} X^T y$$

- **Theorem**: LSE is a Best Linear Unbiased Estimator (BLUE):

$$\mathrm{cov}(\hat{\beta}) \preceq \mathrm{cov}(\hat{\beta}') \text{ for any } \hat{\beta}' \text{ s.t. } \hat{\beta}' = D(X)y \text{ and } \mathbb{E}[\hat{\beta}] = \beta$$

Proof


- **Theorem**: LSE is a Best Linear Unbiased Estimator (BLUE):

$$\mathrm{cov}(\hat{\beta}) \preceq \mathrm{cov}(\hat{\beta}') \text{ for any } \hat{\beta}' \text{ s.t. } \hat{\beta}' = D(X)y \text{ and } \mathbb{E}[\hat{\beta}] = \beta$$

Let $\Delta = D - (X^\top X)^{-1}X^\top$. Then:

$$\begin{aligned}
\mathbb{E}[\hat{\beta}'] &= \mathbb{E}[Dy] \\
&= \mathbb{E}\left[\left((X^\top X)^{-1}X^\top + \Delta\right)(X\beta + \varepsilon)\right] \\
&= \left((X^\top X)^{-1}X^\top + \Delta\right)X\beta + \left((X^\top X)^{-1}X^\top + \Delta\right)\mathbb{E}[\varepsilon] \\
&= \left((X^\top X)^{-1}X^\top + \Delta\right)X\beta \\
&= (X^\top X)^{-1}X^\top X\beta + \Delta X\beta \\
&= \beta + \Delta X\beta.
\end{aligned}$$

Hence, $\hat{\beta}'$ is unbiased iff $\Delta X = 0$.

Proof (continued)

- **Theorem**: LSE is a Best Linear Unbiased Estimator (BLUE):

$$\mathrm{cov}(\hat{\beta}) \preceq \mathrm{cov}(\hat{\beta}') \text{ for any } \hat{\beta}' \text{ s.t. } \hat{\beta}' = D(X)y \text{ and } \mathbb{E}[\hat{\beta}] = \beta$$

Let $\Delta = D - (X^\top X)^{-1}X^\top$. Then β' is unbiased iff $\Delta X = 0$.

$$\begin{aligned}
\text{Cov}(\hat{\beta}') &= \text{Cov}(Dy) \\
&= D\text{Cov}(y)D^T \\
&= \sigma^2 DD^\top \\
&= \sigma^2 \left((X^\top X)^{-1}X^\top + \Delta\right)\left(X(X^\top X)^{-1} + \Delta^\top\right) \\
&= \sigma^2 \left((X^\top X)^{-1}X^\top X(X^\top X)^{-1} + (X^\top X)^{-1}X^\top\Delta^\top + \Delta X(X^\top X)^{-1} + \Delta\Delta^\top\right) \\
&= \sigma^2(X^\top X)^{-1} + \sigma^2(X^\top X)^{-1}(\Delta X)^\top + \sigma^2\Delta X(X^\top X)^{-1} + \sigma^2\Delta\Delta^\top \\
&= \sigma^2(X^\top X)^{-1} + \sigma^2\Delta\Delta^\top \\
&= \text{Cov}(\hat{\beta}) + \sigma^2\Delta\Delta^\top \succeq \text{Cov}(\hat{\beta})
\end{aligned}$$

