CSE 446
Linear Regression

Natasha Jaques

UNIVERSITY of WASHINGTON

Maximum Likelihood Estimation

Observe $X_1, X_2, \ldots, X_n$ 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) $\widehat{\theta}_{MLE} = \arg\max_{\theta} L_n(\theta)$

Learning Gaussian parameters

- MLE:

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

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

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

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

- Unbiased variance estimator:

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

Maximum Likelihood Estimation

Observe $X_1, X_2, \ldots, X_n$ 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) $\widehat{\theta}_{MLE} = \arg\max_{\theta} L_n(\theta)$

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

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

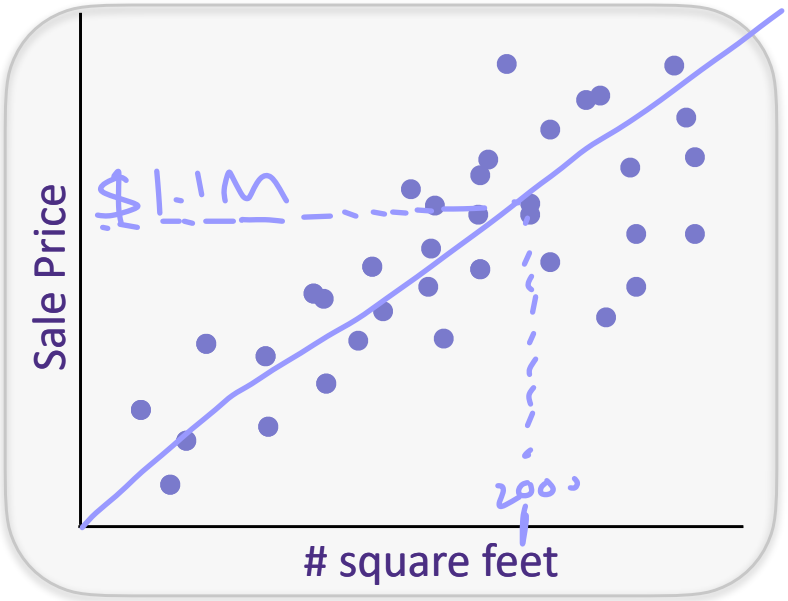
Regression

The regression problem, 1-dimensional

Given past sales data on zillow.com, predict:

y = **House sale price** *from* _

x = **{# sq. ft.}**



Training Data:

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

$x_i \in \mathbb{R}$

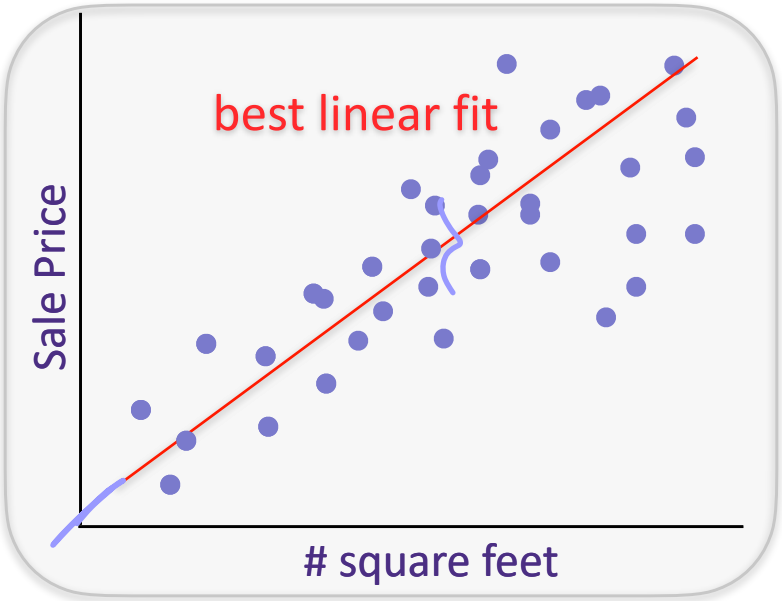
$y_i \in \mathbb{R}$

Fit a function to our data, 1-d

Given past sales data on zillow.com, predict:

y = **House sale price** *from*

x = **{# sq. ft.}**



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

Hypothesis/Model: linear

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

$w \in \mathbb{R}$, slope of line

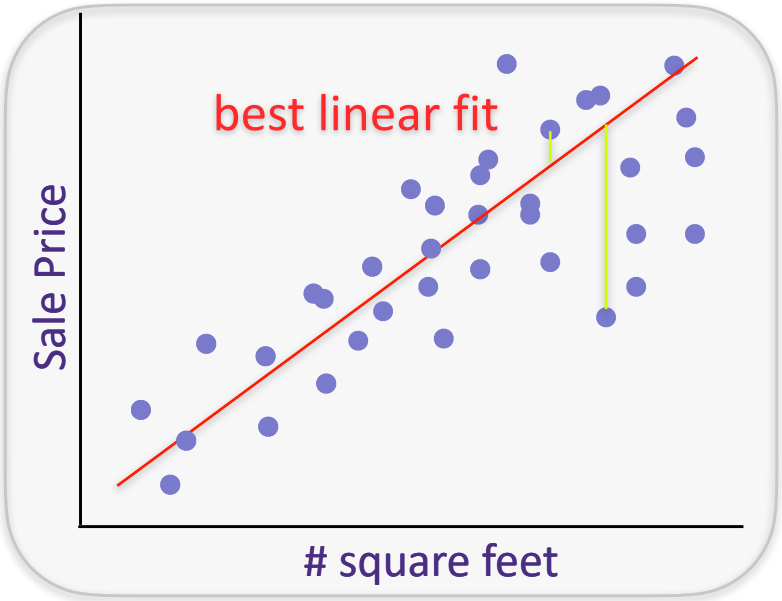
no intercept to make math easy

Fit a function to our data, 1-d

Given past sales data on zillow.com, predict:

y = **House sale price** *from*

x = **{# sq. ft.}**



Training Data:

$$\{(x_i, y_i)\}_{i=1}^{n} \qquad x_i \in \mathbb{R} \quad y_i \in \mathbb{R}$$

Hypothesis/Model: linear

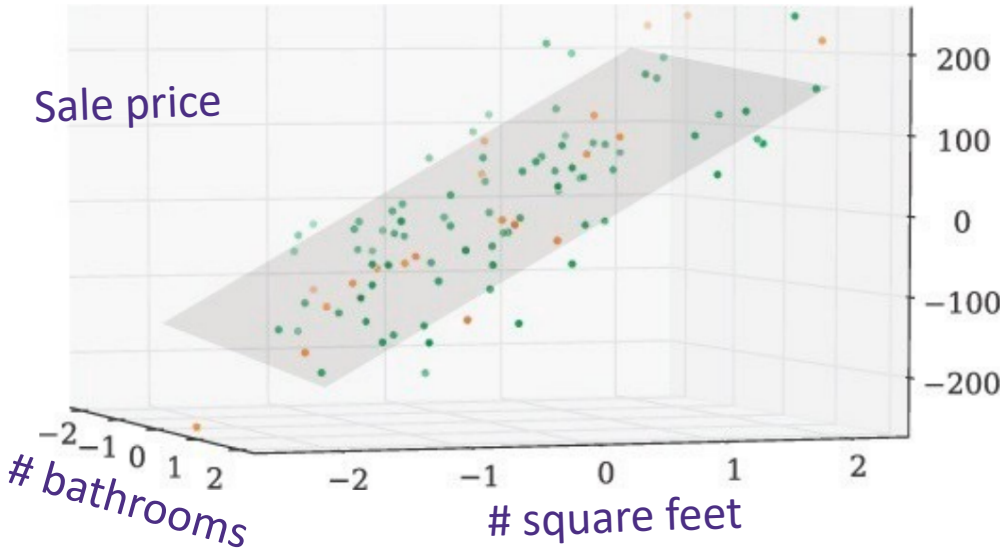
$$y_i = x_i w + \epsilon_i \qquad \epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$$

