

Maximum Likelihood Estimation

Your first consulting job

- *Billionaire*: I have special coin, if I flip it, what's the probability it will be heads?
- *You*: Please flip it a few times:

- *You*: The probability is:

- *Billionaire*: Why?

Coin – Binomial Distribution

- **Data:** sequence $D = (HHTHT\dots)$, **k heads** out of **n flips**
- **Hypothesis:** $P(\text{Heads}) = \theta$, $P(\text{Tails}) = 1 - \theta$
 - Flips are i.i.d.:
 - Independent events
 - Identically distributed according to Binomial distribution

- $P(\mathcal{D}|\theta) =$

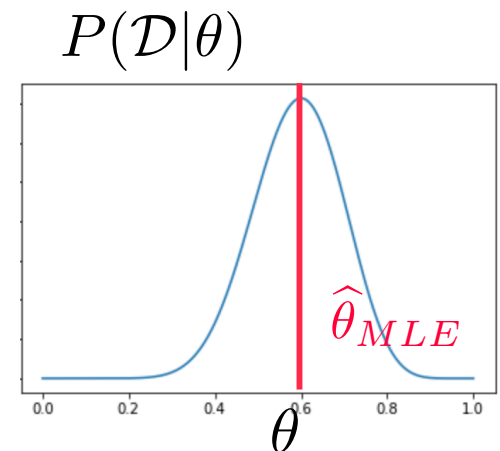
Maximum Likelihood Estimation

- **Data:** sequence $D = (HHTHT\dots)$, **k heads** out of **n flips**
- **Hypothesis:** $P(\text{Heads}) = \theta$, $P(\text{Tails}) = 1 - \theta$

$$P(\mathcal{D}|\theta) = \theta^k (1 - \theta)^{n-k}$$

- Maximum likelihood estimation (MLE): Choose θ that maximizes the probability of observed data:

$$\begin{aligned}\hat{\theta}_{MLE} &= \arg \max_{\theta} P(\mathcal{D}|\theta) \\ &= \arg \max_{\theta} \log P(\mathcal{D}|\theta)\end{aligned}$$



Your first learning algorithm

$$\begin{aligned}\hat{\theta}_{MLE} &= \arg \max_{\theta} \log P(\mathcal{D}|\theta) \\ &= \arg \max_{\theta} \log \theta^k (1 - \theta)^{n-k}\end{aligned}$$

- Set derivative to zero:

$$\frac{d}{d\theta} \log P(\mathcal{D}|\theta) = 0$$

Maximum Likelihood Estimation

Observe X_1, X_2, \dots, X_n drawn IID from $f(x; \theta)$ for some “true” $\theta = \theta_*$

Likelihood function $L_n(\theta) = \prod_{i=1}^n f(X_i; \theta)$

Log-Likelihood function $l_n(\theta) = \log(L_n(\theta)) = \sum_{i=1}^n \log(f(X_i; \theta))$

Maximum Likelihood Estimator (MLE) $\hat{\theta}_{MLE} = \arg \max_{\theta} L_n(\theta)$

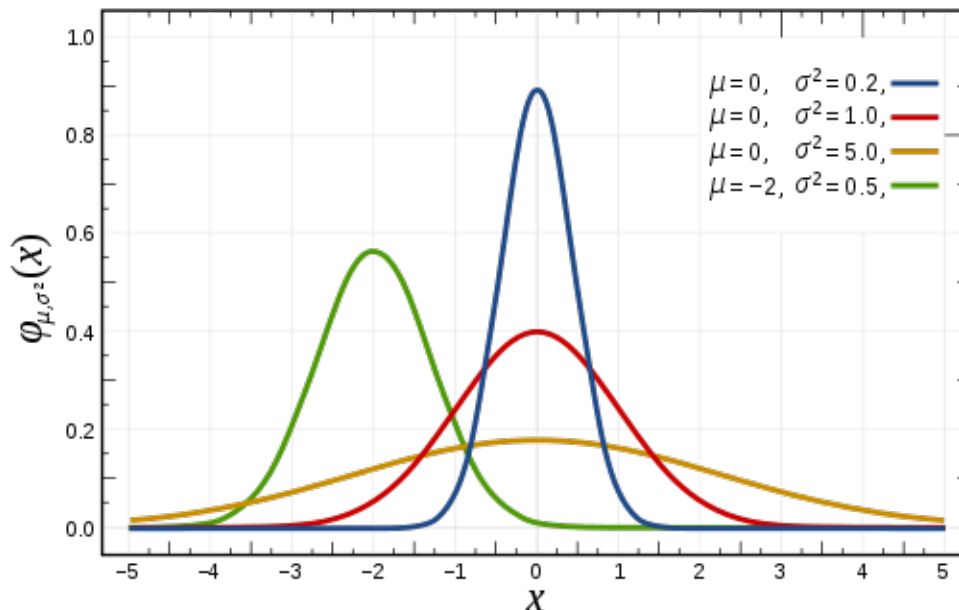
Recap

- Learning is...
 - Collect some data
 - E.g., coin flips
 - Choose a hypothesis class or model
 - E.g., binomial
 - Choose a loss function
 - E.g., data likelihood
 - Choose an optimization procedure
 - E.g., set derivative to zero to obtain MLE

What about continuous variables?

- *Billionaire*: What if I am measuring a **continuous variable**?
- *You*: Let me tell you about **Gaussians**...

$$P(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Some properties of Gaussians

- affine transformation (multiplying by scalar and adding a constant)
 - $X \sim N(\mu, \sigma^2)$
 - $Y = aX + b \rightarrow Y \sim N(a\mu + b, a^2\sigma^2)$
- Sum of Gaussians
 - $X \sim N(\mu_X, \sigma_X^2)$
 - $Y \sim N(\mu_Y, \sigma_Y^2)$
 - $Z = X + Y \rightarrow Z \sim N(\mu_X + \mu_Y, \sigma_X^2 + \sigma_Y^2)$

MLE for Gaussian

- Prob. of i.i.d. samples $D=\{x_1, \dots, x_n\}$ (e.g., temperature):

$$\begin{aligned} P(\mathcal{D}|\mu, \sigma) &= P(x_1, \dots, x_n|\mu, \sigma) \\ &= \left(\frac{1}{\sigma\sqrt{2\pi}} \right)^n \prod_{i=1}^n e^{-\frac{(x_i - \mu)^2}{2\sigma^2}} \end{aligned}$$

- Log-likelihood of data:

$$\log P(\mathcal{D}|\mu, \sigma) = -n \log(\sigma\sqrt{2\pi}) - \sum_{i=1}^n \frac{(x_i - \mu)^2}{2\sigma^2}$$

- What is $\hat{\theta}_{MLE}$ for $\theta = (\mu, \sigma^2)$? Draw a picture!

Your second learning algorithm: MLE for mean of a Gaussian

- What's MLE for mean?

$$\frac{d}{d\mu} \log P(\mathcal{D}|\mu, \sigma) = \frac{d}{d\mu} \left[-n \log(\sigma \sqrt{2\pi}) - \sum_{i=1}^n \frac{(x_i - \mu)^2}{2\sigma^2} \right]$$

MLE for variance

- Again, set derivative to zero:

$$\frac{d}{d\sigma} \log P(\mathcal{D}|\mu, \sigma) = \frac{d}{d\sigma} \left[-n \log(\sigma \sqrt{2\pi}) - \sum_{i=1}^n \frac{(x_i - \mu)^2}{2\sigma^2} \right]$$

Learning Gaussian parameters

- MLE:

$$\hat{\mu}_{MLE} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\hat{\sigma}^2_{MLE} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu}_{MLE})^2$$

- MLE for the variance of a Gaussian is **biased**

$$\mathbb{E}[\hat{\sigma}^2_{MLE}] \neq \sigma^2$$

- Unbiased variance estimator:

$$\hat{\sigma}^2_{unbiased} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \hat{\mu}_{MLE})^2$$

Maximum Likelihood Estimation

Observe X_1, X_2, \dots, X_n drawn IID from $f(x; \theta)$ for some “true” $\theta = \theta_*$

Likelihood function $L_n(\theta) = \prod_{i=1}^n f(X_i; \theta)$

Log-Likelihood function $l_n(\theta) = \log(L_n(\theta)) = \sum_{i=1}^n \log(f(X_i; \theta))$

Maximum Likelihood Estimator (MLE) $\hat{\theta}_{MLE} = \arg \max_{\theta} L_n(\theta)$

Under benign assumptions, as the number of observations $n \rightarrow \infty$ we have $\hat{\theta}_{MLE} \rightarrow \theta_*$

The MLE is a “recipe” that begins with a *model* for data $f(x; \theta)$

Applications preview

Maximum Likelihood Estimation

Observe X_1, X_2, \dots, X_n drawn IID from $f(x; \theta)$ for some “true” $\theta = \theta_*$

Likelihood function $L_n(\theta) = \prod_{i=1}^n f(X_i; \theta)$

Log-Likelihood function $l_n(\theta) = \log(L_n(\theta)) = \sum_{i=1}^n \log(f(X_i; \theta))$

Maximum Likelihood Estimator (MLE) $\hat{\theta}_{MLE} = \arg \max_{\theta} L_n(\theta)$

Under benign assumptions, as the number of observations $n \rightarrow \infty$ we have $\hat{\theta}_{MLE} \rightarrow \theta_*$

Why is it useful to recover the “true” parameters θ_* of a probabilistic model?

- **Estimation** of the parameters θ_* is the goal
- Help **interpret** or summarize large datasets
- Make **predictions** about future data
- **Generate** new data $X \sim f(\cdot; \hat{\theta}_{MLE})$

Estimation

Observe X_1, X_2, \dots, X_n drawn IID from $f(x; \theta)$ for some “true” $\theta = \theta_*$

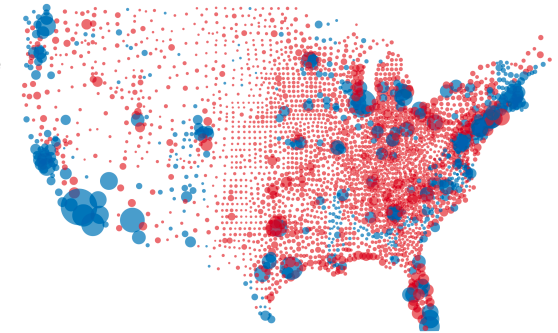
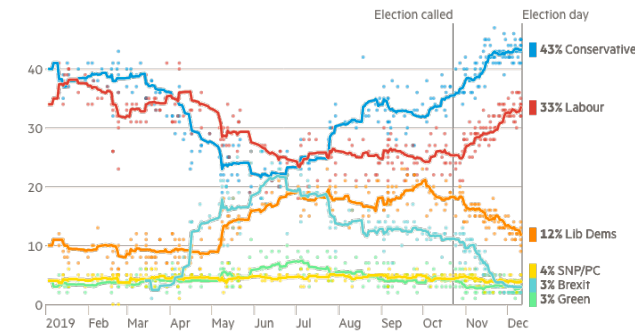
Opinion polls

How does the greater population feel about an issue?
Correct for over-sampling?

