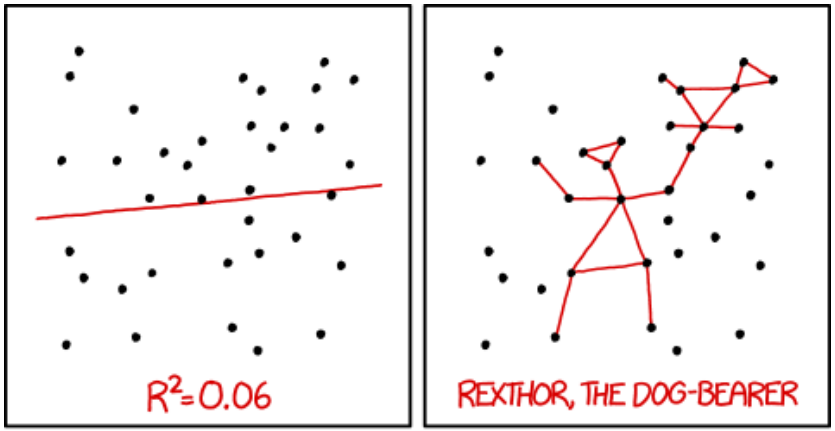
Data Science (COSE471) Spring 2021

Linear Regression



Dept. of Computer Science and Engineering

Korea University

Outline

- Linear Regression
  - Introduction
  - Linear Regression Algorithm
  - Interpretation via Hat Matrix
- Summary

Readings

- Learning from Data by Abu-Mostafa, Magdon-Ismail, and Lin
  - Chapter 3: The Linear Model (Sections 3.1 & 3.2)
- References on linear algebra (optional)
  - Linear Algebra (MIT 18.06)
    - A well-known OCW
  - Introduction to Linear Dynamical Systems (Stanford EE263)
    - Lecture 3: Linear Algebra Review
    - Lecture 5: Least-Squares

The linear model

- Set of lines
- Often a good first choice
  - Small VC (Vapnik-Chervonenkis) dimension
  - Generalize well

Small dimension make more generalization

Three important problems

- Classification
- Regression
- Probability Estimation (aka logistic regression)

Artificial Neural Net
Deep Neural Net

➔ Come with different but related algorithms

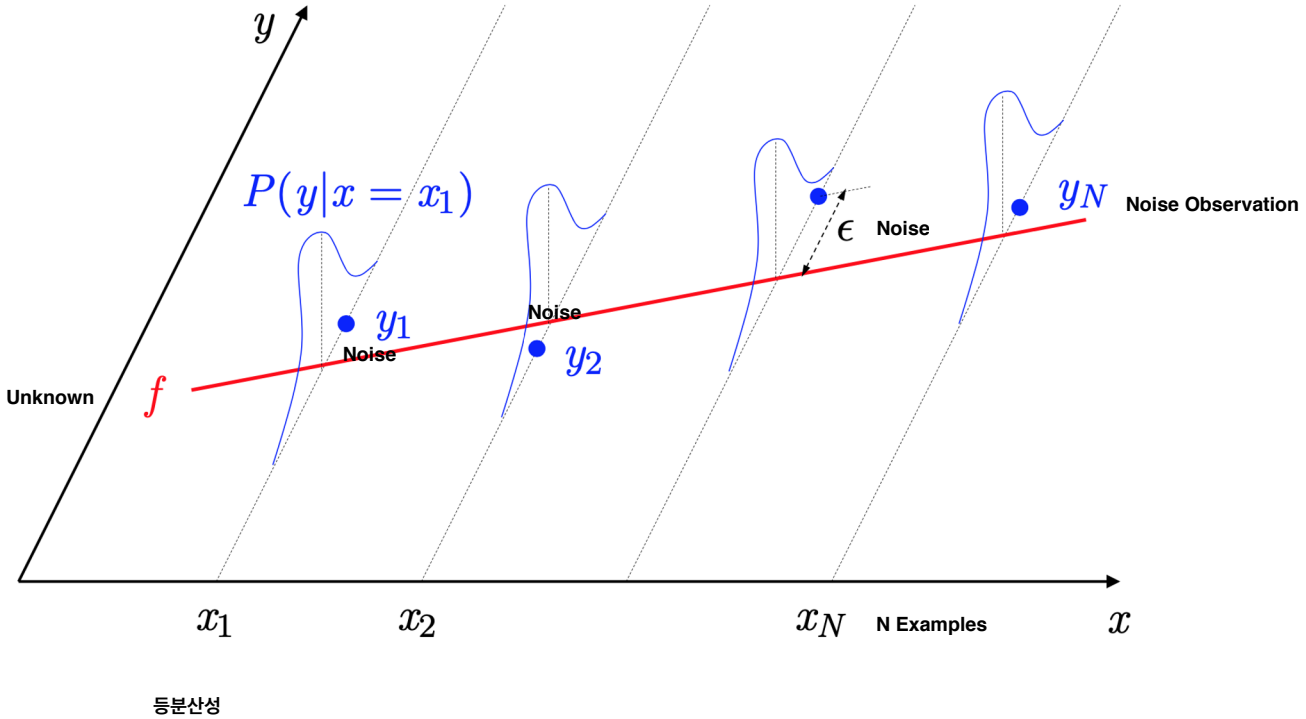
Outline

- Linear Regression
  - Introduction
  - Linear Regression Algorithm
  - Interpretation via Hat Matrix
- Summary

Regression

- A statistical method to study relationship between $\mathbf{x}$ and $\mathbf{y}$
  - input
    $\mathbf{x}$: covariate / predictor variable / independent variable / feature
  - output
    $\mathbf{y}$: response / dependent variable
- Training data $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_N, \mathbf{y}_N)$
  - Noise ϵ is added to target: $y_n = f(\mathbf{x}_n) + \epsilon$ (noise; unknown)
  - $y \sim P(y|\mathbf{x})$ instead of $y = f(\mathbf{x})$
- Our goal
  - find a model $g(\mathbf{x})$ that approximates $\mathbf{y}_n$

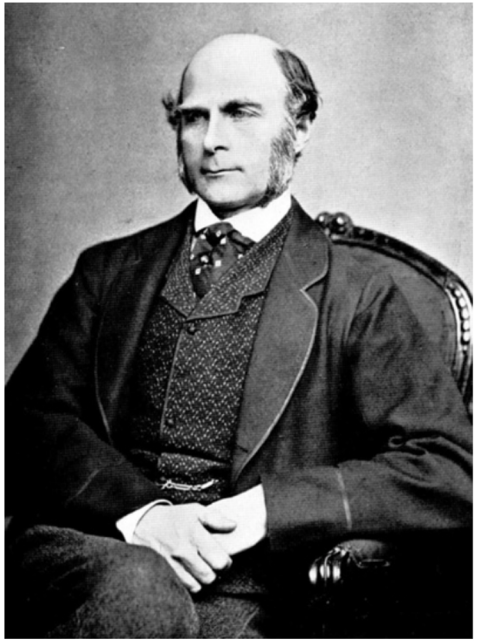
- Illustration (simple linear regression)