The regression problem, d-dim

Given past sales data on zillow.com, predict:

y = **House sale price** *from*

x = **{# sq. ft., # baths, date of sale, etc.}**



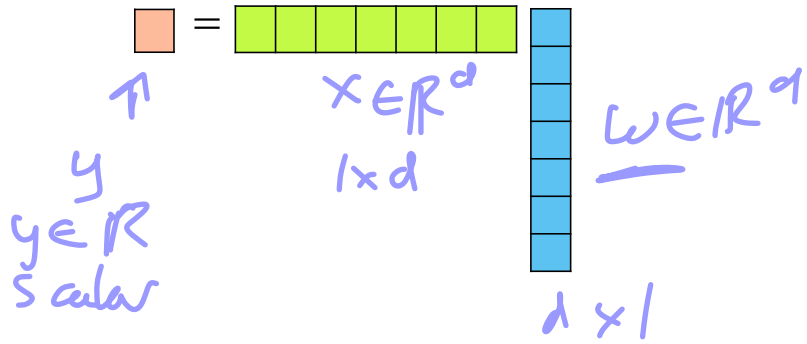
Training Data: $\{(x_i, y_i)\}_{i=1}^n$ $\quad x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$

Hypothesis/Model: linear

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

$$\epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$$

$\hookrightarrow w \in \mathbb{R}^d$



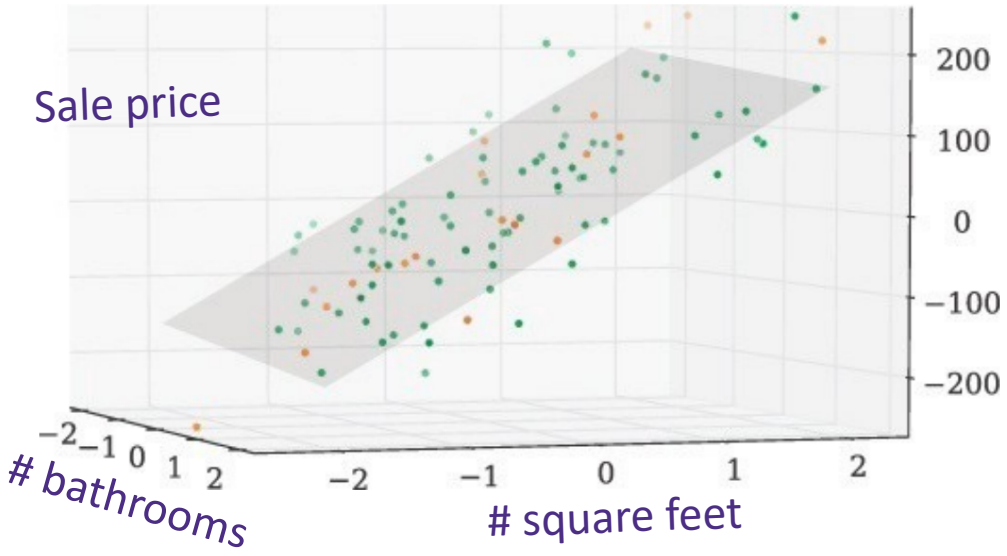

sq ft
bdrms
w_0
w_1
y
house price
x

The regression problem, d-dim

Given past sales data on zillow.com, predict:

y* = House sale price *from

***x* = {# sq. ft., # baths, date of sale, etc.}**



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

Hypothesis/Model: linear

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

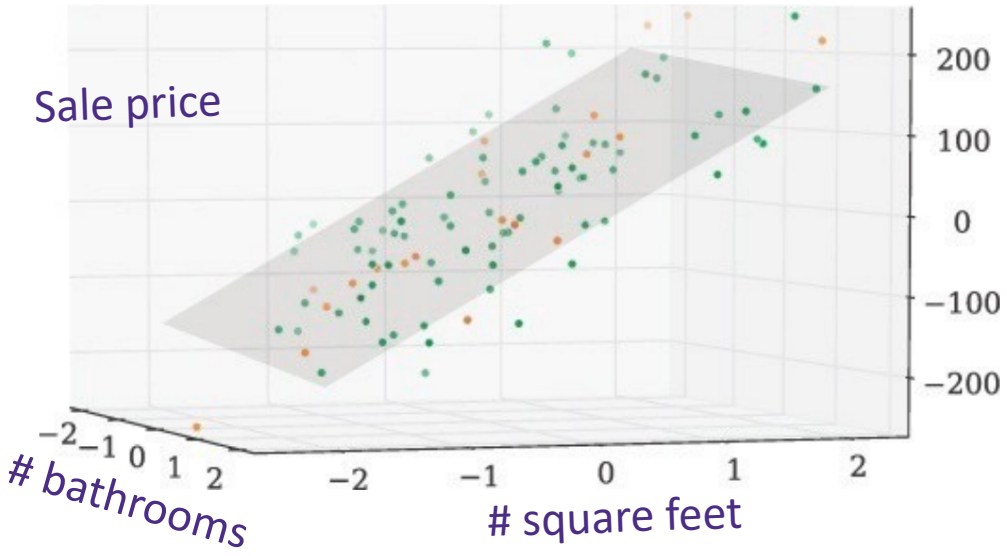
$$p(y|x, w, \sigma) = \mathcal{N}(x^T w, \sigma^2)$$

