# Lecture #2: Linear Regression

Read: Murphy – 4.2, 7.1–7.3, 7.8, 11.1÷11.2

Recall the summary of **Maximum Likelihood Estimation (MLE)**

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:

$$\ell_n(\theta) = \log(L_n(\theta)) = \sum_{i=1}^{n} \log(f(x_i;\theta))$$

MLE: $\hat{\theta}_{MLE} = \arg\max_{\theta} L_n(\theta)$

- As the number of observations $n \to \infty$, we have $\hat{\theta}_{MLE} \to \theta^*$
- MLE is a *recipe* that begins with a *model* for data $f(x;\theta)$

when Recipe is applied to continuous case:
Learning Gaussian Parameters (i.e., our model is assumed to be a Gaussian)

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**!

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

(see also definition of biased in Lecture #1)

- Unbiased variance estimator:

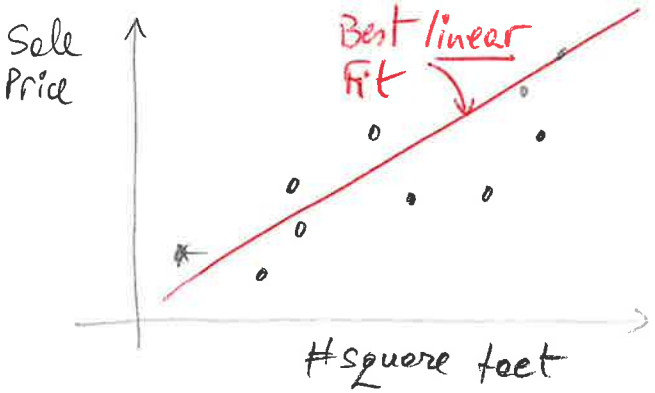
$$\hat{\sigma}^2_{unbiased} = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \hat{\mu}_{MLE})^2$$

# Regression – 1-dimensional (simple linear regression)

Given past sales data on Zillow.com, predict:

$y$ = house sale price from:

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




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

Hypothesis / Model: linear

$$y_i = x_i \cdot w + \varepsilon_i, \quad \varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

$w \in \mathbb{R}$ is the slope of line

weights or regression coefficients.

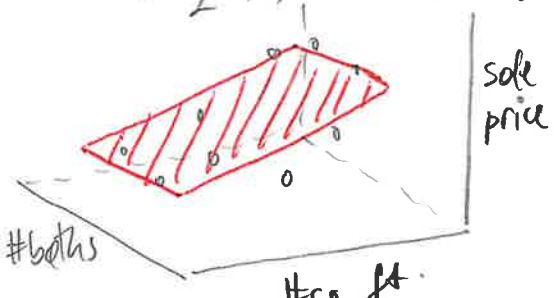
Gaussian Noise

## d-dimensional (multiple linear regression)

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

$y$ = house sale price

$x = \{\#\text{sq. ft}, \#\text{baths}, \text{etc.}\}$




$y \in \mathbb{R}$ scalar $\quad = \quad$ $x_i \in \mathbb{R}^d$ $1 \times d$ vector $\quad$ $w \in \mathbb{R}^d$ $d \times 1$ vector

Hypothesis / Model: Linear

$$y_i = x_i^T \cdot w + \varepsilon_i, \quad \varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

$$p(y \mid x, w, \sigma) = N(x^T w, \sigma^2) \Rightarrow$$

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

(Murphy pp. 110)

Now, maximize Log-Likelihood:

Training Data

$\{(x_i, y_i)\}_{i=1}^n$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$ $\quad p(y|x,w,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(y - x^T w)}{2\sigma^2}}$

Likelihood:

$$P(\mathcal{D}|w,\sigma) = \prod_{i=1}^n P(y_i|x_i,w,\sigma) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(y_i - x_i^T w)}{2\sigma^2}}$$

Maximize – with respect to $\boxed{w}$:

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

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

use: $\log A \cdot B = \log A + \log B$

$$= \underbrace{n \cdot \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)}_{\text{does not depend on } w} - \sum_{i=1}^n \frac{(y_i - x_i^T \cdot w)^2}{2\sigma^2}$$

$$\hat{w}_{MLE} = \arg\max_w \; -\sum_{i=1}^n \frac{(y_i - x_i^T \cdot w)^2}{2\sigma^2}$$

($2\sigma^2$: constant when looking wrt $\boxed{w}$)

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

↑ Minimize squared error!