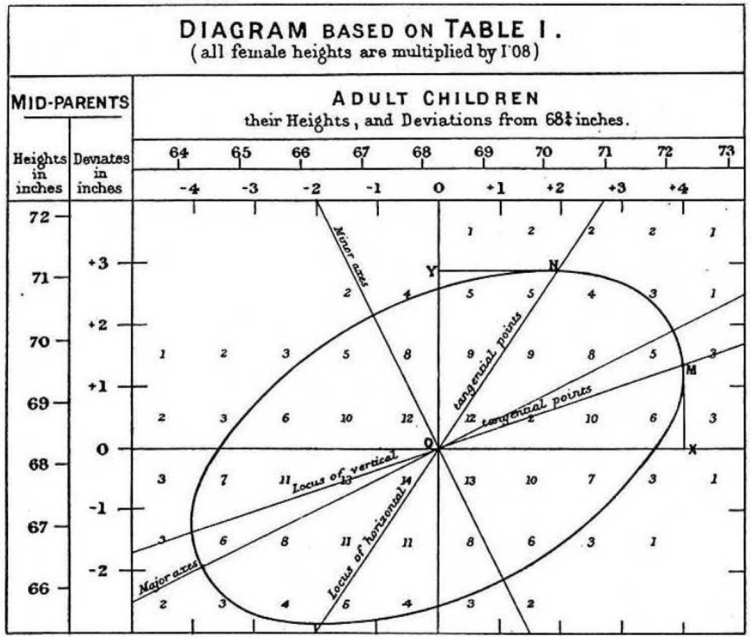


등분산성

Assumption: homoscedasticity (homogeneity of variance; the same finite variance for each value of $\mathbf{x}$)

- Etymology: Sir Francis Galton (1822-1911)
  - re (back)+gression (going): going back from data to formula
  - regression towards the mean: “*tall and short men tend to have sons with heights closer to the mean*”





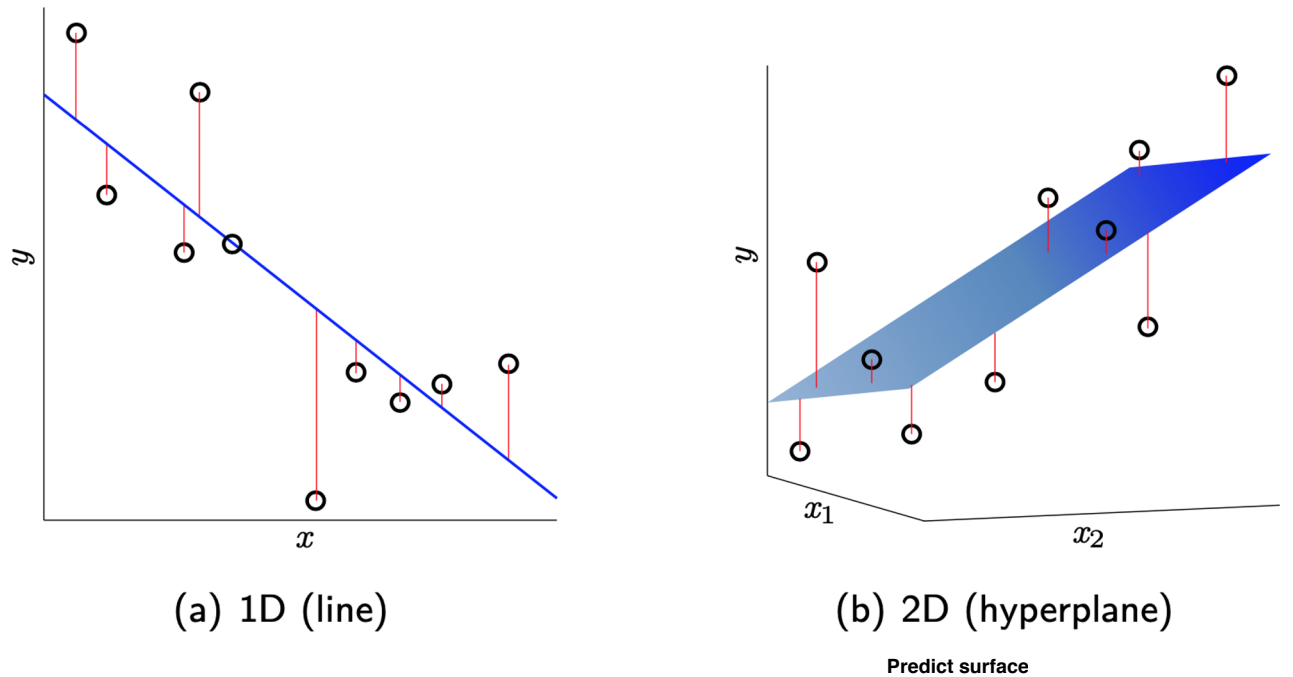
Linear regression

- Popular linear model for predicting a quantitative response
  - Applies to real-valued target functions
  - Long history in statistics, social, and behavioral sciences
  - “Regression” means $\mathbf{y}$ is real-valued (inherited from statistics)
- Simple vs. multiple
  - $d = 1$: simple linear regression (one predictor)
  - $d \geq 2$: multiple linear regression (multiple predictors)
- We discuss simple linear regression from a learning perspective

Parameter estimation for linear regression

- Least squares
  - Ordinary least squares (OLS)
    - Minimizes the sum of squared residuals
    - Leads to a closed-form expression
  - Generalized least squares, iteratively reweighted least squares
- Maximum likelihood
  - Ridge / Lasso regression: regularized
  - Least absolute deviation regression
- Other techniques
  - Bayesian linear regression, principal component regression

Linear regression in 1D and 2D



(a) 1D (line)

(b) 2D (hyperplane)

Figure 4: solution hypothesis (in blue) of linear regression algorithm in 1D and 2D (in OLS, sum of squared error is minimized)

Example: credit approval revisited

- Consider a regression problem (rather than classification)
  - e.g. set a credit limit for each customer
- The bank uses historical records to build a data set $\mathbb{D}$
  - $\mathbb{D}$: $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots, (\mathbf{x}_N, \mathbf{y}_N)$
  - $\mathbf{x}_n$: customer information
  - $\mathbf{y}_n$: credit limit (set by human experts)
- $\mathbf{y}_n$ is now a real number
  - e.g. $123,000 for Alice and $57,000 for Bob

- The bank wants to use learning to find a hypothesis $\boldsymbol{g}$
  - $\boldsymbol{g}$ replicates how human experts determine credit limits
- Our target:
  - not a deterministic function $y = f(\mathbf{x})$
    - there is more than one human expert
  - Instead: noisy target
    - Formalized as a distribution of random variable
- Regression label $\mathbf{y}_n$ comes from
  - Some distribution $P(y \mid \mathbf{x})$
  - Instead of a deterministic function $f(\mathbf{x})$

Outline

- Linear Regression
  - Introduction
  - Linear Regression Algorithm
  - Interpretation via Hat Matrix
- Summary

Linear regression algorithm

- Based on minimizing squared error between $\boldsymbol{h}(\mathbf{x})$ and y
  - Expected value: taken wrt joint pdf $P(\mathbf{x}, y)$

$$\mathrm{E}_{\mathrm{out}}(h) = \mathbb{E}\left[(h(\mathbf{x}) - y)^2\right]$$

- Goal
  - find a hypothesis $\boldsymbol{h}$ that achieves a small $\boldsymbol{E}_{\mathrm{out}}(\boldsymbol{h})$
- Issue: $\boldsymbol{E}_{\mathrm{out}}$ cannot be computed
  - $P(\mathbf{x}, y)$ is unknown