- θ_* is “true” average opinion
- X_1, X_2, \dots are sample calls

UK poll tracker

Lines represent weighted averages, points represent polls (%)



A/B testing

How do we figure out which ad results in more click-through?

- θ_* are the “true” average rates
- X_1, X_2, \dots are binary “clicks”

Save on prescription drugs - over \$3,637* a year!

Control

Explore Humana's Medicare plans

Treatment

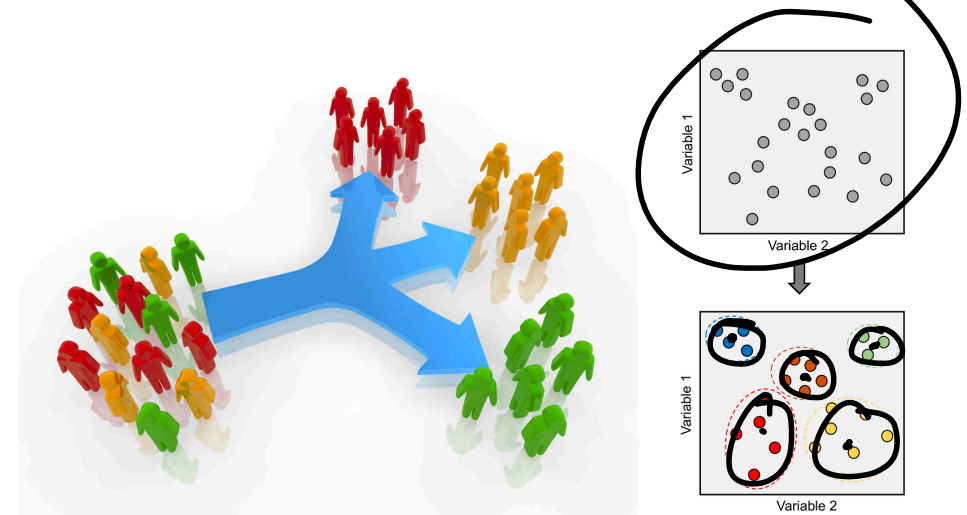
Interpret

Observe X_1, X_2, \dots, X_n drawn IID from $f(x; \theta)$ for some “true” $\theta \equiv \theta_*$

Customer segmentation / clustering

Can we identify distinct groups of customers by their behavior?

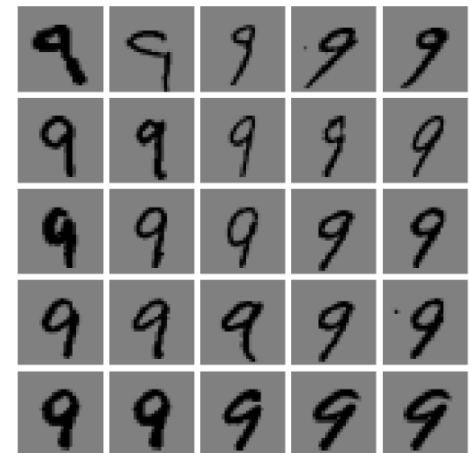
- θ_* describes “center” of distinct groups
- X_1, X_2, \dots are individual customers



Data exploration

What are the degrees of freedom of the dataset?

- θ_* describes the principle directions of variation
- X_1, X_2, \dots are the individual images



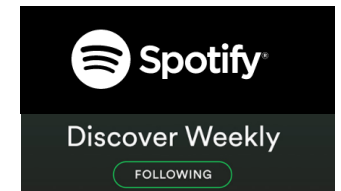
Predict

Observe X_1, X_2, \dots, X_n drawn IID from $f(x; \theta)$ for some “true” $\theta = \theta_*$

Content recommendation

Can we predict how much someone will like a movie based on past ratings?

- θ_* describes user’s preferences
- X_1, X_2, \dots are (movie, rating) pairs



Object recognition / classification

Identify a flower given just its picture?

- θ_* describes the characteristics of each kind of flower
- X_1, X_2, \dots are the (image, label) pairs



(a)



(b)



(c)

Figure 1.1: Three types of Iris flowers: Setosa, Versicolor and Virginica. Used with kind permission of Dennis Krumb and SIGNA.

index	sl	sw	pl	pw	label
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
...					
50	7.0	3.2	4.7	1.4	Versicolor
...					
149	5.9	3.0	5.1	1.8	Virginica

Generate

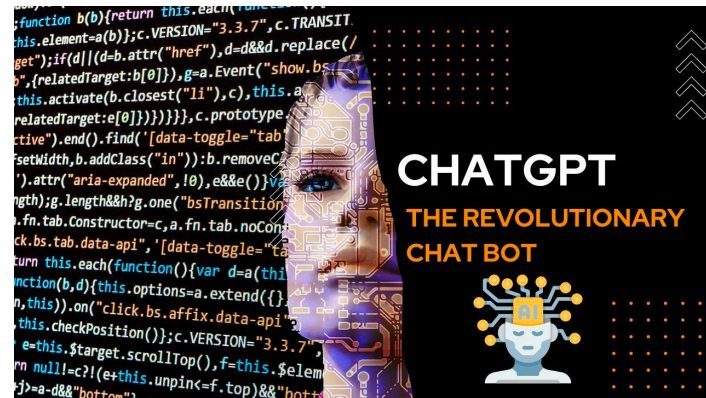
Observe X_1, X_2, \dots, X_n drawn IID from $f(x; \theta)$ for some “true” $\theta = \theta_*$

Text generation

Can AI generate text that could have been written like a human?

- θ_* describes language structure
- X_1, X_2, \dots are text snippets found online

“Kaia the dog wasn't a natural pick to go to mars.
No one could have predicted she would...”



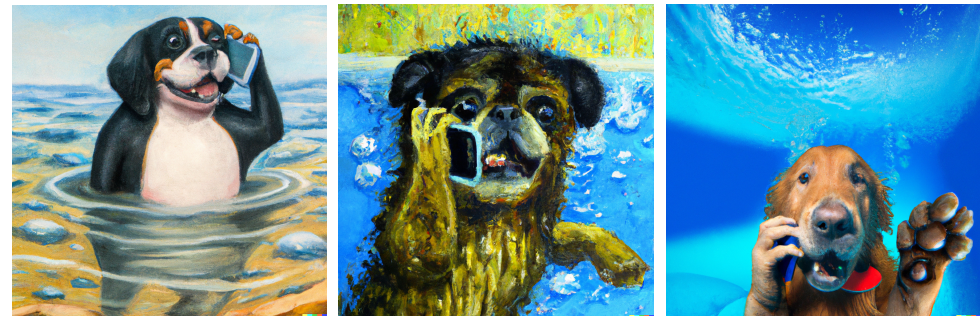
<https://chat.openai.com/chat>

Image to text generation

Can AI generate an image from a prompt?

- θ_* describes the coupled structure of images and text
- X_1, X_2, \dots are the (image, caption) pairs found online

“dog talking on cell phone under water, oil painting”



<https://labs.openai.com/>

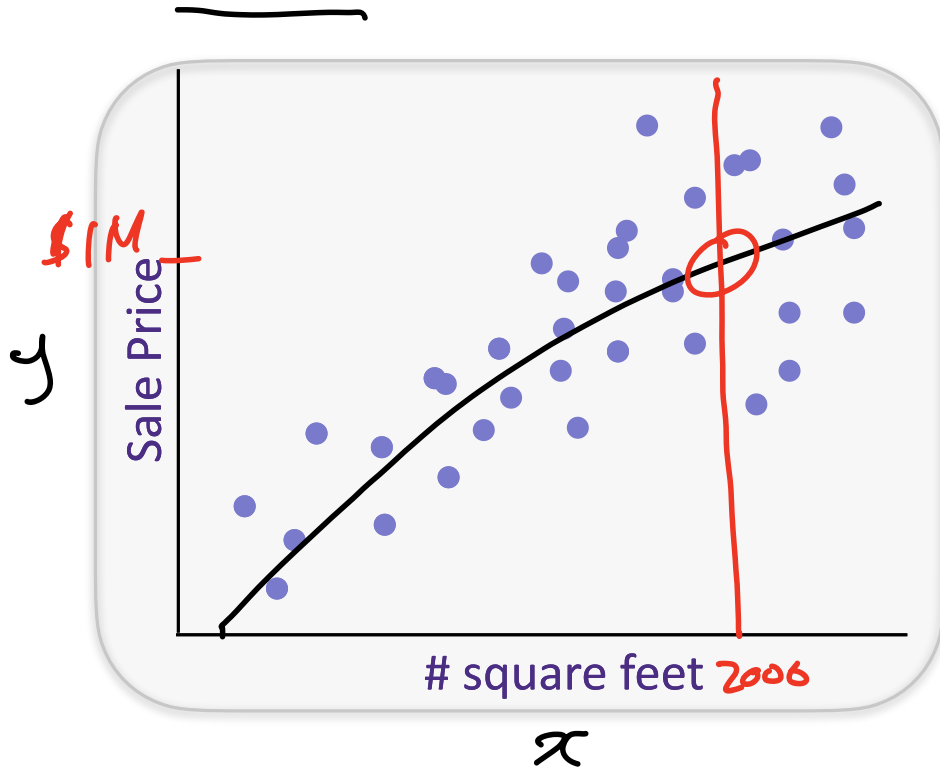
Linear Regression

The regression problem, 1-dimensional

Given past sales data on zillow.com, predict:

$y =$ House sale price from

$x =$ {# sq. ft.}



Training Data:

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

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

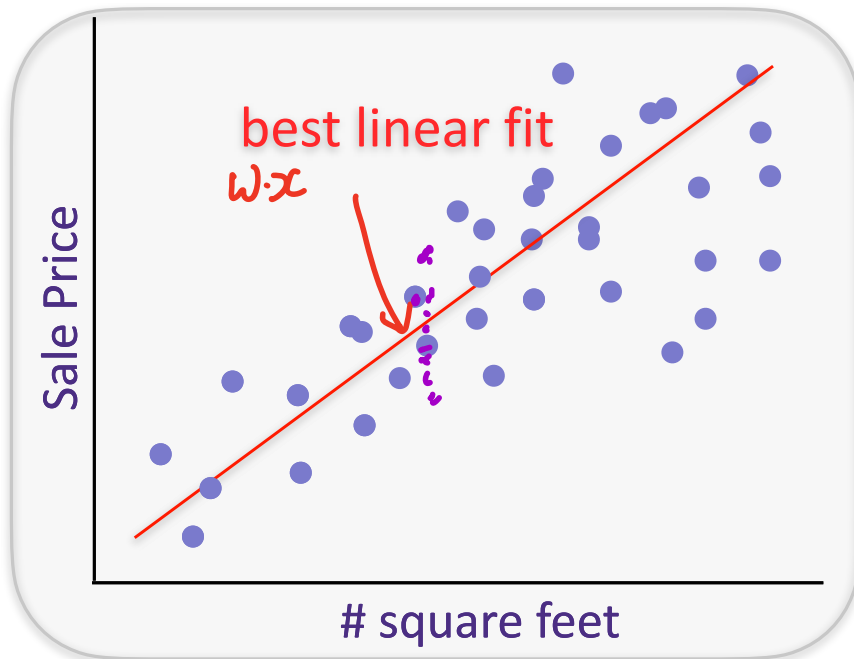
$$y_i \in \mathbb{R}$$

Fit a function to our data, 1-d

Given past sales data on [zillow.com](https://www.zillow.com), predict:

$y =$ House sale price from

$x = \{\# \text{ sq. ft.}\}$



Training Data: $x_i \in \mathbb{R}$
 $y_i \in \mathbb{R}$
 $\{(x_i, y_i)\}_{i=1}^n$

Hypothesis/Model: linear

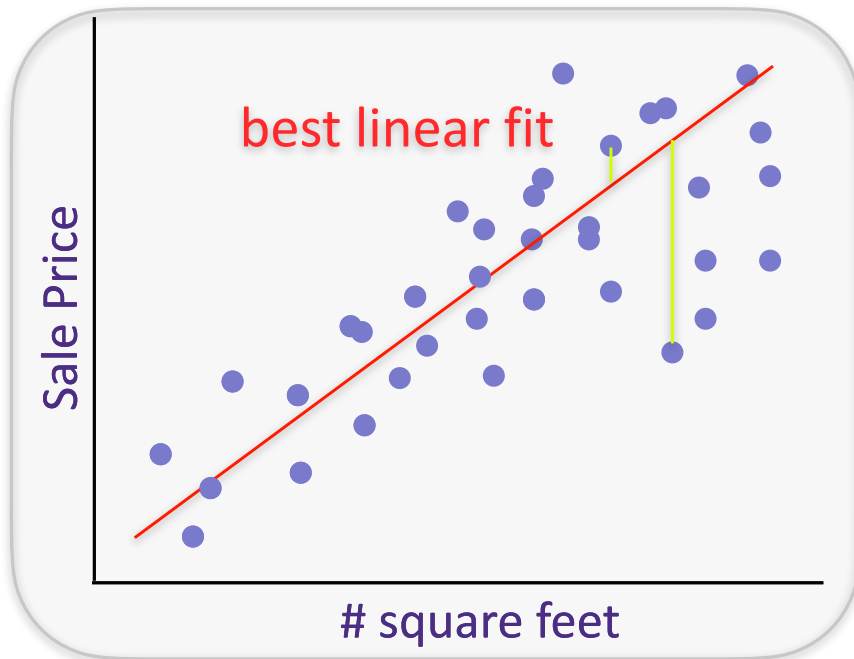
$$y_i = \underbrace{x_i w}_{w \in \mathbb{R}} + \epsilon_i \quad \epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$$

Fit a function to our data, 1-d

Given past sales data on [zillow.com](https://www.zillow.com), predict:

$y =$ House sale price from

$x = \{\# \text{ sq. ft.}\}$



Training Data: $x_i \in \mathbb{R}$
 $y_i \in \mathbb{R}$
 $\{(x_i, y_i)\}_{i=1}^n$

Hypothesis/Model: linear

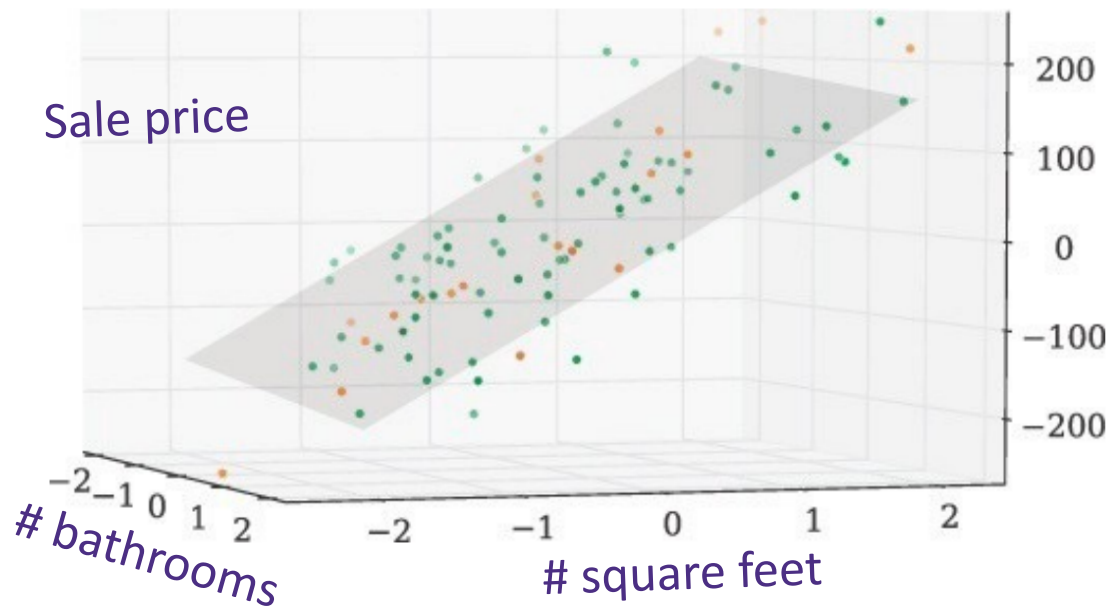
$$y_i = x_i w + \epsilon_i \quad \epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$$

The regression problem, d-dim

Given past sales data on [zillow.com](https://www.zillow.com), predict:

$y =$ House sale price *from*

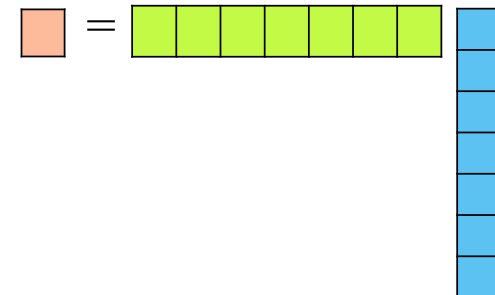
$x =$ {# sq. ft., zip code, date of sale, etc.}



Training Data: $x_i \in \mathbb{R}^d$
 $y_i \in \mathbb{R}$
 $\{(x_i, y_i)\}_{i=1}^n$

Hypothesis/Model: linear

$$y_i = x_i^T w + \epsilon_i \quad \epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$$



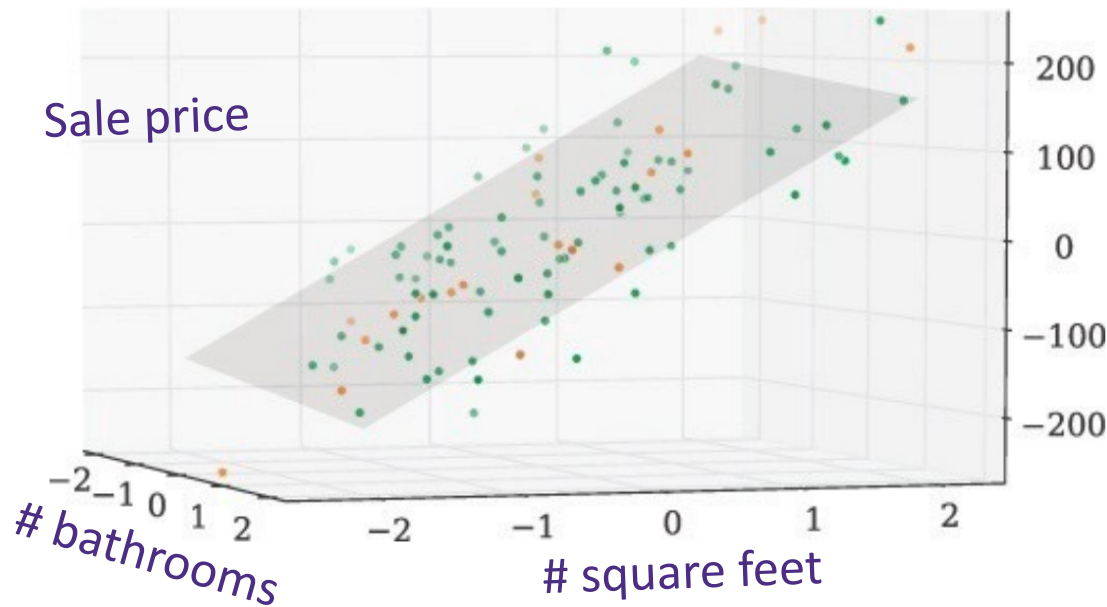
The regression problem, d-dim

Given past sales data on [zillow.com](https://www.zillow.com), predict:

$y =$ House sale price from

$x = \{\# \text{ sq. ft.}, \text{ zip code}, \text{ date of sale}, \text{ etc.}\}$

$$(y_i - x_i^T w) = \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$



Training Data: $x_i \in \mathbb{R}^d$
 $y_i \in \mathbb{R}$
 $\{(x_i, y_i)\}_{i=1}^n$

Hypothesis/Model: linear

$$y_i = x_i^T w + \epsilon_i \quad \epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$$

$$p(y|x, w, \sigma) =$$

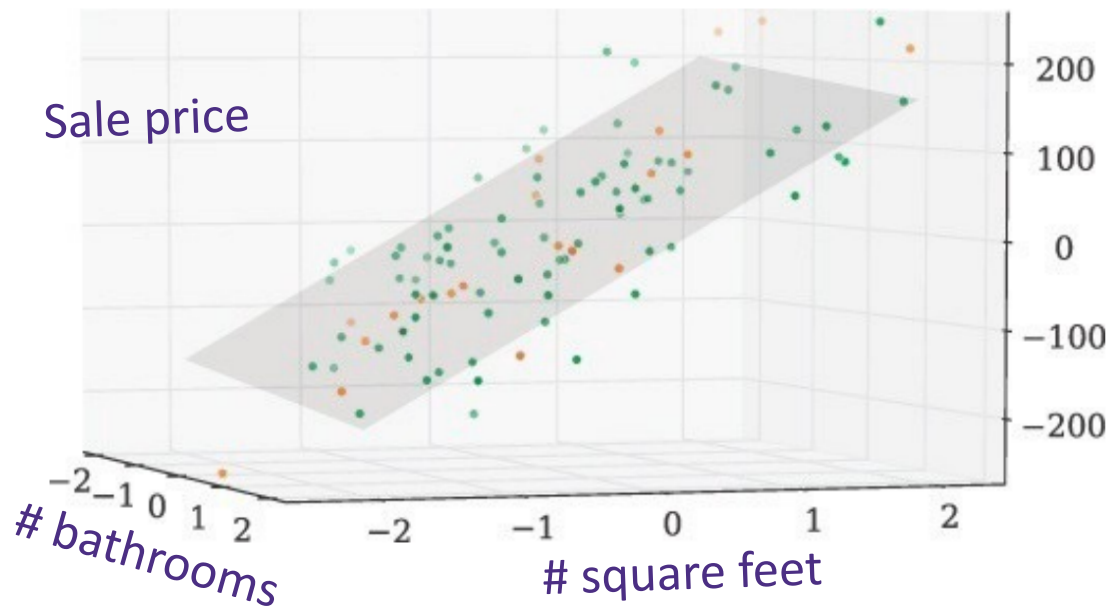
$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - x_i^T w)^2}{2\sigma^2}\right)$$