Implications

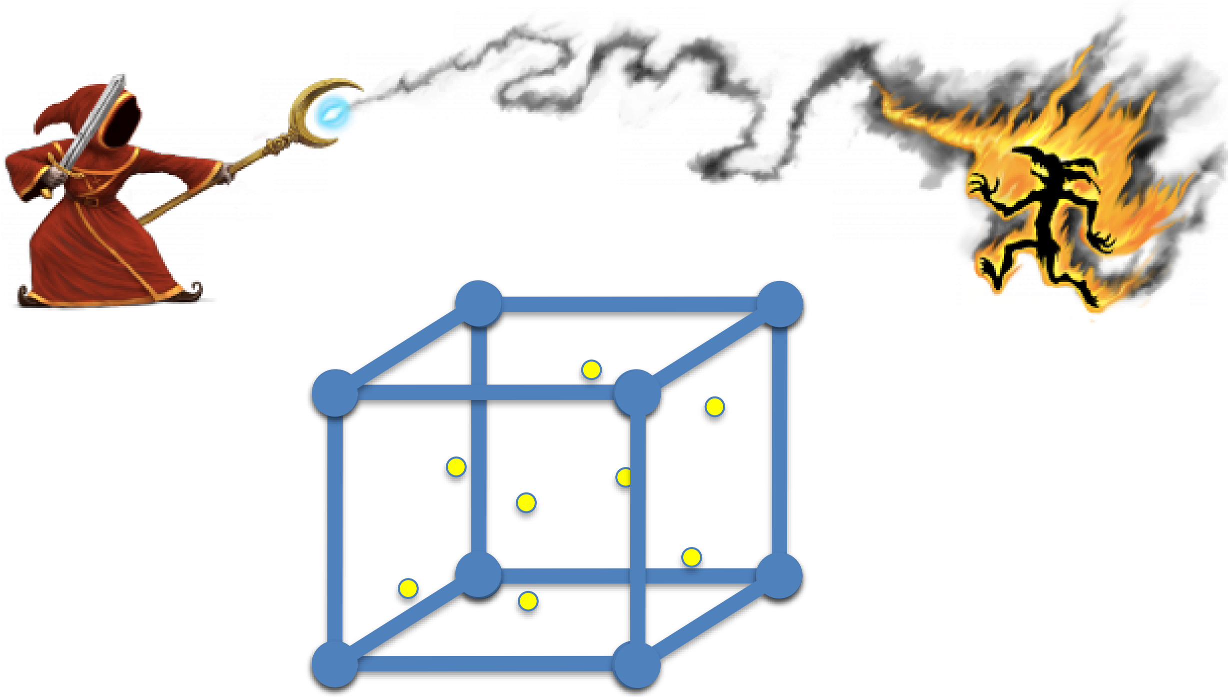
- **Theorem**: LSE is a Best Linear Unbiased Estimator (BLUE):

$$\mathrm{cov}(\hat{\beta}) \preceq \mathrm{cov}(\hat{\beta}') \text{ for any } \hat{\beta}' \text{ s.t. } \hat{\beta}' = D(X)y \text{ and } \mathbb{E}[\hat{\beta}] = \beta$$

$$\text{EPE: } \mathbb{E}[(y_0 - \hat{y}_0)^2] = \sigma^2 + x_0^\top \mathrm{Cov}(\hat{\beta}) x_0$$

In other words, LSE achieves the smallest EPE among all unbiased linear estimators, **in all possible directions!**

What Happened to the Curse?



What Happened to the Curse?

EPE: $\mathbb{E}[(y_0 - \hat{y}_0)^2] = \sigma^2 + x_0^\top \mathsf{Cov}(\hat{\beta}) x_0 = \sigma^2 + \sigma^2 x_0^\top (X^\top X)^{-1} x_0$

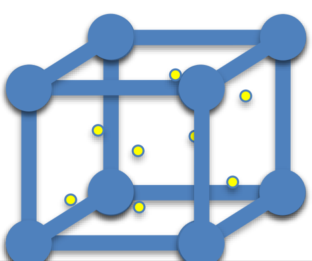
Suppose that $x_i \in \mathbb{R}^d, i = 0, 1, \ldots, n,$ are sampled from some distribution with mean 0 and covariance Σ.

$$\mathbb{E}[x] = 0, \mathbb{E}[xx^\top] = \Sigma$$

Then,

$$\frac{1}{n} X^\top X = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^\top \to \Sigma \quad \text{w.p. 1}$$

by the law of large numbers.



Hence:

$$
\begin{aligned}
\mathbb{E}[\text{EPE}] &= \sigma^2 + \sigma^2 \mathbb{E}\left[x_0^\top \left(X^\top X\right)^{-1} x_0\right] \\
&= \sigma^2 + \sigma^2 \mathbb{E}\left[\mathsf{trace}\left(\left(X^\top X\right)^{-1} x_0 x_0^\top\right)\right] \\
&= \sigma^2 + \sigma^2 \frac{1}{n} \mathbb{E}\left[\mathsf{trace}\left(\left(\frac{1}{n} X^\top X\right)^{-1} x_0 x_0^\top\right)\right] \\
&\approx \sigma^2 + \sigma^2 \frac{1}{n} \mathbb{E}\left[\mathsf{trace}\left(\Sigma^{-1} x_0 x_0^\top\right)\right] \\
&= \sigma^2 + \sigma^2 \frac{1}{n} \mathsf{trace}(\Sigma^{-1} \mathbb{E}[x_0 x_0^\top]) \\
&= \sigma^2 + \sigma^2 \frac{1}{n} \mathsf{trace}(\Sigma^{-1} \Sigma) \\
&= \sigma^2 + \sigma^2 \frac{d}{n}
\end{aligned}
$$

Northeastern

What Happened to the Curse?