- Thus, we resort to in-sample error instead:

$$\mathrm{E}_{\text{in}}(h) = \frac{1}{N}\sum_{n=1}^{N}(h(\mathbf{x}_n) - y_n)^2$$

- In linear regression, $\boldsymbol{h}$ takes the form of
  - a linear combination of the components of $\mathbf{x}$:

$$h(\mathbf{x}) = \sum_{i=0}^{d} w_i x_i = \mathbf{w}^\top \mathbf{x}$$

  - $\mathbf{w}^\top \mathbf{x}$: also called signal

Matrix representation: data

- data matrix $X \in \mathbb{R}^{N \times (d+1)}$
  - rows: inputs $\mathbf{x}_n$ as row vectors
- Target vector $\mathbf{y} \in \mathbb{R}^N$
  - components: target values y_n
- example:

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Matrix representation: in-sample error

- in-sample error is a function of $\mathbf{w}$ and data X, $\mathbf{y}$:

$$\begin{aligned}
\mathrm{E}_{\text{in}}(\mathbf{w}) &= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}^\top \mathbf{x}_n - y_n)^2 \\
&= \frac{1}{N}\left|\left|\begin{bmatrix} \mathbf{x}_1^\top \mathbf{w} - y_1 \\ \mathbf{x}_2^\top \mathbf{w} - y_2 \\ \vdots \\ \mathbf{x}_n^\top \mathbf{w} - y_n \end{bmatrix}\right|\right|^2 && (2) \\
&= \frac{1}{N}|| \qquad\quad ||^2 && (3) \\
&= \frac{1}{N}(\mathbf{w}^\top X^\top X\mathbf{w} - 2\mathbf{w}^\top X^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y}) && (4)
\end{aligned}$$

- $||\cdot||$: Euclidean norm of a vector
- Scalar $\mathbf{y}^\top X\mathbf{w} = \left(\mathbf{w}^\top X^\top \mathbf{y}\right)^\top = \mathbf{w}^\top X^\top \mathbf{y}$

- Example

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

$$\mathrm{E}_{\mathrm{in}}(\mathbf{w}) = \frac{1}{4}\sum_{i=1}^{4}(\mathbf{w}^\top \mathbf{x}_n - y_n)^2$$

$$= \frac{1}{4}\left\| \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \right\|^2$$

Getting the solution

- $\mathbf{w}_{\text{lin}}$: the solution to linear regression
  - derived by minimizing $\boldsymbol{E}_{\text{in}}(\mathbf{w})$ over all possible $\mathbf{w} \in \mathbb{R}^{d+1}$

$$\mathbf{w}_{\text{lin}} = \underset{\mathbf{w}\in\mathbb{R}^{d+1}}{\operatorname{argmin}} \mathrm{E}_{\text{in}}(\mathbf{w}) \tag{5}$$

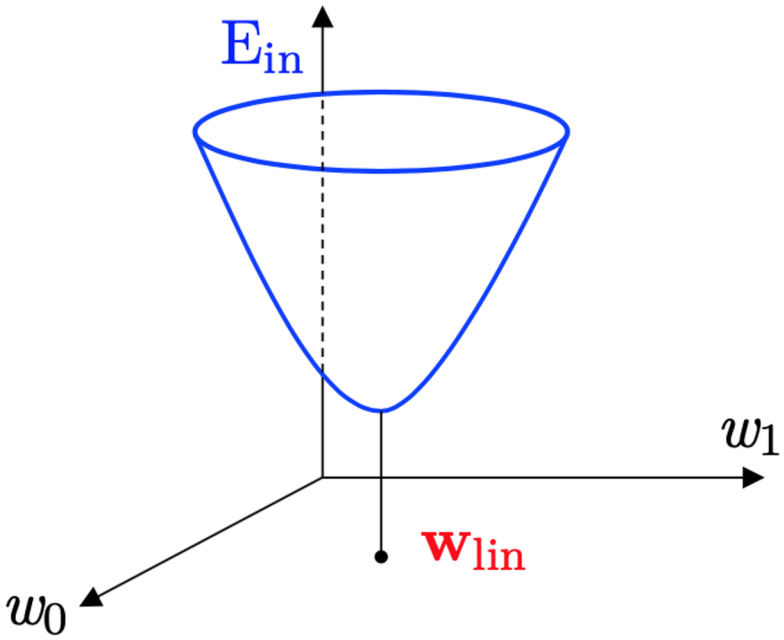
$$= \underset{\mathbf{w}\in\mathbb{R}^{d+1}}{\operatorname{argmin}} \frac{1}{N}||X\mathbf{w} - \mathbf{y}||^2 \tag{6}$$

Minimization

- Eq. (4) implies
  - $\boldsymbol{E}_{\text{in}}(\mathbf{w})$ is continuous, differentiable, and convex
  - We can use standard matrix calculus to find $\mathbf{w}$ that minimizes $\boldsymbol{E}_{\text{in}}(\mathbf{w})$ by requiring:

$$\nabla \boldsymbol{E}_{\text{in}}(\mathbf{w}) = 0$$

- General optimization techniques
  - e.g. gradient descent



- Recall: gradient identifies

$$\nabla_{\mathbf{w}}(\mathbf{w}^\top A\mathbf{w}) = (A + A^\top)\mathbf{w}$$
$$\nabla_{\mathbf{w}}(\mathbf{w}^\top \mathbf{b}) = \mathbf{b}$$

- scalar w

$$\mathrm{E}_{\text{in}}(w) = aw^2 - 2bw + c$$
$$\frac{\partial}{\partial w}\mathrm{E}_{\text{in}}(w) = 2aw - 2b$$

- vector $\mathbf{w}$

$$\mathrm{E}_{\text{in}}(\mathbf{w}) = \mathbf{w}^\top A\mathbf{w} - 2\mathbf{w}^\top \mathbf{b} + c$$
$$\nabla \mathrm{E}_{\text{in}}(\mathbf{w}) = (A + A^\top)\mathbf{w} - 2\mathbf{b}$$

The solution

- From eq. (4) $\frac{1}{N}(\mathbf{w}^\top X^\top X\mathbf{w} - 2\mathbf{w}^\top X^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y})$ (4)

$$\nabla \mathrm{E}_{\mathrm{in}}(\mathbf{w}) = \frac{2}{N}(X^\top X\mathbf{w} - X^\top \mathbf{y})$$

  - both $\mathbf{w}$ and $\boldsymbol{\nabla E}_{\mathrm{in}}(\mathbf{w})$ are column vectors

- Finally, one should solve for $\mathbf{w}$ that satisfies the _____ equations:

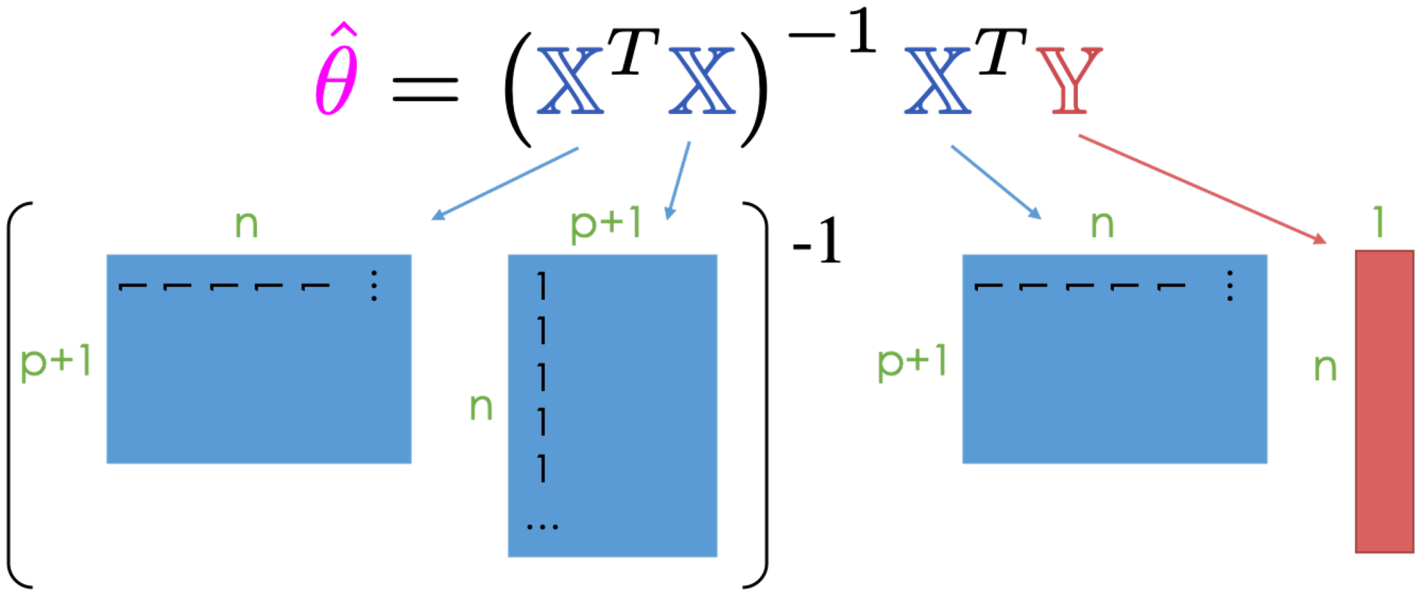
$$X^\top X\mathbf{w} = X^\top \mathbf{y}$$

Two scenarios

- If $\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}$ is invertible, $\mathbf{w} = \boldsymbol{X}^{\dagger}\boldsymbol{y}$
  - $\boldsymbol{X}^{\dagger} = (\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathsf{T}}$ is pseudo-inverse of $\boldsymbol{X}$
  - resulting $\mathbf{w}$ is the unique optimal solution to (5)
- if $\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}$ is not invertible
  - a pseudo-inverse can still be defined, but no unique solution
  - there will be _____ solutions for $\mathbf{w}$ that minimizes $\boldsymbol{E}_{\text{in}}$

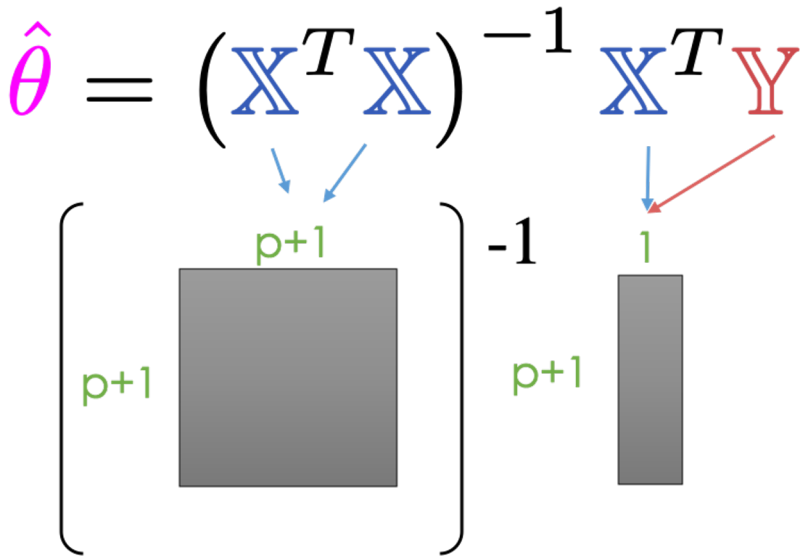
Understanding the solution matrices

Typically, **n is much larger than p**.



Understanding the solution matrices

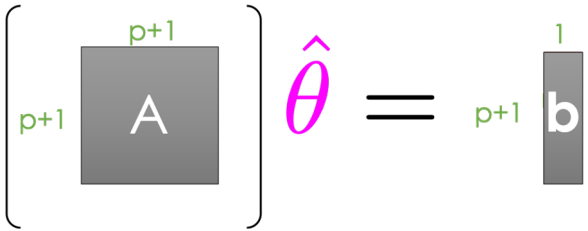
Typically, **n is much larger than p**.



Understanding the solution matrices

The **Normal Equation**:

$$\mathbb{X}^T\mathbb{X}\hat{\theta} = \mathbb{X}^T\mathbb{Y}$$



Our optimal parameter vector can be thought of as the solution to a set of p + 1 equations, with p + 1 unknowns.

In practice

- $\boldsymbol{X}^\mathsf{T}\boldsymbol{X}$ is invertible in most cases
  - since N is often much bigger than d+1
  - there will likely be d+1 linearly independent vectors $\mathbf{x}_n$
- when $\boldsymbol{X}^\mathsf{T}\boldsymbol{X}$ is (almost) singular
  - use a well-implemented $\boldsymbol{X}^\dagger$ routine instead of $(\boldsymbol{X}^\mathsf{T}\boldsymbol{X})^{-1}\boldsymbol{X}^\mathsf{T}$
  - this is for numerical stability
- we have derived the following linear regression algorithm

The linear regression algorithm

- Construct matrix $\boldsymbol{X}$ and vector $\boldsymbol{y}$ as follows:
  - use data set $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_N, \mathbf{y}_N)$
  - each $\mathbf{x}$ includes $x_0 = 1$ bias coordinate

$$\underbrace{X = \begin{bmatrix} — \mathbf{x}_1^\top — \\ — \mathbf{x}_2^\top — \\ \vdots \\ — \mathbf{x}_N^\top — \end{bmatrix}}_{\text{input data matrix}} \qquad \underbrace{\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\text{target vector}}$$

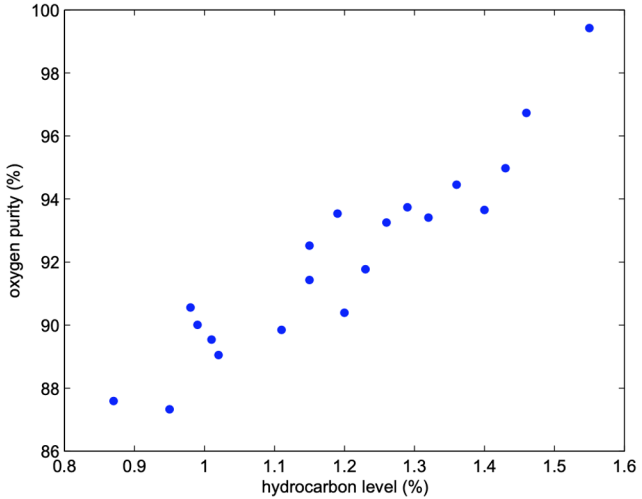
- Compute pseudo-inverse $\boldsymbol{X}^\dagger$ of $\boldsymbol{X}$
- return $\mathbf{w}_{\text{lin}} = \boldsymbol{X}^\dagger \boldsymbol{y}$