The regression problem, d-dim

Given past sales data on [zillow.com](https://www.zillow.com), predict:

$y =$ House sale price *from*

$x =$ {# sq. ft., zip code, date of sale, etc.}



Training Data: $x_i \in \mathbb{R}^d$
 $y_i \in \mathbb{R}$
 $\{(x_i, y_i)\}_{i=1}^n$

Hypothesis/Model: linear

$$y_i = x_i^T w + \epsilon_i \quad \epsilon_i \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$$

$$p(y|x, w, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y-x^T w)^2/2\sigma^2}$$

Maximizing log-likelihood

Training Data: $x_i \in \mathbb{R}^d$
 $\{(x_i, y_i)\}_{i=1}^n$ $y_i \in \mathbb{R}$

$$p(y|x, w, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y-x^\top w)^2/2\sigma^2}$$

IID

Likelihood: $P(\mathcal{D}|w, \sigma) = \prod_{i=1}^n p(y_i|x_i, w, \sigma) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i-x_i^\top w)^2/2\sigma^2}$

Maximum Likelihood Estimation

Observe X_1, X_2, \dots, X_n drawn IID from $f(x; \theta)$ for some “true” $\theta = \theta_*$

Likelihood function $L_n(\theta) = \prod_{i=1}^n f(X_i; \theta)$

Log-Likelihood function $l_n(\theta) = \log(L_n(\theta)) = \sum_{i=1}^n \log(f(X_i; \theta))$

Maximum Likelihood Estimator (MLE) $\hat{\theta}_{MLE} = \arg \max_{\theta} L_n(\theta)$

Under benign assumptions, as the number of observations $n \rightarrow \infty$ we have $\hat{\theta}_{MLE} \rightarrow \theta_*$

Why is it useful to recover the “true” parameters θ_* of a probabilistic model?

- **Estimation** of the parameters θ_* is the goal
- Help **interpret** or summarize large datasets
- Make **predictions** about future data
- **Generate** new data $X \sim f(\cdot; \hat{\theta}_{MLE})$

Maximizing log-likelihood

$$\log(\exp(x)) = x$$

Training Data:

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

σ - known

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

$$y_i \in \mathbb{R}$$

$$p(y|x, w, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y-x^\top w)^2/2\sigma^2}$$

Likelihood:
$$P(\mathcal{D}|w, \sigma) = \prod_{i=1}^n p(y_i|x_i, w, \sigma) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i-x_i^\top w)^2/2\sigma^2}$$

Maximize (wrt w):
$$\begin{aligned} \log P(\mathcal{D}|w, \sigma) &= \log \left(\prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i-x_i^\top w)^2/2\sigma^2} \right) \\ &= \sum_{i=1}^n \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{(y_i-x_i^\top w)^2}{2\sigma^2} \right) \right) \\ &= n \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \sum_{i=1}^n \frac{(y_i-x_i^\top w)^2}{2\sigma^2} \end{aligned}$$

$$\operatorname{argmax}_w \log P(\mathcal{D}|w, \sigma) = \operatorname{argmax}_w - \sum_{i=1}^n \frac{(y_i-x_i^\top w)^2}{2\sigma^2}$$

$$= \operatorname{argmax}_w - \sum_i (y_i-x_i^\top w)^2$$

$$= \underset{w}{\operatorname{argmin}} \sum_{i=1}^n (y_i - x_i^T w)^2$$

$$f(w) = (y_i - x_i^T w)^2 = y_i^2 - 2x_i^T w + (w^T x_i)^2$$

$$= y_i^2 - 2w^T x_i + w^T x_i x_i^T w$$

$$\nabla_w \log P(D|w, \sigma) = - \sum_{i=1}^n \frac{(y_i - x_i^T w)}{\sigma^2} \cdot (-x_i) = -\frac{1}{\sigma^2} \sum_{i=1}^n y_i x_i + \frac{1}{\sigma^2} \sum_{i=1}^n x_i x_i^T w$$

$$\nabla_w \log P(D|w, \sigma) = -\frac{1}{\sigma^2} \sum_{i=1}^n y_i x_i + \frac{1}{\sigma^2} \sum_{i=1}^n x_i x_i^T w$$

Set = 0 solve for w:

$$\sum_{i=1}^n y_i x_i = \sum_{i=1}^n x_i x_i^T w$$

$$= \left(\sum_{i=1}^n x_i x_i^T \right) w$$

\hat{w}_{MLE} is any w that satisfies

Suppose $\left(\sum_{i=1}^n x_i x_i^T \right)^{-1}$ exists

$$\begin{matrix} \boxed{} \\ \boxed{d \times 1} \cdot \boxed{1 \times d} \end{matrix} = d \times d$$

$$\hat{w}_{MLE} = \left(\sum_{i=1}^n x_i x_i^T \right)^{-1} \sum_{i=1}^n x_i y_i$$

$$f(w) = (y - x^T w)^2$$

$$\begin{aligned}\nabla_w f(w) &= 2(y - x^T w) \cdot \underbrace{\nabla_w (y - x^T w)}_{= \nabla_w (y - w^T x)} \\ &= -x\end{aligned}$$

$$= 2(y - x^T w)(-x)$$

$$= -(yx - xx^T w)$$

Maximizing log-likelihood

Training Data: $x_i \in \mathbb{R}^d$
 $y_i \in \mathbb{R}$
 $\{(x_i, y_i)\}_{i=1}^n$

$$p(y|x, w, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y-x^\top w)^2/2\sigma^2}$$

Likelihood: $P(\mathcal{D}|w, \sigma) = \prod_{i=1}^n p(y_i|x_i, w, \sigma) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i-x_i^\top w)^2/2\sigma^2}$

Maximize (wrt w): $\log P(\mathcal{D}|w, \sigma) = \log \left(\prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y_i-x_i^\top w)^2/2\sigma^2} \right)$

$$\hat{w}_{MLE} = \arg \min_w \sum_{i=1}^n (y_i - x_i^\top w)^2$$

Maximizing log-likelihood

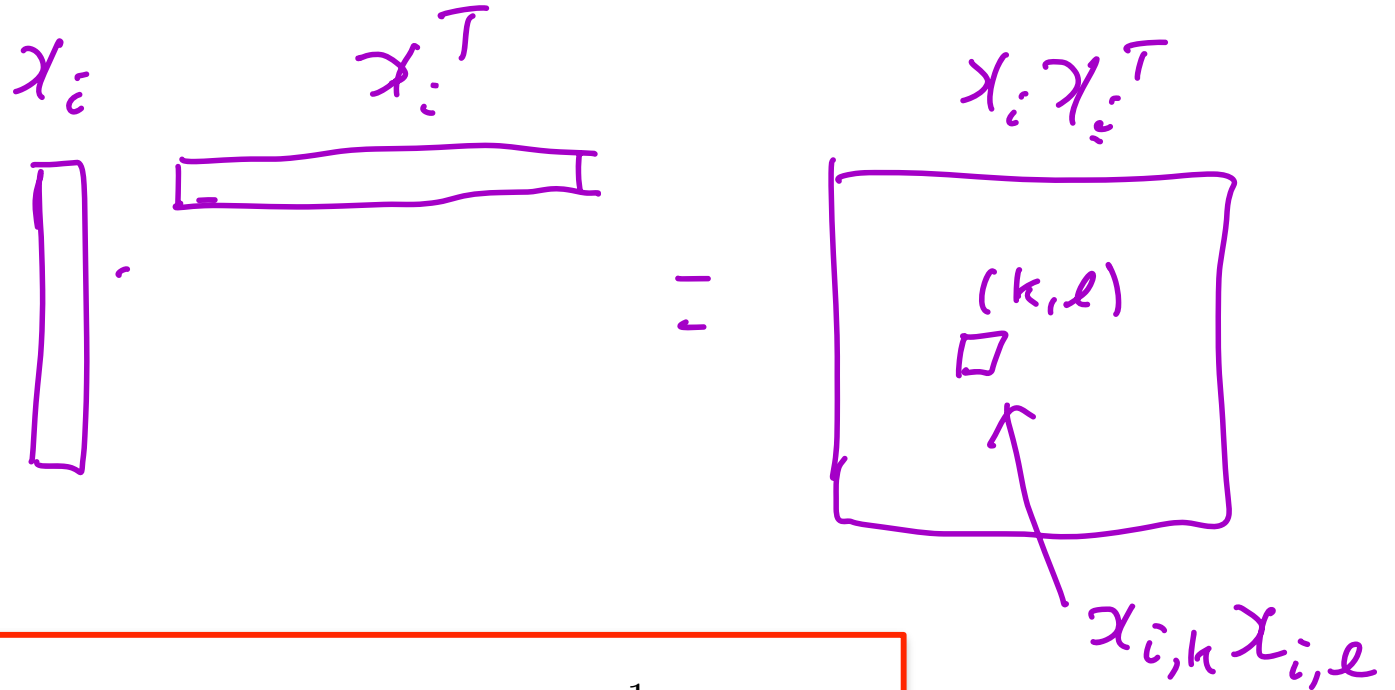
$$\hat{w}_{MLE} = \arg \min_w \sum_{i=1}^n (y_i - x_i^\top w)^2$$

Set derivate=0, solve for w

Maximizing log-likelihood

$$\hat{w}_{MLE} = \arg \min_w \sum_{i=1}^n (y_i - x_i^\top w)^2$$

Set derivative=0, solve for w



$$\hat{w}_{MLE} = \left(\sum_{i=1}^n x_i x_i^\top \right)^{-1} \sum_{i=1}^n x_i y_i$$

The regression problem in matrix notation

$$\hat{w}_{MLE} = \arg \min_w \sum_{i=1}^n (y_i - x_i^\top w)^2$$

$$\mathbb{R}^n \rightarrow y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$$