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

Same idea:
- Take derivative
- set to zero
- Solve for parameter $\omega$.

Set Gradient = 0

$$\nabla_\omega f(x) = \begin{bmatrix} \frac{d}{d\omega_0} f(x) \\ \frac{d}{d\omega_1} f(x) \\ \vdots \\ \frac{d}{d\omega_d} f(x) \end{bmatrix}$$

$$\nabla_\omega f^2(x) = 2 f(x) \nabla_\omega f(x)$$ (chain rule.)

$$\nabla_\omega x^T \omega = x$$

from "matrix cookbook". (See also Murphy 7.8)

in our case:

$$\nabla_\omega \sum_{i=1}^{n} (y_i - x_i^T \omega)^2 = \sum_{i=1}^{n} 2(y_i - x_i^T \omega) \cdot \nabla_\omega (y_i - x_i^T \omega)$$

Set to zero.

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

$(y_i - x_i^T \omega)$: $1 \times 1$; $(-x_i)$: $d \times 1$; product: $1 \times d \cdot d \times 1 = 1 \times 1$; $\vec{0}$: $d \times 1$

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

$x_i^T \omega = \text{scalar} = a$; $a\vec{v} = \vec{v} \cdot a$

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

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

$x_i$: $d \times 1$; $x_i^T$: $1 \times d$; $x_i \cdot x_i^T$: $d \times d$

matrix must be invertible

multiply with inverse of matrix

to arrive at:

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

nice!

# The regression problem in matrix notation:

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

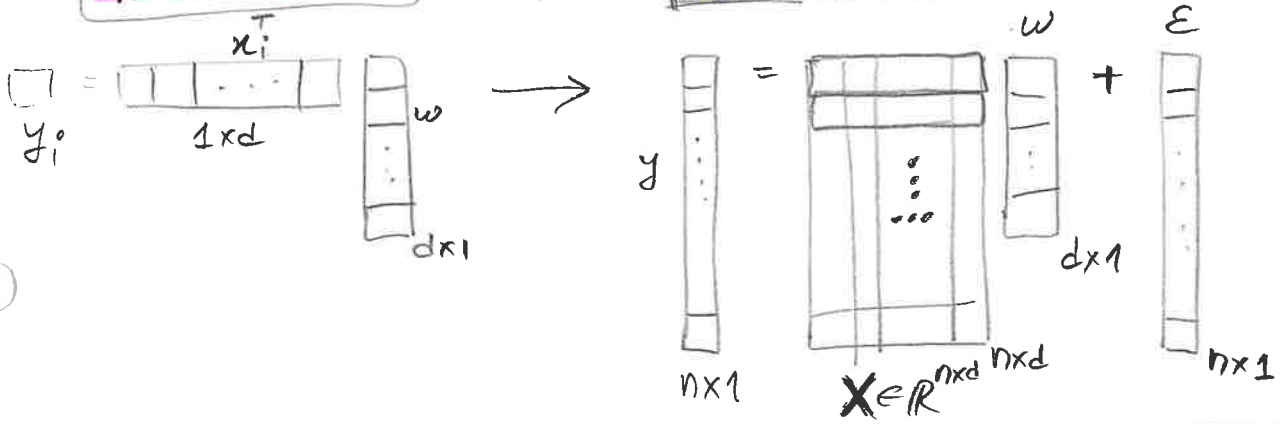
$n$ is # of examples/datapoints

$d$ is # of features.

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \text{Labels} \qquad X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix} \quad \text{Features}$$

$$x_1^T = [x_{11}, x_{12}, \cdots, x_{1d}]_{1\times d}$$

$$y_i = x_i^T \cdot w + \varepsilon_i \xrightarrow{\text{matrix format}} y = X \cdot w + \varepsilon$$




$\ell_2$ norm: $\|z\|_2 = \sqrt{\sum_{i=1}^{n} z_i^2} = \sqrt{z^T \cdot z}$ (Murphy pp. 229)

$z_i = y_i - x_i^T \cdot w$

## Least Squares: (works with Matrices)

$$\hat{w}_{LS} = \arg\min_{w} \left( \|y - Xw\|_2 \right)^2$$

$$\hat{w}_{LS} = \arg\min_{w} (y - Xw)^T (y - Xw)$$

$\triangleq$ Residual Sum of Squares (RSS)