Example: oxygen and hydrocarbon levels

no.	x	y
1	0.99	90.01
2	1.02	89.05
3	1.15	91.43
4	1.29	93.74
5	1.46	96.73
6	1.36	94.45
7	0.87	87.59
8	1.23	91.77
9	1.55	99.42
10	1.40	93.65
11	1.19	93.54
12	1.15	92.52
13	0.98	90.56
14	1.01	89.54
15	1.11	89.85
16	1.20	90.39
17	1.26	93.25
18	1.32	93.41
19	1.43	94.98
20	0.95	87.33

y: purity (%) of oxygen produced in a chemical distillation process

x: hydrocarbons (%) that are present in the main condenser of the distillation unit



- training data:

$$\text{data matrix } X = \begin{bmatrix} 1 & 0.99 \\ 1 & 1.02 \\ 1 & 1.15 \\ 1 & 1.29 \\ 1 & 1.46 \\ 1 & 1.36 \\ 1 & 0.87 \\ 1 & 1.23 \\ 1 & 1.55 \\ 1 & 1.40 \\ 1 & 1.19 \\ 1 & 1.15 \\ 1 & 0.98 \\ 1 & 1.01 \\ 1 & 1.11 \\ 1 & 1.20 \\ 1 & 1.26 \\ 1 & 1.32 \\ 1 & 1.43 \\ 1 & 0.95 \end{bmatrix}, \text{target vector } \mathbf{y} = \begin{bmatrix} 90.01 \\ 89.05 \\ 91.43 \\ 93.74 \\ 96.73 \\ 94.45 \\ 87.59 \\ 91.77 \\ 99.42 \\ 93.65 \\ 93.54 \\ 92.52 \\ 90.56 \\ 89.54 \\ 89.85 \\ 90.39 \\ 93.25 \\ 93.41 \\ 94.98 \\ 87.33 \end{bmatrix}$$

- $\boldsymbol{X}^\top\boldsymbol{X}$ is invertible:

$$X^\top X = \begin{bmatrix} 20.00 & 23.92 \\ 23.92 & 29.29 \end{bmatrix} \Rightarrow (X^\top X)^{-1} = \begin{bmatrix} 2.15 & -1.76 \\ -1.76 & 1.47 \end{bmatrix}$$

- $(\boldsymbol{X}^\top\boldsymbol{X})\boldsymbol{X}^\top\boldsymbol{y}$ yields

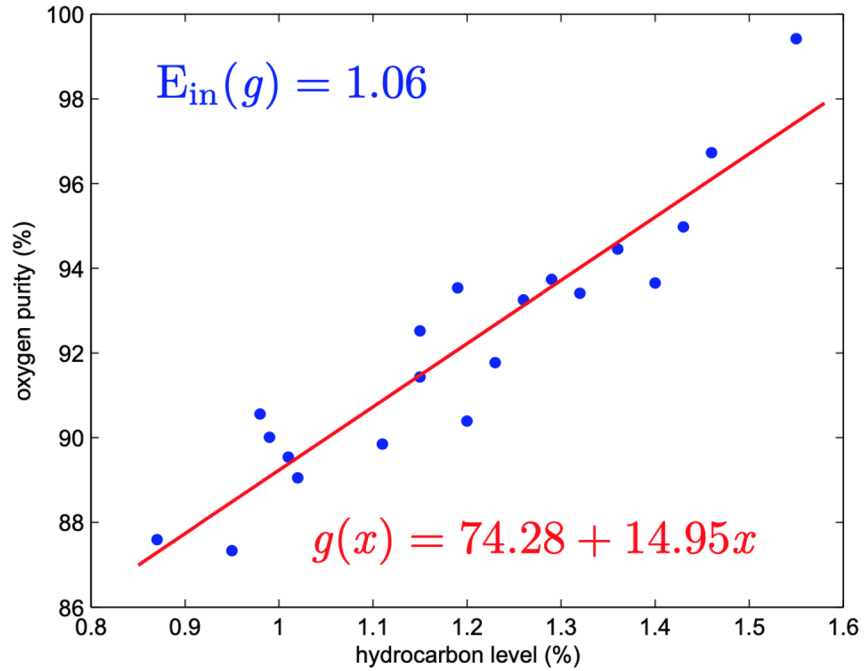
$$\mathbf{w}_{\text{lin}} = \begin{bmatrix} 74.28 \\ 14.95 \end{bmatrix}$$

- learned model:

$$g(x) = \mathbf{w}_{\text{lin}}^\top \mathbf{x}$$
$$= 74.28 + 14.95x$$

- error:

$$\mathrm{E}_{\text{in}}(g) = 1.06$$
$$\mathrm{E}_{\text{out}}(g) \approx 1.45$$



Remarks

- The linear regression algorithm
  - also called ordinary least squares (OLS)
  - provides the best linear unbiased estimator (BLUE)
- linear regression does not really look like learning
  - compared with PLA
  - hypothesis $\mathbf{w}_{\text{lin}}$ comes from an analytic solution (matrix inversion/multiplications) rather than iterative learning steps
- But learning has occured ("_____" learning)
  - as long as hypothesis $\mathbf{w}_{\text{lin}}$ has a decent $\boldsymbol{E}_{\text{out}}$

- Linear regression: a rare case
  - analytic formula for learning that is easy to evaluate
  - very popular
- there are methods for computing the pseudo-inverse directly
  - without _____ a matrix
  - numerically more stable than matrix inversion

---

[5] *e.g.* assume: the SVD of input matrix X is $X = U\Gamma V^\top$

- ▷ $U \in \mathbb{R}^{N\times\rho}$ satisfies $U^\top U = I_\rho$,
- ▷ $V \in \mathbb{R}^{(d+1)\times\rho}$ satisfies $V^\top V = I_\rho$,
- ▷ $\Gamma \in \rho \times \rho$ is a positive diagonal matrix with $\rho = \mathrm{rank}(X)$

▶ then $\mathbf{w}_{\mathrm{lin}} = V\Gamma^{-1}U^\top\mathbf{y}$ satisfies $X^\top X\mathbf{w}_{\mathrm{lin}} = X^\top\mathbf{y}$

Outline

- Linear Regression
  - Introduction
  - Linear Regression Algorithm
  - Interpretation via Hat Matrix
- Summary

Hat matrix H (aka projection matrix)

- H relates to in-sample and out-of-sample errors
  - one of analysis tools developed in statistics
- H maps the vector of observed values to the vector of fitted values
  - if $\boldsymbol{y}$ and $\hat{\mathbf{y}}$ denote observed and fitted values, respectively

$$\hat{\mathbf{y}} = H\mathbf{y}$$

  - H: observed values $\longmapsto$ fitted values
  - matrix H 'puts a hat' on $\boldsymbol{y}$, hence the name