EPE: $\mathbb{E}[(y_0 - \hat{y}_0)^2] = \sigma^2 + x_0^\top \text{Cov}(\hat{\beta}) x_0 = \sigma^2 + \sigma^2 x_0^\top (X^\top X)^{-1} x_0$

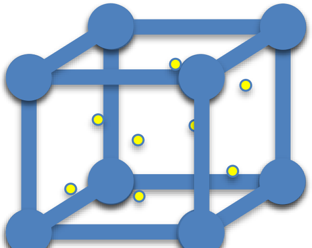
Suppose that $x_i \in \mathbb{R}^d, i = 0, 1, \ldots, n,$ are sampled from some distribution with mean 0 and covariance Σ.

$$\mathbb{E}[x] = 0, \mathbb{E}[xx^\top] = \Sigma$$

Then,

$$\frac{1}{n} X^\top X = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^\top \to \Sigma \quad \text{w.p. } 1$$

by the law of large numbers.

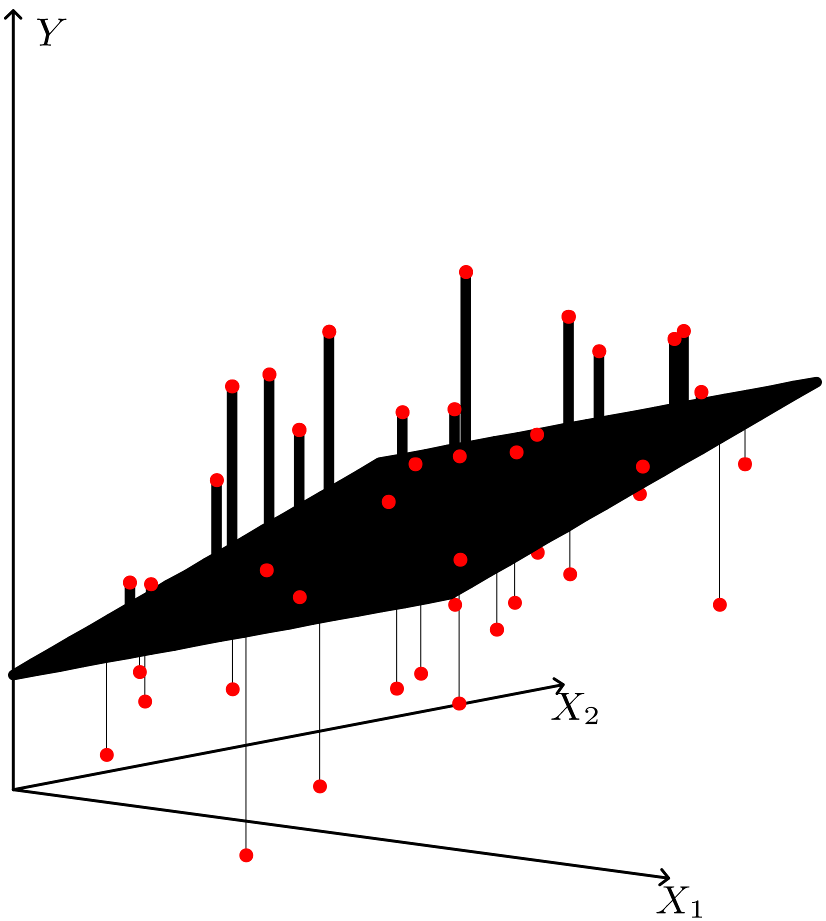


Number of samples: $n \geq \frac{d\sigma^2}{\epsilon}$

$$\mathbb{E}[\text{EPE}] \approx \sigma^2 + \sigma^2 \frac{d}{n}$$

k-NN: $n \geq \left(\frac{1}{\epsilon}\right)^d$

Another Way of Seeing This…



Northeastern