

18-661 Introduction to Machine Learning

Linear Regression – Part I

Spring 2025

ECE – Carnegie Mellon University

1. Recap of MLE/MAP
2. Linear Algebra Review
3. Linear Regression
 - Formulation
 - Univariate Solution
 - Multivariate Solution
 - Probabilistic Interpretation

Recap of MLE/MAP

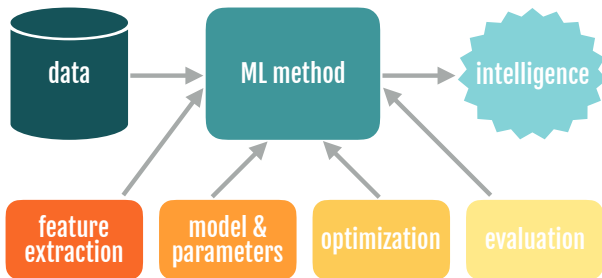
- Scenario: You find a coin on the ground.



- *You ask yourself: Is this a fair or biased coin? What is the probability that I will flip a heads?*

- You flip the coin 10 times ...
- It comes up as 'H' 8 times and 'T' 2 times
- Can we learn the bias of the coin from this data?

Recall: Machine Learning Pipeline



Two approaches that we discussed:

- Maximum likelihood Estimation (MLE)
- Maximum a posteriori Estimation (MAP)

Maximum Likelihood Estimation (MLE)

- **Data:** Observed set D of n_H heads and n_T tails
- **Model:** Each flip follows a Bernoulli distribution

$$P(H) = \theta, P(T) = 1 - \theta, \theta \in [0, 1]$$

Thus, the likelihood of observing sequence D is

$$P(D | \theta) = \theta^{n_H} (1 - \theta)^{n_T}$$

- **Question:** Given this model and the data we've observed, can we calculate an estimate of θ ?
- **MLE:** Choose θ that maximizes the *likelihood* of the observed data

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

MAP for Dogecoin

$$\hat{\theta}_{MAP} = \arg \max_{\theta} P(D | \theta)P(\theta) = \arg \max_{\theta} P(\theta | D)$$

- Recall that $P(D | \theta) = \theta^{n_H}(1 - \theta)^{n_T}$
- How should we set the prior, $P(\theta)$?
- Common choice for a binomial likelihood is to use the **Beta distribution**, $\theta \sim \text{Beta}(\alpha, \beta)$:

$$P(\theta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}, \text{ where } B(\alpha, \beta) = \int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1} d\theta$$

- Interpretation: α = number of expected heads, β = number of expected tails. Larger value of $\alpha + \beta$ denotes more confidence (and smaller variance).

Putting It All Together

$$\hat{\theta}_{MLE} = \frac{n_H}{n_H + n_T}$$
$$\hat{\theta}_{MAP} = \frac{\alpha + n_H - 1}{\alpha + \beta + n_H + n_T - 2}$$

Learning involves:

- Collect some data
 - e.g., coin flips
- Set up the problem: Choose a model / loss function
 - e.g., Bernoulli model, data likelihood, prior distribution
- Solve the problem: Choose an optimization procedure
 - e.g., set derivative of log to zero and solve to find MLE/MAP

Key idea: these are *choices*. It's important to understand the implications of these choices and evaluate their trade-offs for the problem at hand.

Bayesians vs. Frequentists

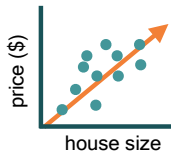
You are no good when sample is small



You give a different answer for different priors

Linear Algebra Review

Data Can Be Compactly Represented by Matrices



- Learn parameters (w_1, w_0) of the orange line $y = w_1x + w_0$

Sq.ft

House 1: $1000 \times w_1 + w_0 = 200,000$

House 2: $2000 \times w_1 + w_0 = 350,000$

- Can represent compactly in matrix notation

$$\begin{bmatrix} 1000 & 1 \\ 2000 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_0 \end{bmatrix} = \begin{bmatrix} 200,000 \\ 350,000 \end{bmatrix}$$

Matrix Inverse

The **inverse** of a matrix $A \in \mathbb{R}^{n \times n}$ is a matrix $A^{-1} \in \mathbb{R}^{n \times n}$ such that:

$$AA^{-1} = A^{-1}A = I_n$$

- If A^{-1} exists, then A is called invertible or non-singular
- Matrix A is invertible iff $\det(A) \neq 0$
- If A^{-1} exists, then it is unique
- Can be used to solve the house-price prediction problem:

$$\begin{bmatrix} 1000 & 1 \\ 2000 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_0 \end{bmatrix} = \begin{bmatrix} 200,000 \\ 350,000 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} w_1 \\ w_0 \end{bmatrix} = \left(\begin{bmatrix} 1000 & 1 \\ 2000 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 200,000 \\ 350,000 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} w_1 \\ w_0 \end{bmatrix} = \begin{bmatrix} 150 \\ 50,000 \end{bmatrix} \quad (3)$$