The regression problem, d-dim

Given past sales data on zillow.com, predict:

y = **House sale price** *from*

x = **{# sq. ft., # baths, date of sale, etc.}**



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

Hypothesis/Model: linear

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

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

$$= \mathcal{N}(x^T w, \sigma^2)$$

Maximizing log-likelihood

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

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

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

Maximum Likelihood Estimation

Observe $X_1, X_2, \ldots, X_n$ 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) $\widehat{\theta}_{MLE} = \arg\max_{\theta} L_n(\theta)$

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

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 $X \sim f(\,\cdot\,;\widehat{\theta}_{MLE})$

Maximizing log-likelihood

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

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

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

LL

$$= \sum_{i=1}^n \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_i - x_i^\top w)^2}{2\sigma^2} \right) \right]$$

$\log AB = \log A + \log B$

$$= n \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \sum_{i=1}^n \frac{(y_i - x_i^\top w)^2}{2\sigma^2}$$

$\leftarrow$ LL

$$\arg\max_w \; - \sum_{i=1}^n \frac{(y_i - x_i^\top w)^2}{2\sigma^2}$$

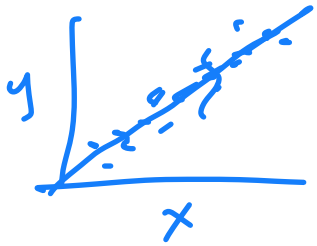
Recap

- ML "algos" we've learned so far...
  - Maximum Likelihood Estimation (MLE)

x $p(x)$ → fit a Bernoulli distribution (coin flips)
→ fit a Gaussian distribution (μ, σ)

x, y $p(y|x)$ → (currently) fit a linear predictor $x \to y$ with Gaussian noise

Maximizing log-likelihood



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

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

$\theta \approx w$

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

y_i

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

minimize squared error

Maximizing log-likelihood

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

vector of partial derivatives

Set gradient=0, solve for w

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

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

$$= \sum_{i=1}^{n} 2(y_i - x_i^T w)(-x_i) = \vec{0}$$

(1×1) (1×d) (d×1) (d×1) → (d×1)

$$= 2 \sum_{i=1}^{n} [y_i x_i + x_i^T w x_i] = 0$$

$$= 2 \sum_{i=1}^{n} [-x_i y_i + x_i x_i^T w] = 0 \Leftarrow$$

$$\sum_{i=1}^{n} x_i y_i = \left[\sum_{i=1}^{n} x_i x_i^T\right] w$$

$$\nabla_w f(x) = \begin{bmatrix} \frac{d}{dw_0} f(x) \\ \frac{d}{dw_1} f(x) \\ \vdots \\ \frac{d}{dw_d} f(x) \end{bmatrix}$$

$$\nabla_w f(x)^2 = 2 f(x) \cdot \nabla_w f(x)$$

// chain rule

$$\nabla_w x^T w = x$$

"Matrix Cookbook"