$\mathbb{R}^{n \times d}$

d : # of features

n : # of examples/datapoints

The regression problem in matrix notation

$$\hat{w}_{MLE} = \arg \min_w \sum_{i=1}^n (y_i - x_i^\top w)^2$$

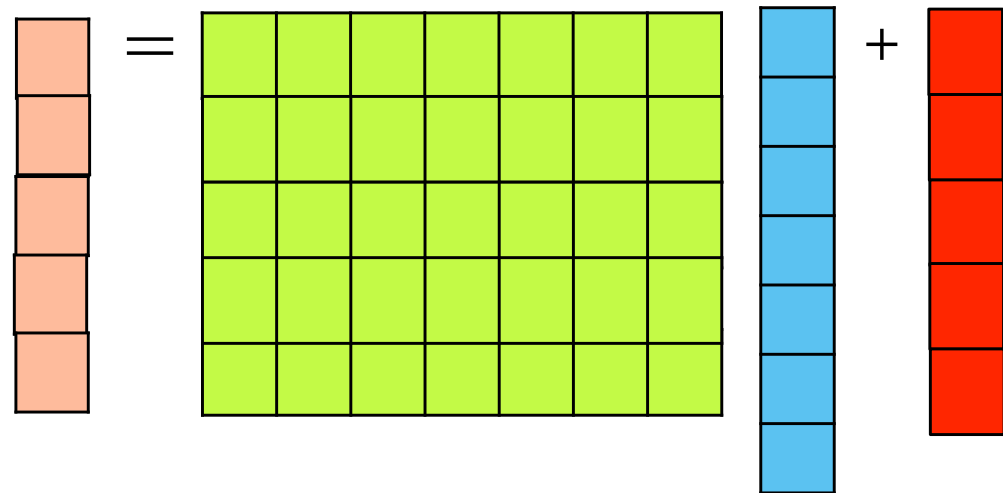
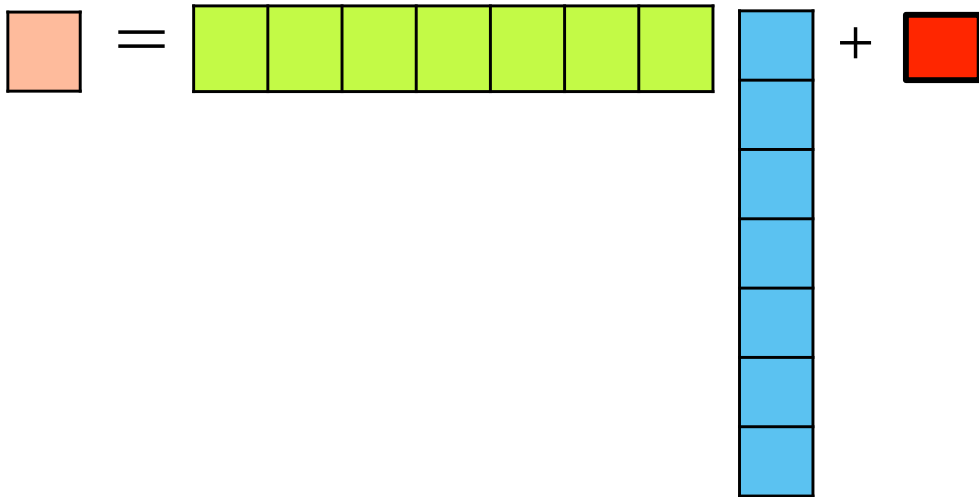
$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{bmatrix}$$

d : # of features

n : # of examples/datapoints

$$y_i = x_i^\top w + \epsilon_i$$

$$\mathbf{y} = \mathbf{X}w + \epsilon$$



The regression problem in matrix notation

$$\hat{w}_{MLE} = \arg \min_w \sum_{i=1}^n (y_i - x_i^T w)^2$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$$

$$y_i = x_i^T w + \epsilon_i$$

$$f(w) = w^T B y$$

$$\nabla_w f(w) = B y$$

$$g(w) = y^T A w = w^T A^T y$$

d : # of features

n : # of examples/datapoints

$$\nabla_w g(w) = A^T y$$

$$\mathbf{y} = \mathbf{X}w + \epsilon$$

$$\nabla_w f(w) = 0 \quad f(w)$$

$$\begin{aligned} \hat{w}_{LS} &= \arg \min_w \|\mathbf{y} - \mathbf{X}w\|_2^2 \\ &= \arg \min_w \underbrace{(\mathbf{y} - \mathbf{X}w)^T}_{g(w)} \underbrace{(\mathbf{y} - \mathbf{X}w)}_{h(w)} \leftarrow \\ &= \end{aligned}$$

$$\ell_2 \text{ norm: } \|z\|_2 = \sqrt{\sum_{i=1}^n z_i^2} = \sqrt{\underline{z^T z}}$$

$$-X^T(y - Xw) + -X^T(y - Xw) = -2X^T(y - Xw)$$

$$= -2X^T y + 2X^T X \omega = 0$$

if $(X^T X)^{-1}$ exists then

$$\hat{\omega}_{MLE} = (X^T X)^{-1} X^T y$$

$$f(\omega) = g(\omega) h(\omega)$$

$$f'(\omega) = g'(\omega) h(\omega) + g(\omega) h'(\omega)$$

$$[\nabla_{\omega} f(\omega)]_i = \frac{\partial f(\omega)}{\partial \omega_i}$$

$$\underline{f(\omega)} = \omega^T B \omega$$

$$= \sum_{i,j=1}^n \omega_i \omega_j B_{i,j}$$

$$\frac{\partial f(\omega)}{\partial \omega_k} = \frac{\partial}{\partial \omega_k} \sum_{i,j} \omega_i \omega_j B_{i,j} =$$

$$\begin{cases} 0 & i \neq j \neq k \\ 2\omega_k B_{k,k} & i = j = k \\ \sum_{i \neq k} \omega_j B_{k,i} & i = k \neq j \\ \sum_{i \neq k} \omega_i B_{i,k} & i \neq k = j \end{cases}$$

$$\nabla_{\omega} f(\omega) = (B + B^T) \omega$$

The regression problem in matrix notation

$$\hat{w}_{MLE} = \arg \min_w \sum_{i=1}^n (y_i - x_i^\top w)^2$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{bmatrix}$$

d : # of features

n : # of examples/datapoints

$$y_i = x_i^\top w + \epsilon_i$$

$$\mathbf{y} = \mathbf{X}w + \epsilon$$

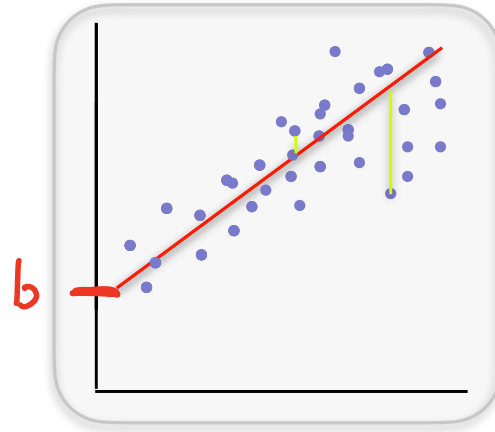
$$\begin{aligned} \hat{w}_{LS} &= \arg \min_w \|\mathbf{y} - \mathbf{X}w\|_2^2 \\ &= \arg \min_w (\mathbf{y} - \mathbf{X}w)^\top (\mathbf{y} - \mathbf{X}w) \end{aligned}$$

$$\ell_2 \text{ norm: } \|z\|_2 = \sqrt{\sum_{i=1}^n z_i^2} = \sqrt{z^\top z}$$

$$\hat{w}_{LS} = \hat{w}_{MLE} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$$

The regression problem in matrix notation

$$\begin{aligned}\hat{w}_{LS} &= \arg \min_w \|\mathbf{y} - \mathbf{X}w\|_2^2 \\ &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}\end{aligned}$$



What about an offset?

$$\begin{aligned}\hat{w}_{LS}, \hat{b}_{LS} &= \arg \min_{w,b} \sum_{i=1}^n (y_i - (x_i^T w + b))^2 \\ &= \arg \min_{w,b} \|\mathbf{y} - (\mathbf{X}w + \mathbf{1}b)\|_2^2\end{aligned}$$

Dealing with an offset

$$\hat{w}_{LS}, \hat{b}_{LS} = \arg \min_{w, b} \|\mathbf{y} - (\mathbf{X}w + \mathbf{1}b)\|_2^2$$

Dealing with an offset

$$\hat{w}_{LS}, \hat{b}_{LS} = \arg \min_{w, b} \|\mathbf{y} - (\mathbf{X}w + \mathbf{1}b)\|_2^2$$

$$\mathbf{X}^T \mathbf{X} \hat{w}_{LS} + \hat{b}_{LS} \mathbf{X}^T \mathbf{1} = \mathbf{X}^T \mathbf{y}$$

$$\mathbf{1}^T \mathbf{X} \hat{w}_{LS} + \hat{b}_{LS} \mathbf{1}^T \mathbf{1} = \mathbf{1}^T \mathbf{y}$$

If $\mathbf{X}^T \mathbf{1} = 0$ (i.e., if each feature is mean-zero) then

$$\hat{w}_{LS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

$$\hat{b}_{LS} = \frac{1}{n} \sum_{i=1}^n y_i$$

Make Predictions

$$\hat{\mathbf{w}}_{LS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

$$\hat{b}_{LS} = \frac{1}{n} \sum_{i=1}^n y_i$$

A new house is about to be listed. What should it sell for?