[Murphy 11.2, pp. 367]

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

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

$$= \arg\min_w \; y^T y - y^T X w - \underbrace{\left(w^T X^T y\right)^T}_{1\times 1} + w^T X^T X w$$

($y^T y$: $1\times n \cdot n\times d \cdot d\times 1 = 1\times 1$; $w^T X^T y$: $1\times d \cdot d\times n \cdot n\times 1 = 1\times 1$)

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

$S \in \mathbb{R}, \; S = S^T$

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

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

$$\nabla_w \left[ -2 y^T X w + w^T X^T X w \right] = 0$$

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

Useful gradients:

① $\nabla_w (x^T w) = x$

② $\nabla_w (x^T A w) = A^T x$

③ $\nabla_w (w^T A w) = (A + A^T) w$

if $A = A^T$ symmetric then:

$\nabla_w (w^T A w) = 2Aw$

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

multiply with $(X^T X)^{-1}$ | $X^T X w = X^T y$

NOTE: ① this assumes $X^T X$ is invertible!

② if it is singular (determinant = 0) $\Rightarrow$ many optimal solutions exist!

$$\hat{w}_{LS} = (X^T X)^{-1} X^T y$$ ← unique sol.

($(X^T X)^{-1}$: $d\times n \cdot n\times d = d\times d$)

③ $X^{\dagger} \triangleq (X^T X)^{-1} X^T$

called the pseudo-inverse in Abu-Mustafa ... and would define $X^{\dagger}$ in other ways if $X^T X$ is singular!

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

($x_i x_i^T$: $d\times 1 \cdot 1\times d = d\times d$)

Hence, we note that:

Linear Regression (see page 4) ≡ OLS (Ordinary) Least Squares

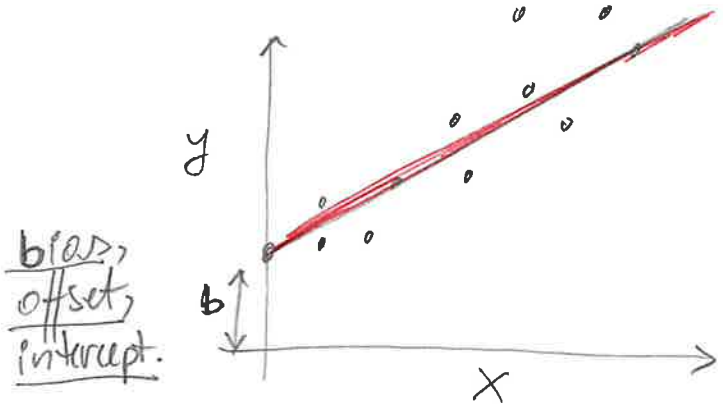
$$\hat{w}_{LS} = \hat{w}_{MLE} = (X^T X)^{-1} X^T y$$

Called "Normal Equation" in Geron's Book (pp. 131)

(Murphy pp 111, 4.2)

# What about an offset?

$$\hat{\omega}_{LS} = \arg\min_{\omega} \|y - X\omega\|_2^2$$
$$= (X^T X)^{-1} \cdot X^T y$$




bias, offset, intercept.

$$\hat{\omega}_{LS}, \hat{b}_{LS} = \arg\min_{\omega, b} \sum_{i=1}^{n} (y_i - (x_i^T \omega + b))^2$$

$$\hat{\omega}_{LS}, \hat{b}_{LS} = \arg\min_{\omega, b} \left( \|y - (X\omega + \mathbf{1}b)\|_2 \right)^2 \leftarrow \text{in matrix format.}$$

$\mathbf{1}$: $n \times 1$, $b \in \mathbb{R}$ scalar

$$\begin{cases} X^T X \hat{\omega}_{LS} + \hat{b}_{LS} X^T \cdot 1 = X^T y \\ 1^T X \hat{\omega}_{LS} + \hat{b}_{LS} 1^T \cdot 1 = 1^T y \end{cases}$$

$$\begin{bmatrix} X^T X & X^T \cdot 1 \\ 1^T \cdot X & 1^T \cdot 1 \end{bmatrix} \cdot \begin{bmatrix} \omega \\ b \end{bmatrix} = \begin{bmatrix} X^T \cdot y \\ 1^T \cdot y \end{bmatrix}$$

$$\tilde{x}_i = \begin{bmatrix} x_i \\ 1 \end{bmatrix} \quad \tilde{\omega} = \begin{bmatrix} \omega \\ b \end{bmatrix} \quad y_i = \tilde{x}_i^T \cdot \tilde{\omega}$$

scalar $\in \mathbb{R}$

Murphy introduces notation $b \triangleq \omega_0$, which is essentially absorbed in $\omega$.