$x^T w$ = scalar = a

$$a\vec{v} = \vec{v}a \Leftarrow$$

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

$d \times 1$ $\quad$ $d \times d$

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

matrix must be invertible

$n \geq d$

Maximizing log-likelihood

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

Set gradient=0, solve for w

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

The regression problem in matrix notation

we ♡ matrix notation

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

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

d : # of features

n : # of examples/datapoints

x_1^T :- 

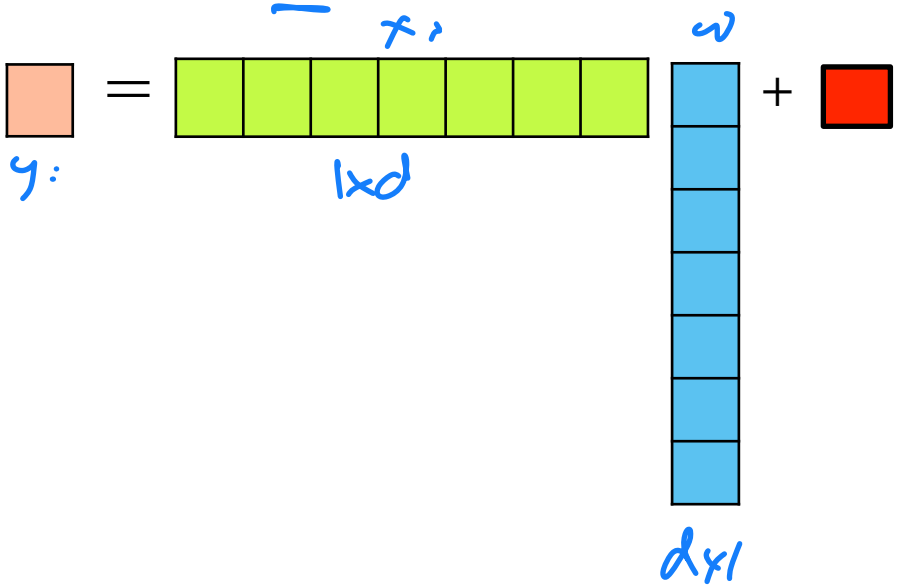
The regression problem in matrix notation

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

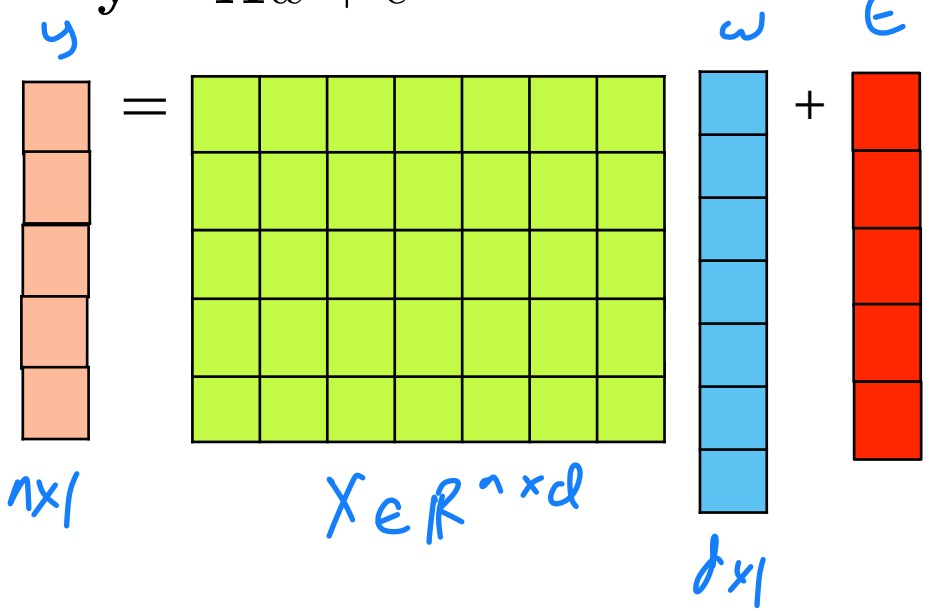
$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$$

d : # of features
n : # of examples/datapoints

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



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



The regression problem in matrix notation

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

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

d : # of features
n : # of examples/datapoints

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

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

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

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

$z_i = y_i - x_i^\top w$

$(ABC)^T = C^T B^T A^T$

$s \in \mathbb{R}$

$s = s^T$

$$= \arg\min_w (y - Xw)^T (y - Xw)$$

$$= \arg\min_w y^T y - y^T Xw - (Xw)^T y + (Xw)^T (Xw)$$

$$= \arg\min_w y^T y - y^T Xw - (w^T X^T y)^T + w^T X^T Xw$$

$1\times n \;\; n\times d \;\; d\times 1$ → 1×1 ; $1\times d \;\; d\times n \;\; n\times 1$ → 1×1