Hat matrix in linear regression

- The linear regression weight vector $\mathbf{w}_{\text{lin}}$:
  - an attempt to map inputs $\mathbf{X}$ to outputs $\boldsymbol{y}$
- however, $\mathbf{w}_{\text{lin}}$ does not produce $\boldsymbol{y}$ directly
  - but produces an estimate
  - which differs from $\boldsymbol{y}$ due to _____ error
- substituting the expression for $\mathbf{w}_{\text{lin}}$
  - we get (assuming $\boldsymbol{X}^{\top}\boldsymbol{X}$ is invertible)

$$\hat{\mathbf{y}} = \underbrace{X(X^{\top}X)^{-1}X^{\top}}_{H \in \mathbb{R}^{N\times N}}\mathbf{y}$$

- therefore, estimate $\hat{\mathbf{y}}$ is a linear transformation of actual $\boldsymbol{y}$
  - through matrix multiplication with H where

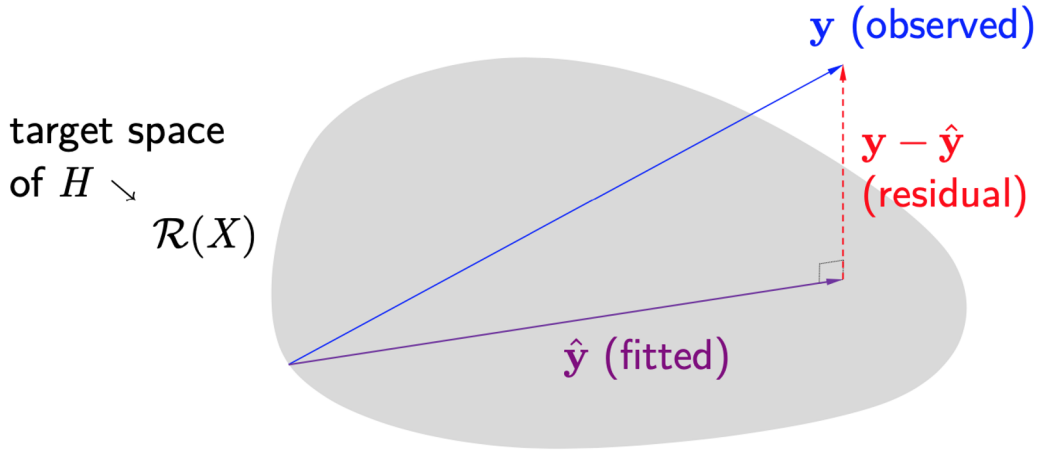
$$H = X(X^\top X)^{-1}X^\top$$

- bottom line:

$$\begin{aligned}\hat{\mathbf{y}} &= X\mathbf{w}_{\text{lin}} \\ &= X\underbrace{(X^\top X)^{-1}X^\top \mathbf{y}}_{\mathbf{w}_{\text{lin}}} \\ &= \underbrace{X(X^\top X)^{-1}X^\top}_{H}\mathbf{y}\end{aligned}$$

$\therefore$ $\hat{\mathbf{y}}$: orthogonal projection of $\mathbf{y}$ onto the _____ of X

Geometric interpretation



to minimize $||\mathbf{y} - \hat{\mathbf{y}}||$, $\hat{\mathbf{y}}$ should be the orthogonal projection of $\mathbf{y}$ onto $\mathcal{R}(X)$

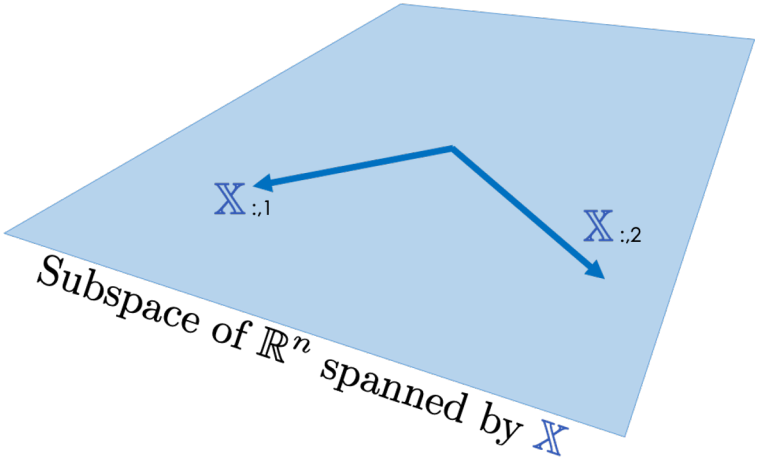
$$\hat{\mathbf{y}} = X\mathbf{w}_{\text{lin}} = H\mathbf{y}$$
$$H = X(X^\top X)^{-1}X^\top$$
$$\mathcal{R}(X) = \text{range (column span) of } X$$

$$\mathbf{y}, \hat{\mathbf{y}}, \mathcal{R}(X) \in \mathbb{R}^N$$
$$H \in \mathbb{R}^{N \times N}$$
$$X \in \mathbb{R}^{N \times (d+1)}$$
$$\mathbf{w}_{\text{lin}} \in \mathbb{R}^{d+1}$$

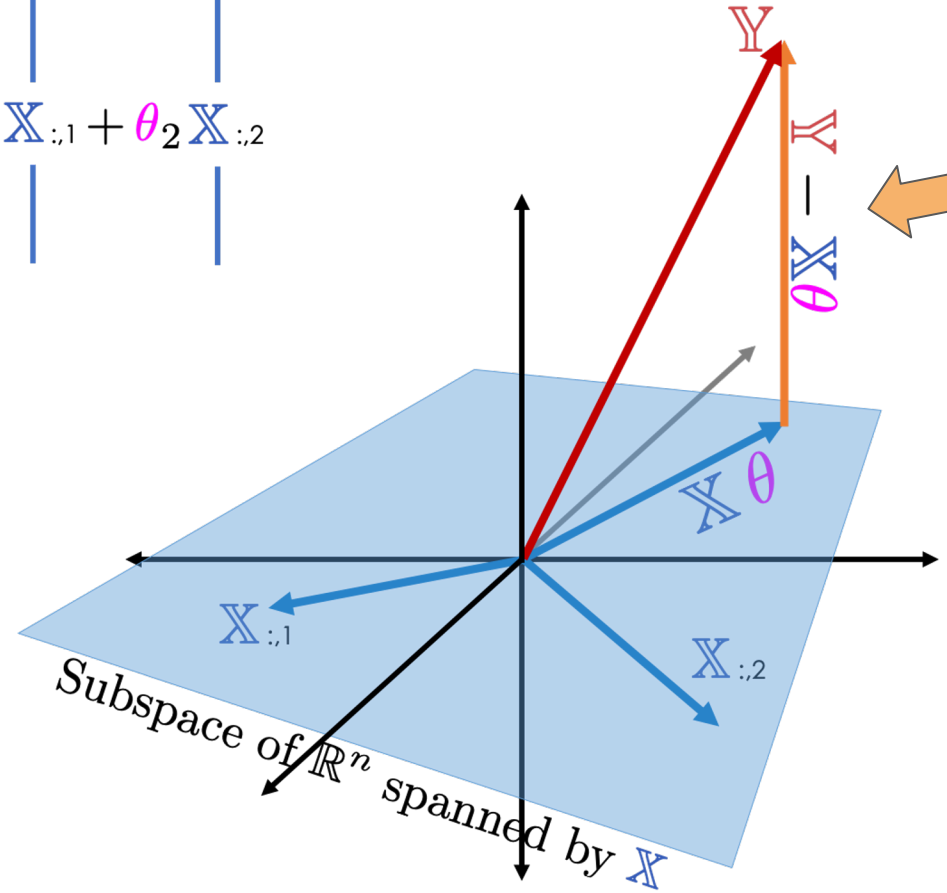
Span

Our prediction is a linear combination of the columns of X.