$$\tilde{X} = \begin{bmatrix} \tilde{x}_1^T \\ \tilde{x}_2^T \\ \vdots \\ \tilde{x}_n^T \end{bmatrix} = [X \; 1]$$

$$y = \tilde{X}^T \cdot \tilde{\omega}$$

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

# Appendix A

Derivation of the two-equations expressed as a compact block-matrix system ≡ is mathematically equivalent to augmenting $X$ with a column of 1's and solving the usual normal equations.

- Note: augmenting $X$ can be done with ones at the end or at the beginning

- Start with: $\hat{w}_{LS}, \hat{b}_{LS} = \underset{w,b}{\operatorname{argmin}} \underbrace{\left( \| y - (Xw + 1b) \|_2 \right)^2}_{\text{"}\mathcal{L}(\theta)\text{"} - \text{"loss function"},\ \theta = (w,b)}$ : metrix formod.

- Derivative/gradient w.r.t. $w$:

$$\nabla_w \mathcal{L}(w,b) = -2X^T(y - Xw - \underbrace{1}_{n\times 1} b) \quad \text{scaler}$$

Set to zero:

$$\boxed{X^T y = X^T X w + b X^T 1} \quad (1) \qquad 1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1}$$

- Gradient w.r.t. $b$:

$$\nabla_b \mathcal{L}(w,b) = -2 \cdot 1^T (y - Xw - 1b)$$

Set to zero:

$$\boxed{1^T y = 1^T X + b\, 1^T 1} \quad (2)$$

(1), (2) $\Rightarrow$

$$\begin{bmatrix} X^T y \\ 1^T y \end{bmatrix} = \underbrace{\begin{bmatrix} X^T X & X^T 1 \\ 1^T X & 1^T 1 \end{bmatrix}}_{\text{if this is invertible, then, we can drive:}} \cdot \begin{bmatrix} w \\ b \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} \hat{w} \\ \hat{b} \end{bmatrix} = \begin{bmatrix} X^T X & X^T 1 \\ 1^T X & 1^T 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} X^T y \\ 1^T y \end{bmatrix}$$

which is equivalent to what we would get if we appended $x_i$ with ones, append $w$ with $b$, write metrix formod and solve usual normal regression as done before:

$$\tilde{x}_i = \begin{bmatrix} x_i \\ 1 \end{bmatrix}, \quad \tilde{w} = \begin{bmatrix} w \\ b \end{bmatrix} \qquad \tilde{y}_i = \tilde{x}_i^T \cdot \tilde{w} \xrightarrow{\text{metrix formod}} \boxed{y = \tilde{X}^T \cdot \tilde{w}}, \text{ with } \tilde{X} = \begin{bmatrix} \tilde{x}_1^T \\ \tilde{x}_2^T \\ \vdots \\ \tilde{x}_n^T \end{bmatrix} = [X\,1]$$

Linear Regression $\Rightarrow$

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

So, idea is to preprocess features to have mean zero.

if $X^T \cdot 1 = 0$ (i.e., if each feature is mean-zero) then:

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

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

Now, **Making Predictions:**

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

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

↳ if train features were normalized have to normalize by substracting the training mean.

**Process:**

- Decide on a **model** for the likelihood function $f(x; \theta)$
- Find the function that fits best the **data**
  - > Choose a **loss** function - least squares
  - > Pick the function that **minimizes** loss on data.
- Use function to make prediction on new examples.

# Recap So Far

ML algorithms we looked at so far:

— Maximum Likelihood Estimation (MLE)

$x$, $p(x)$
- → Fit a Bernoulli distribution (coin flips)
- → Fit a Gaussian distribution $(\mu, \sigma)$

$x, y$, $p(y|x)$ → Fit a Linear Predictor $x \to y$ with Gaussian Noise