$$= \arg\min_w -y^T Xw - y^T Xw + w^T X^T Xw$$

$$= \arg\min_w -2y^T Xw + w^T X^T Xw$$

$$\nabla_w \left[\underbrace{-2y^T Xw}_{②} + \underbrace{w^T X^T Xw}_{③} \right] = 0$$

$$= -2X^T y + 2X^T Xw = 0$$

Useful gradients

① $\nabla_w x^T w = x$

② $\nabla_w x^T Aw = A^T x$

③ $\nabla_w w^T Aw = (A + A^T)w$

// quadratic form

if $A = A^T$ // symmetric

then $\nabla_w w^T Aw = 2Aw$

$(X^T X)^T = X^T X$

$$= -2X^T y + 2X^T X w = 0$$

$$X^T X w = X^T y$$

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

$d \times n \quad n \times d$ / $d \times d$

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

$d \times 1 \quad 1 \times d$ / $d \times d$

Linear regression $\equiv$ OLS (Ordinary Least Squares)

The regression problem in matrix notation

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

x_i : [features]

w = parameters

d : # of features

n : # of examples/datapoints

labels

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

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

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

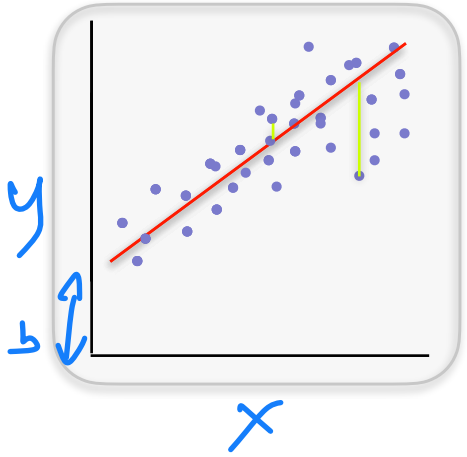
ℓ_2 norm: $\|z\|_2 = \sqrt{\sum_{i=1}^{n} z_i^2} = \sqrt{z^\top z}$

$$\widehat{w}_{LS} = \arg\min_{w} ||\mathbf{y} - \mathbf{X}w||_2^2$$
$$= \arg\min_{w} (\mathbf{y} - \mathbf{X}w)^T (\mathbf{y} - \mathbf{X}w)$$

$$\hat{w}_{LS} = \hat{w}_{MLE} = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

The regression problem in matrix notation

$$\widehat{w}_{LS} = \arg\min_{w} ||\mathbf{y} - \mathbf{X}w||_2^2$$
$$= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$



What about an offset?

$$\widehat{w}_{LS}, \widehat{b}_{LS} = \arg\min_{w,b} \sum_{i=1}^{n} \left(y_i - (x_i^T w + b)\right)^2$$
$$= \arg\min_{w,b} ||\mathbf{y} - (\mathbf{X}w + \mathbf{1}b)||_2^2$$

$\hat{y}_i = x_i^T w + b$

$(y_i - \hat{y}_i)^2$

1×1

$b \in \mathbb{R}$ scalar

Dealing with an offset

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

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

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

$$\begin{bmatrix} X^TX & X^T\underline{1} \\ \underline{1}^TX & \underline{1}^T\underline{1} \end{bmatrix} \begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} X^Ty \\ \underline{1}^Ty \end{bmatrix}$$

$\begin{bmatrix} w \\ b \end{bmatrix}$ = np.linalg.solve(...)

Dealing with an offset

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

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

$\tilde{x}_i = \begin{bmatrix} x_i \\ 1 \end{bmatrix}$ ← scalar $\in \mathbb{R}$

$\tilde{w} = \begin{bmatrix} w \\ b \end{bmatrix}$

$\hat{y}_i = \tilde{x}_i^T \tilde{w}$

$\tilde{X} = \begin{bmatrix} \tilde{x}_1^T \\ \vdots \\ \tilde{x}_n^T \end{bmatrix} = \begin{bmatrix} X & 1 \end{bmatrix}$

$y = \tilde{X}^T \tilde{w}$

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

Dealing with an offset

pre-process features to have mean zero

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

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

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

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

$$\widehat{w}_{LS} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

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

Make Predictions

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

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

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

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

↳ if train features were normalized, have to normalize by subtracting the training mean

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