- The set of all possible linear combinations of the columns of X is called the **span** of the columns of X (denoted $\text{span}(\mathbb{X})$).
  - Also called the **column space**.
- Intuitively, this is all of the vectors you can "reach" using the columns of X.
- Since each column of X has length n, $\text{span}(\mathbb{X})$ is a subspace of $\mathbb{R}^n$.
- Our goal is to find the vector in that is closest to $\mathbb{Y}$.



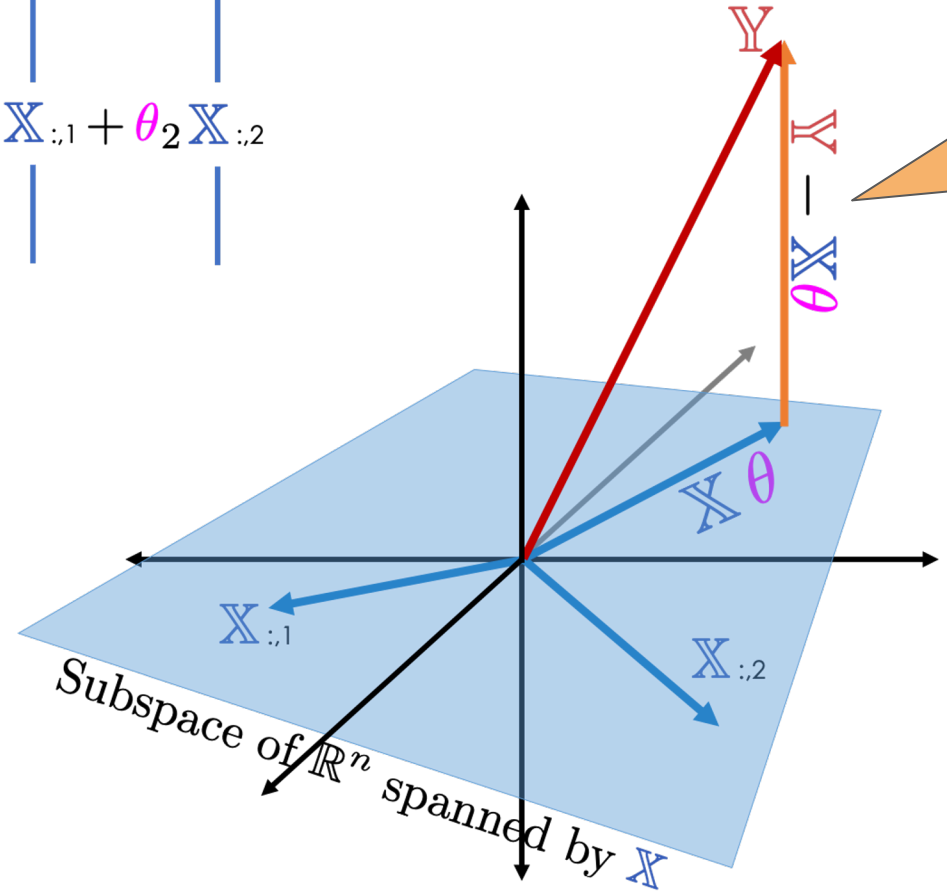
$$\begin{bmatrix} \\ \hat{\mathbb{Y}} \\ \\ \end{bmatrix}_{n \times 1} = \theta_1 \mathbb{X}_{:,1} + \theta_2 \mathbb{X}_{:,2}$$



Recall, this is the residual vector, $e = \mathbb{Y} - \hat{\mathbb{Y}}$.

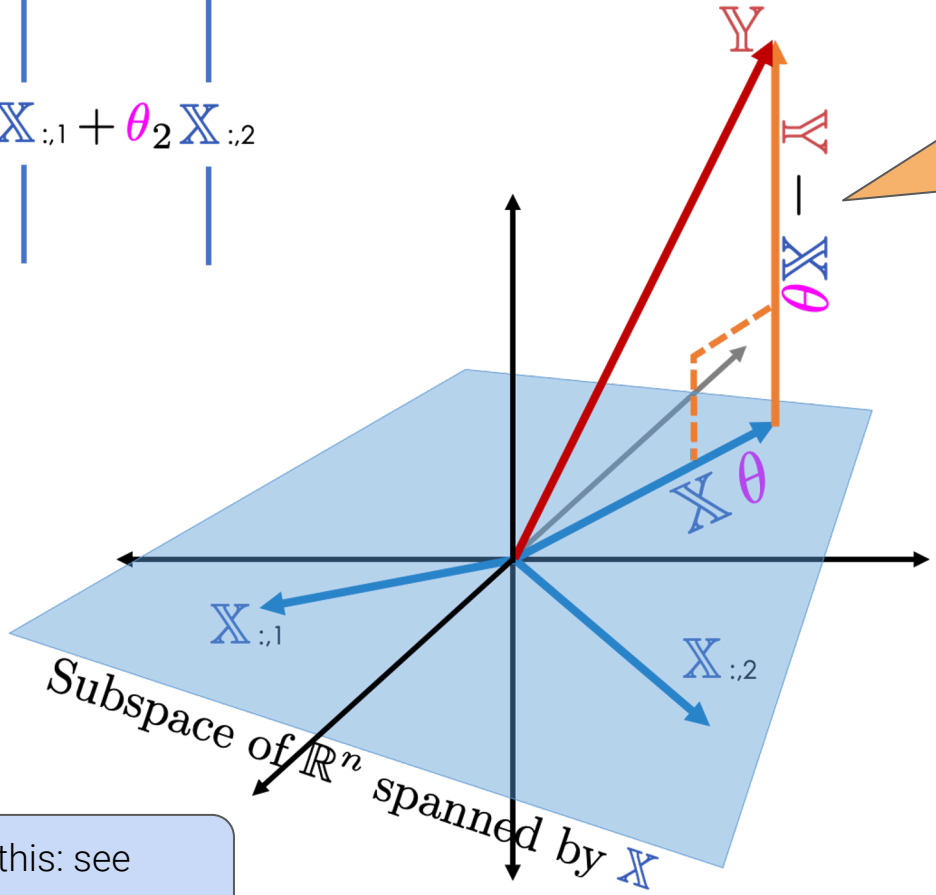
Our goal is to minimize the L_2 norm of the residual vector, i.e. we want our predictions to be "as close" to our true y values as possible.

$$\underset{n}{\begin{bmatrix} \\ \hat{\mathbb{Y}} \\ \\ \end{bmatrix}}_{1} = \theta_1 \mathbb{X}_{:,1} + \theta_2 \mathbb{X}_{:,2}$$