$$\hat{y}_{\text{new}} = x_{\text{new}}^T \hat{\mathbf{w}}_{LS} + \hat{b}_{LS}$$

Process

Decide on a **model** for the likelihood function $f(x; \theta)$

Find the function which fits the data best

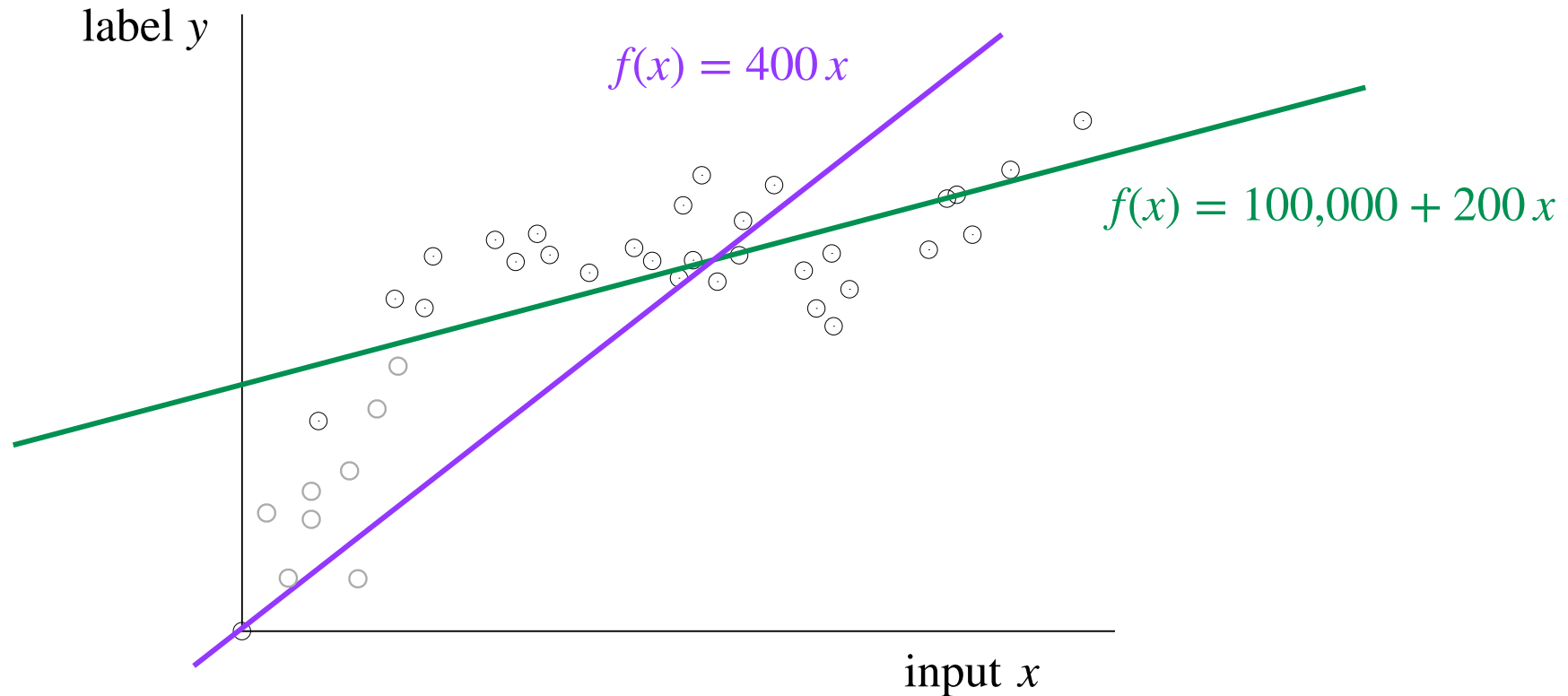
Choose a loss function- least squares

Pick the function which minimizes loss on data

Use function to make prediction on new examples

Linear regression with non-linear basis functions

Recap: Linear Regression



- In general high-dimensions, we fit a linear model with intercept $y_i \simeq w^T x_i + b$, or equivalently $y_i = w^T x_i + b + \epsilon_i$ with model parameters $(w \in \mathbb{R}^d, b \in \mathbb{R})$ that minimizes ℓ_2 -loss

$$\mathcal{L}(w, b) = \sum_{i=1}^n \underbrace{(y_i - (w^T x_i + b))^2}_{\text{error } \epsilon_i}$$

Recap: Linear Regression

- The least squares solution, i.e. the minimizer of the ℓ_2 -loss can be written in a **closed form** as a function of data \mathbf{X} and \mathbf{y} as

or equivalently using straightforward linear algebra by setting the gradient to zero:

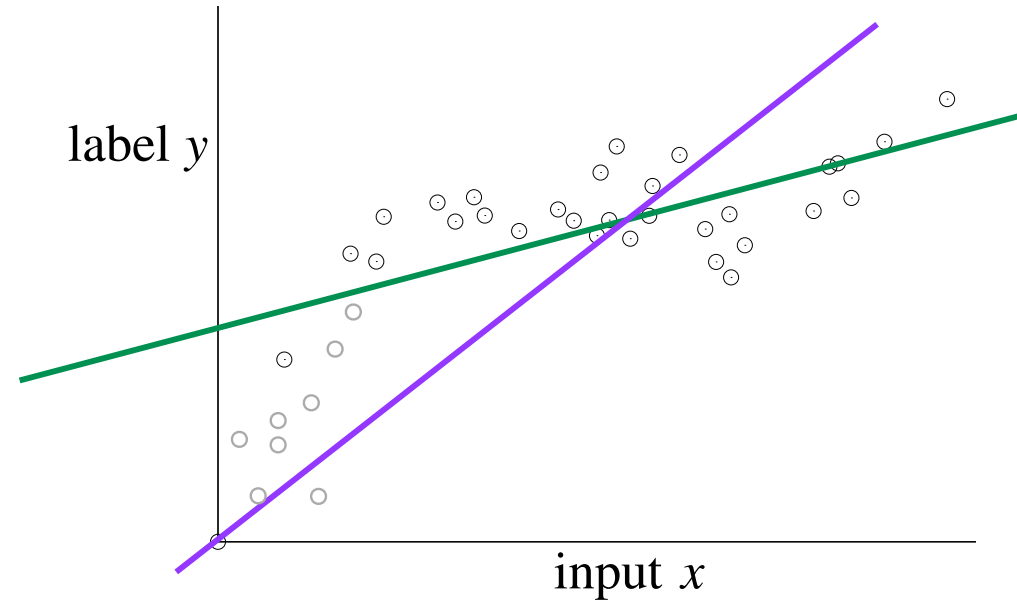
$$\begin{bmatrix} \hat{\mathbf{w}}_{\text{LS}} \\ \hat{b}_{\text{LS}} \end{bmatrix} = \left(\begin{bmatrix} \mathbf{X}^T \\ \mathbf{1}^T \end{bmatrix} [\mathbf{X} \ \mathbf{1}] \right)^{-1} \begin{bmatrix} \mathbf{X}^T \\ \mathbf{1}^T \end{bmatrix} \mathbf{y}$$

Quadratic regression in 1-dimension

- **Data:** $\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

- **Linear model with parameter (b, w_1) :**

- $\hat{y}_i = \underline{b} + \underline{w_1 x_i}$



Quadratic regression in 1-dimension

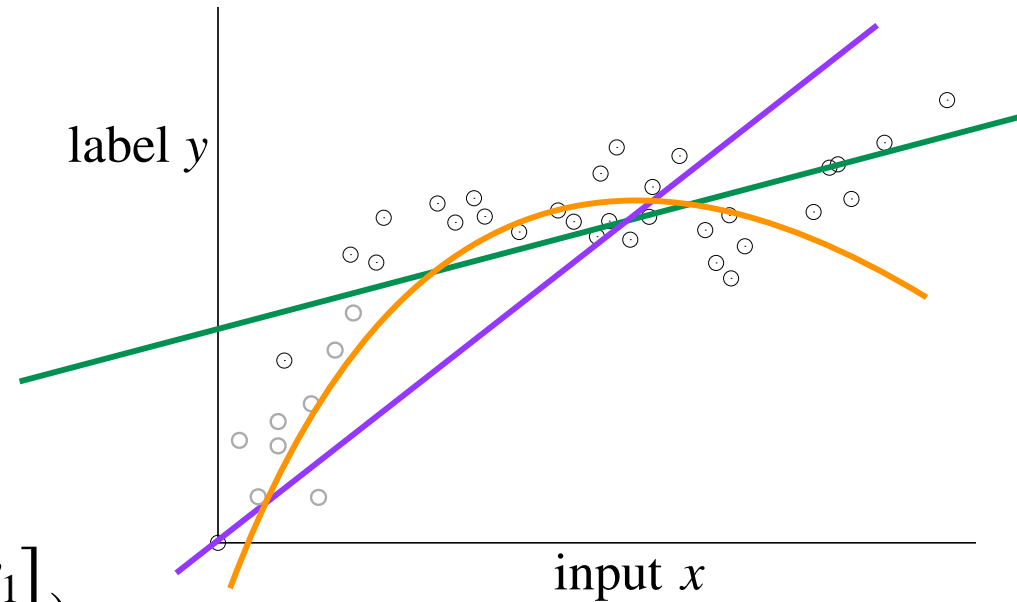
- **Data:** $\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

- **Linear model with parameter (b, w_1) :**

- $\hat{y}_i = \underline{b} + \underline{w_1 x_i}$

- **Quadratic model with parameter $(b, w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix})$:**

- $\hat{y}_i = \underline{b} + \underline{w_1 x_i} + \underline{w_2 x_i^2}$



Quadratic regression in 1-dimension

- Data: $\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

- Linear model with parameter (b, w_1) :

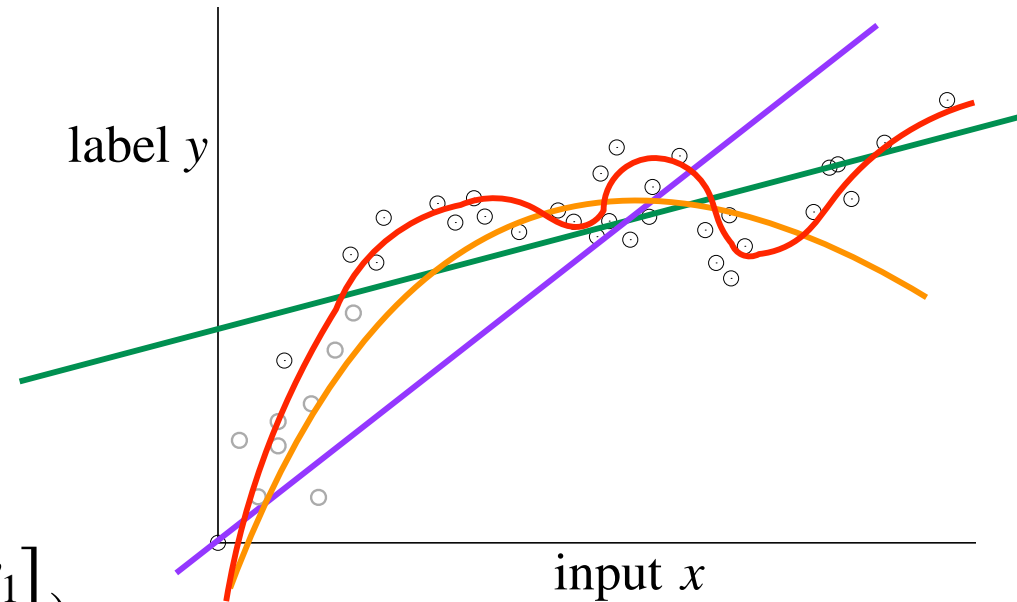
- $\hat{y}_i = b + w_1 x_i$

- Quadratic model with parameter $(b, w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix})$:

- $\hat{y}_i = b + w_1 x_i + w_2 x_i^2$

- Degree- p polynomial model with parameter $(b, w = \begin{bmatrix} w_1 \\ \vdots \\ w_p \end{bmatrix})$:

- $\hat{y}_i = b + w_1 x_i + w_2 x_i^2 + \dots + w_p x_i^p$



Quadratic regression in 1-dimension

- **Data:** $\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

- **Linear model with parameter (b, w_1) :**

- $\hat{y}_i = b + w_1 x_i$

- **Quadratic model with parameter $(b, w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix})$:**

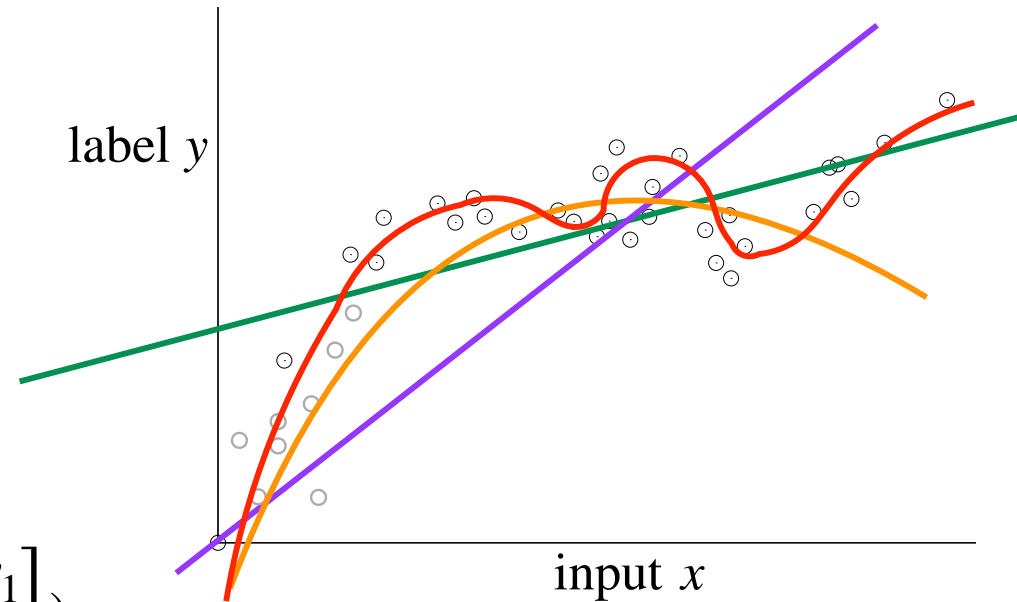
- $\hat{y}_i = b + w_1 x_i + w_2 x_i^2$

- **Degree-p polynomial model with parameter $(b, w = \begin{bmatrix} w_1 \\ \vdots \\ w_p \end{bmatrix})$:**

- $\hat{y}_i = b + w_1 x_i + w_2 x_i^2 + \dots + w_p x_i^p$

- **General p-features with parameter $w = \begin{bmatrix} w_1 \\ \vdots \\ w_p \end{bmatrix}$:**

- $\hat{y}_i = \langle w, h(x_i) \rangle$ where $h : \mathbb{R} \rightarrow \mathbb{R}^p$



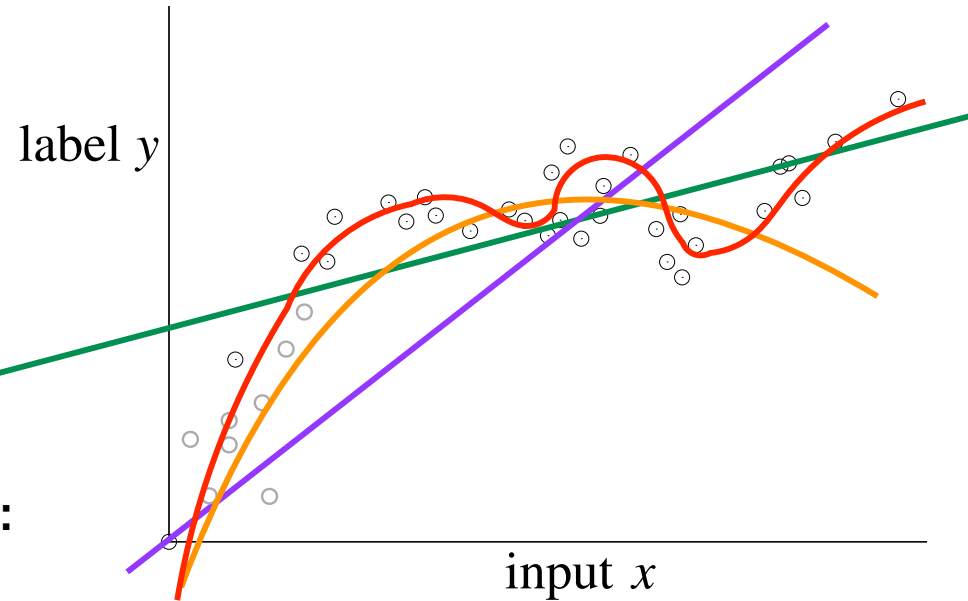
Quadratic regression in 1-dimension

- **Data:** $\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

- **General p-features with parameter** $w = \begin{bmatrix} w_1 \\ \vdots \\ w_p \end{bmatrix}$:
 - $\hat{y}_i = \langle w, h(x_i) \rangle$ where $h : \mathbb{R} \rightarrow \mathbb{R}^p$

Note: h can be arbitrary non-linear functions!

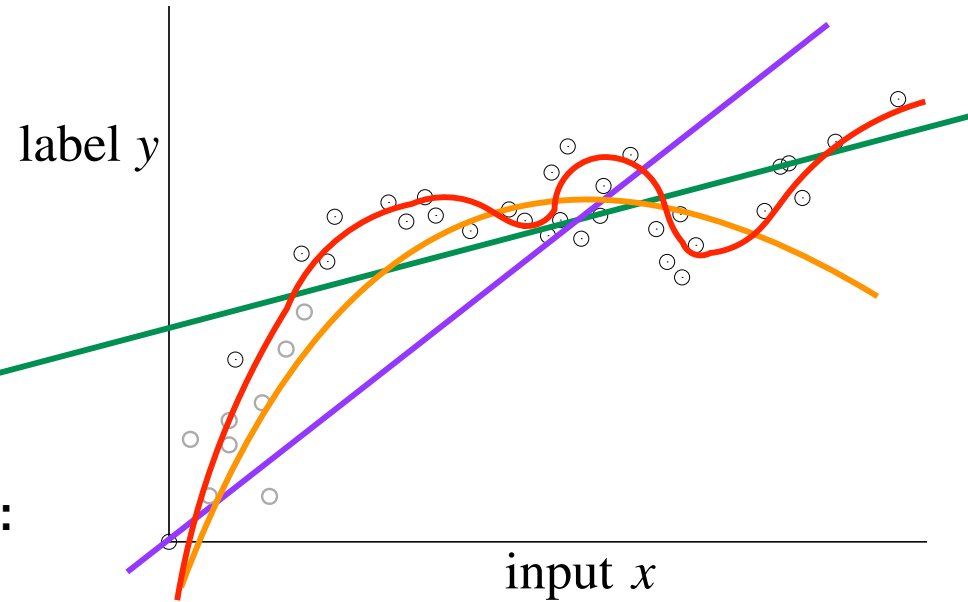
$$h(x) = \left[\log(x), x^2, \sin(x), \sqrt{x} \right]^T$$



Quadratic regression in 1-dimension

- **Data:** $\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

- **General p-features with parameter** $w = \begin{bmatrix} w_1 \\ \vdots \\ w_p \end{bmatrix}$:
 - $\hat{y}_i = \langle w, h(x_i) \rangle$ where $h : \mathbb{R} \rightarrow \mathbb{R}^p$

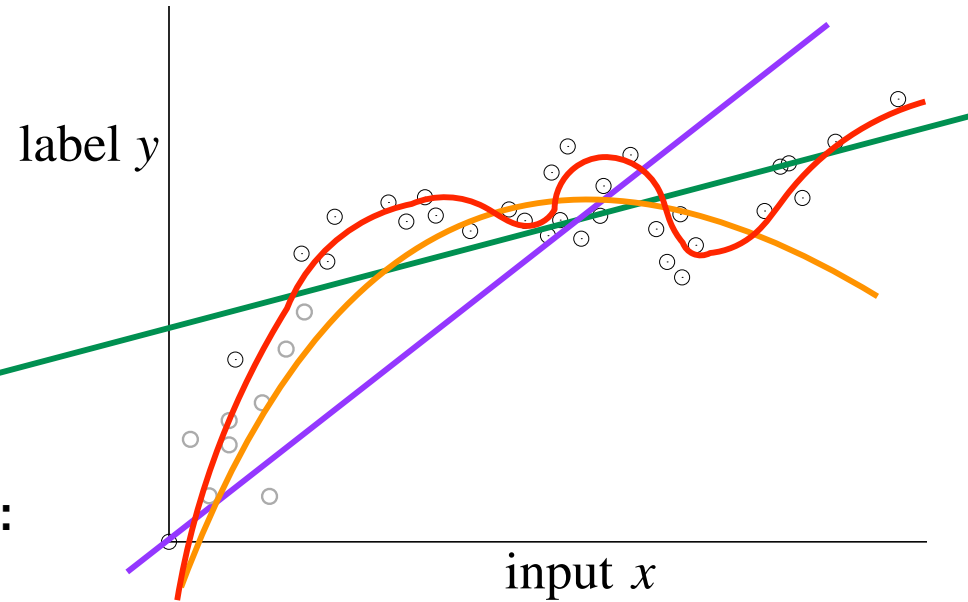


How do we learn w ?

Quadratic regression in 1-dimension

- **Data:** $\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

- **General p-features with parameter** $w = \begin{bmatrix} w_1 \\ \vdots \\ w_p \end{bmatrix}$:
 - $\hat{y}_i = \langle w, h(x_i) \rangle$ where $h : \mathbb{R} \rightarrow \mathbb{R}^p$



How do we learn w ?