How do we minimize this distance – the norm of the residual vector (squared)?

$$\begin{bmatrix} \\ \hat{\mathbb{Y}} \\ \\ \end{bmatrix}_{n \times 1} = \theta_1 \mathbb{X}_{:,1} + \theta_2 \mathbb{X}_{:,2}$$

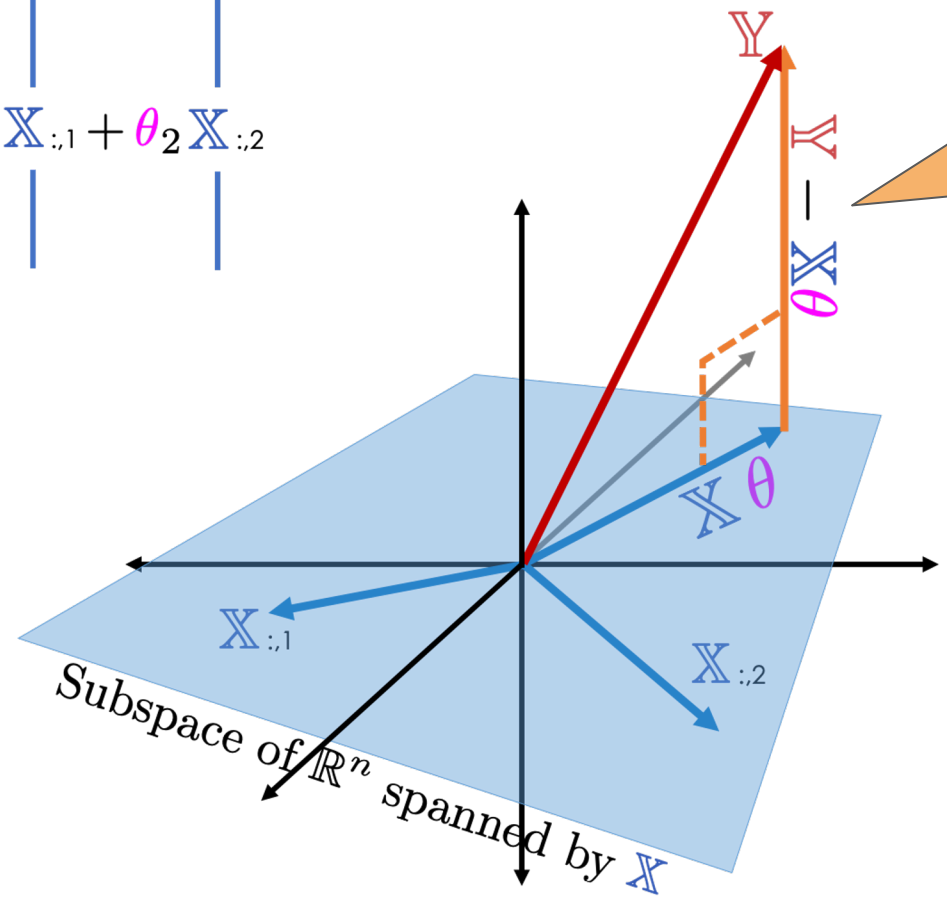


How do we minimize this distance – the norm of the residual vector (squared)?

The vector in span(X) that is closest to Y is the **orthogonal projection** of Y onto span(X).

We will not prove this: see Khan Academy.

$$\underset{n}{\begin{bmatrix} | \\ \hat{\mathbb{Y}} \\ | \end{bmatrix}}_{1} = \theta_1 \mathbb{X}_{:,1} + \theta_2 \mathbb{X}_{:,2}$$



How do we minimize this distance – the norm of the residual vector (squared)?

The vector in span(X) that is closest to Y is the **orthogonal projection** of Y onto span(X).

Thus, we should choose the θ that makes the residual vector **orthogonal** to span(X).

Residuals are orthogonal to the span of X

We want the θ such that the residual vector is orthogonal to $\text{span}(\mathbb{X})$.

By the definition of orthogonality: $\mathbb{X}^T \left(\mathbb{Y} - \mathbb{X}\hat{\theta} \right) = 0$

Still the zero vector!

Rearranging: $\mathbb{X}^T \mathbb{Y} - \mathbb{X}^T \mathbb{X}\hat{\theta} = 0$

The **normal equation**: $\mathbb{X}^T \mathbb{X}\hat{\theta} = \mathbb{X}^T \mathbb{Y}$

Assuming $\mathbb{X}^T \mathbb{X}$ is full rank: $\boxed{\hat{\theta} = \left(\mathbb{X}^T \mathbb{X} \right)^{-1} \mathbb{X}^T \mathbb{Y}}$

$$\hat{\theta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$$

This result is so important, it deserves its own slide.
It is the **least squares estimate** for θ.

let $H \in \mathbb{R}^{N \times N}$ (*i.e.*, H is a square matrix):

- H is called a projection/hat matrix iff $H^2 = H$
- two basic properties of hat matrix H
  - symmetric: $H^\top = H$
  - idempotent: H^k = H (no effect on vectors already on space)
- for hat matrix H and identity matrix $I \in \mathbb{R}^{N \times N}$
  - $I - H$ is also a hat matrix:

$$(I - H)^2 = I - 2H + H^2 = I - H$$

  - $H^\top (I - H) = 0 \Rightarrow$ target spaces are orthogonal

- for hat matrix $H = X(X^\top X)^{-1}X^\top$ where $X \in \mathbb{R}^{N\times(d+1)}$
  - H : orthogonal projection onto the column space of X
  - range $\mathcal{R}(X)$ is thus called the target space of H
  - $\text{trace}(H) =$ _____

$$\begin{aligned}\text{trace}(H) &= \text{trace}(X(X^\top X)^{-1}X^\top)\\ &= \text{trace}(X^\top X(X^\top X)^{-1})\\ &= \text{trace}(I)\\ &= \_\_\_\_\_\end{aligned}$$

    - ▷ note: $X^\top X \in \mathbb{R}^{(d+1)\times(d+1)}$ and thus $I \in \mathbb{R}^{(d+1)\times(d+1)}$

- recall: $\text{trace}(H)$ is sum of diagonal elements in H
  - property: $\text{trace}(AB) = \text{trace}(BA)$ for two matrices A and B

Outline

- Linear Classification
  - Binary Classification
  - Example: Handwritten Digit Recognition
- Linear Regression
  - Introduction
  - Linear Regression Algorithm
  - Interpretation via Hat Matrix
- Summary

Summary

- linear models
  - used for classification, regression, probability estimation
  - have small d_{VC} and generalize well $\Rightarrow$ first practical choice
  - use the 'signal': $\sum_{i=0}^{d} w_i x_i = \mathbf{w}^\top \mathbf{x}$

- linear classification

$$\mathcal{Y} = \{-1, +1\}$$
$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x})$$
$$\text{e}(\hat{y}, y) = [\![\hat{y} \neq y]\!]$$

  - NP-hard in general

- linear regression

$$\mathcal{Y} = \mathbb{R}$$
$$h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$$
$$\text{e}(\hat{y}, y) = (\hat{y} - y)^2$$

  - efficient closed-form solution

- linear regression algorithm (1-step learning): $\boxed{\mathbf{w} = (X^\top X)^{-1} X^\top \mathbf{y}}$