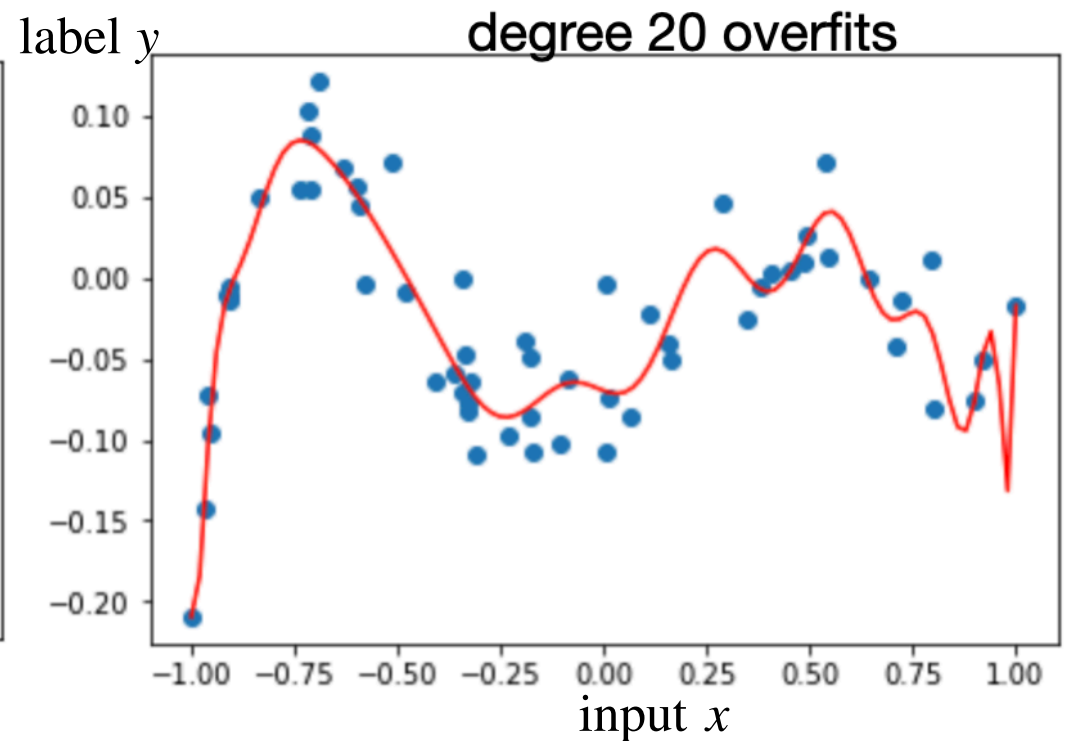
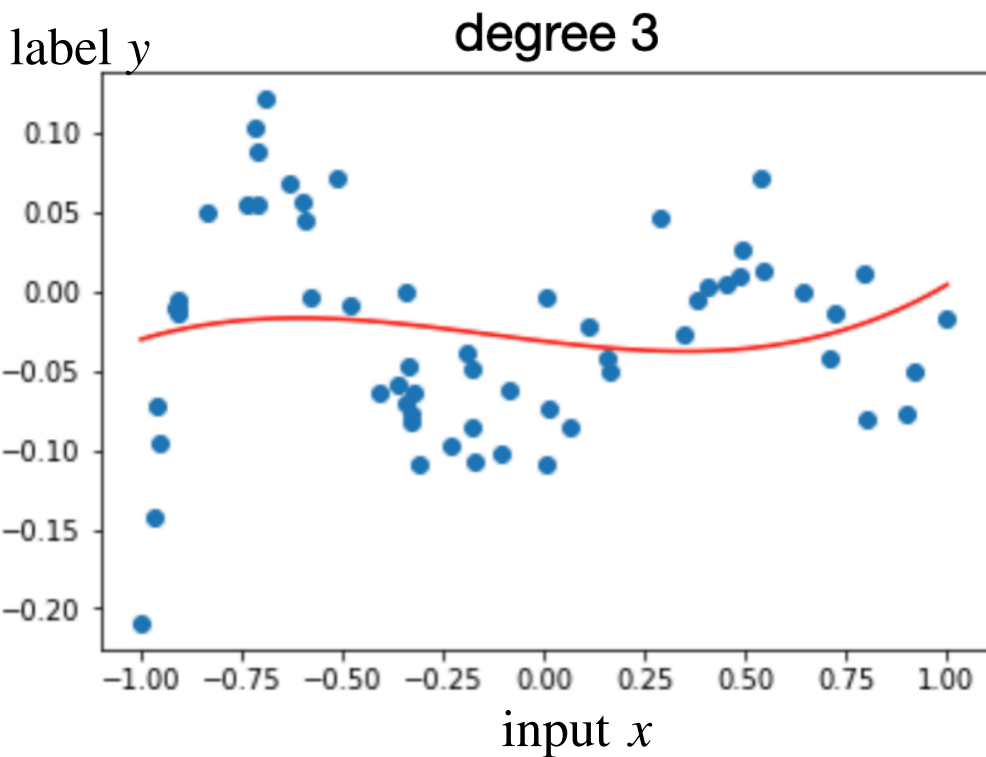
$$\mathbf{H} = \begin{bmatrix} - & - & h(x_1)^\top & - & - \\ & & \vdots & & \\ - & - & h(x_n)^\top & - & - \end{bmatrix} \in \mathbb{R}^{n \times p}$$

$$\hat{w} = \arg \min_w \|\mathbf{H}w - \mathbf{y}\|_2^2$$

For a new test point x , predict
 $\hat{y} = \langle \hat{w}, h(x) \rangle$

Which p should we choose?

- First instance of class of models with different representation power = model complexity



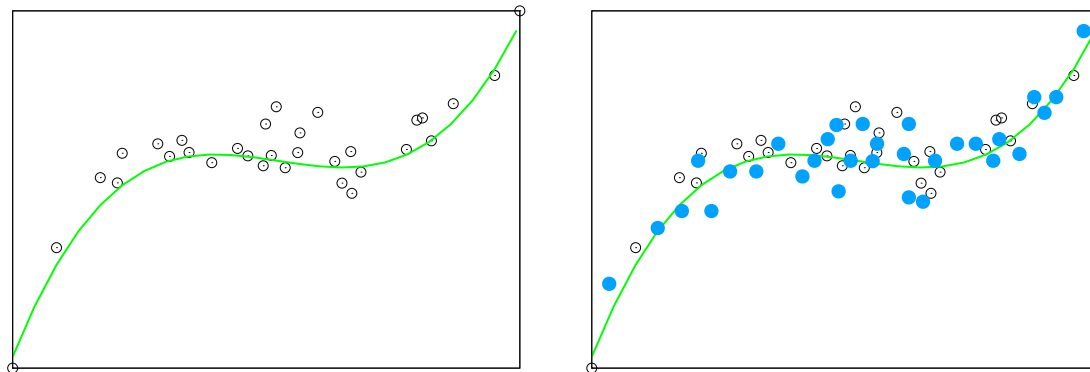
- How do we determine which is better model?

Generalization

- we say a predictor **generalizes** if it performs as well on unseen data as on training data (we will formalize the next lecture)
- the data used to train a predictor is **training data** or **in-sample data**
- we want the predictor to work on **out-of-sample data**
- we say a predictor **fails to generalize** if it performs well on in-sample data but does not perform well on out-of-sample data

Generalization

- we say a predictor **generalizes** if it performs as well on unseen data as on training data (we will formalize the next lecture)
- the data used to train a predictor is **training data** or **in-sample data**
- we want the predictor to work on **out-of-sample data**
- we say a predictor **fails to generalize** if it performs well on in-sample data but does not perform well on out-of-sample data



- **train** a cubic predictor on 32 (**in-sample**) white circles: Mean Squared Error (MSE) 174
- **predict** label y for 30 (**out-of-sample**) blue circles: MSE 192
- conclude this predictor/model generalizes, as in-sample MSE \simeq out-of-sample MSE

Split the data into **training** and **testing**

- a way to mimic how the predictor performs on unseen data
- given a single dataset $S = \{(x_i, y_i)\}_{i=1}^n$
- we split the dataset into two: training set and test set (e.g., 90/10)
- **training set** used to train the model

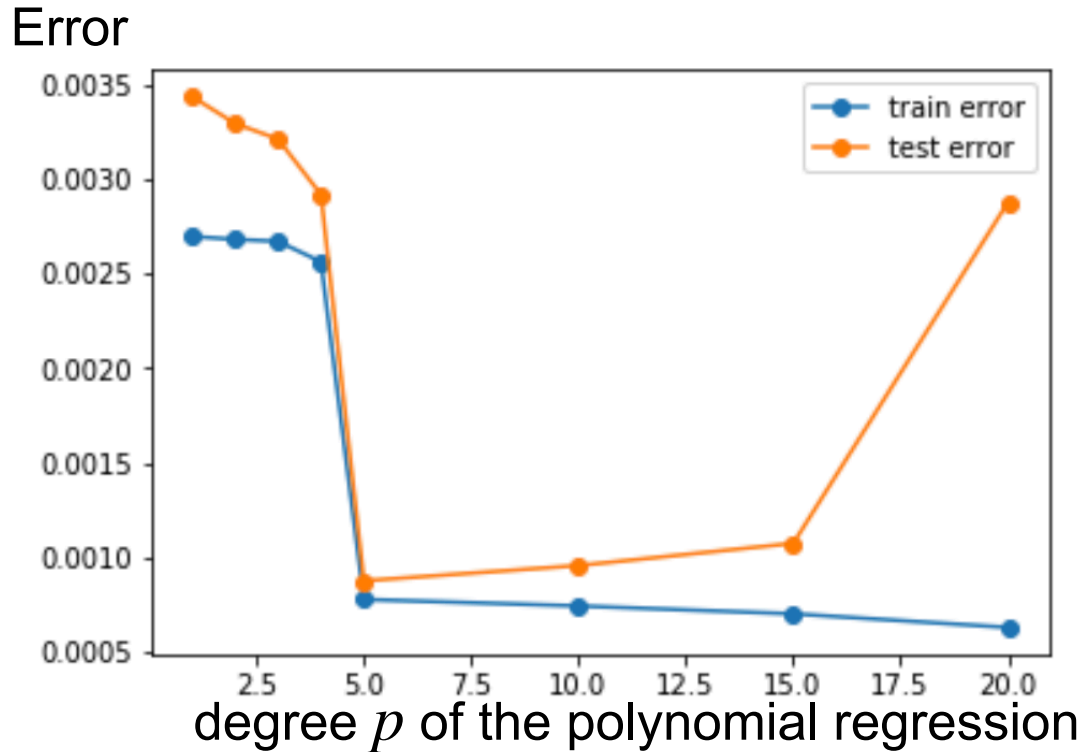
- minimize $\mathcal{L}_{\text{train}}(w) = \frac{1}{|S_{\text{train}}|} \sum_{i \in S_{\text{train}}} (y_i - x_i^T w)^2$

- **test set** used to evaluate the model

- $\mathcal{L}_{\text{test}}(w) = \frac{1}{|S_{\text{test}}|} \sum_{i \in S_{\text{test}}} (y_i - x_i^T w)^2$

- this assumes that test set is similar to unseen data
- **test set should never be used in training or picking unknowns**

Train/test error vs. complexity



- Degree $p = 5$, since it achieves **minimum test error**
- **Train error** monotonically decreases with model complexity
- **Test error** has a U shape

test set should never be used in training or picking degree