Norms and Loss Functions

- You could have data from many houses

$$\begin{matrix} \begin{bmatrix} 1000 & 1 \\ 2000 & 1 \\ 1500 & 1 \\ \vdots & \vdots \\ 2500 & 1 \end{bmatrix} & & \begin{bmatrix} w_1 \\ w_0 \end{bmatrix} = & & \begin{bmatrix} 200,000 \\ 350,000 \\ 300,000 \\ \vdots \\ 450,000 \end{bmatrix} \\ A & \times & w = & & y \end{matrix}$$

- There isn't a $w = [w_1, w_0]^T$ that will satisfy all equations
- Want to find w that minimizes the difference between Aw, y
- But since this a vector, we need an operator that can map the vector $y - Aw$ to a scalar

Norms and Loss Functions

- A vector **norm** is any function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with
 - $f(x) \geq 0$ and $f(x) = 0 \iff x = 0$
 - $f(ax) = |a|f(x)$ for $a \in \mathbb{R}$
 - $f(x+y) \leq f(x) + f(y)$
- e.g., ℓ_2 norm: $\|x\|_2 = \sqrt{x^\top x} = \sqrt{\sum_{i=1}^n x_i^2}$
- e.g., ℓ_1 norm: $\|x\|_1 = \sum_{i=1}^n |x_i|$
- **Question:** What is the ℓ_1 norm of $y - Aw$ for the following problem?

$$\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1.5 & 1 \\ 2.5 & 1 \end{bmatrix} \quad \times \quad \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 2 \\ 3.5 \\ 3 \\ 4.5 \end{bmatrix}$$

$A \qquad \qquad \qquad w \qquad \qquad \qquad y$

- **Answer:** $\|y - Aw\|_1 = 0.5$

Eigenvalues and Eigenvectors

- For $A \in \mathbb{R}^{n \times n}$, λ is an eigenvalue and $x \neq 0$ is an eigenvector if $Ax = \lambda x$.
- Eigenvalues are the roots of $\det(A - \lambda I_n) = 0$
- Eigenvectors are non-zero solutions of $Ax = \lambda x$
- Viewing A as a linear transformation
 - The vectors that remain unchanged and only get re-scaled are the eigenvectors.
 - Their scaling factors are the eigenvalues!
- **Question:** Find the eigenvalues and eigenvectors of

$$\begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}$$

Eigenvalue Decomposition

- Group the eigenvectors and eigenvalues into the following matrices.

$$P = \begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} 5 & 0 \\ 0 & -1 \end{bmatrix}$$

- If the eigenvectors are linearly independent, we can express A as

$$\begin{aligned} A &= P\Lambda P^{-1} \\ &= \begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 5 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}^{-1} \end{aligned}$$

Eigenvalue Decomposition

- Why is this useful?
- Suppose we want to find powers of A , eg. A^4
- One option, that is quite tedious is:

$$A^4 = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}$$

- Instead we could use the eigenvalue decomposition

$$\begin{aligned} A^4 &= P\Lambda P^{-1}P\Lambda P^{-1}P\Lambda P^{-1}P\Lambda P^{-1} \\ &= P\Lambda^4 P^{-1} \\ &= \begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 5^4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}^{-1} \end{aligned}$$

Singular value decomposition (SVD)

- EVD only works for square, diagonalizable matrices
- SVD works for matrices of any size! It decomposes $A \in \mathbb{R}^{m \times n}$ as follows.

$$A = U \Sigma V^T,$$

- $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices (i.e. $U^T = U^{-1}$)
- $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix with *singular values* of A denoted by σ_i appearing by non-increasing order: $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m,n)} \geq 0$.
- The squared singular values of A are the eigenvalues of the matrix AA^T or $A^T A$, i.e., $\sigma_i(A) = \sqrt{\lambda_i(AA^T)} = \sqrt{\lambda_i(A^T A)}$
- V is the matrix of eigenvectors of $A^T A$
- U is the matrix of eigenvectors of AA^T

Linear Regression

Recap of MLE/MAP

Linear Algebra Review

Linear Regression

- Formulation

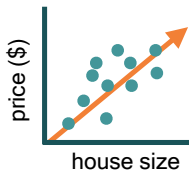
- Univariate Solution

- Multivariate Solution

- Probabilistic Interpretation

Task 1: Regression

How much should you sell your house for?



input: houses & features **learn:** $x \rightarrow y$ relationship **predict:** y (*continuous*)

Course Covers: Linear/Ridge Regression, Loss Function, SGD, Feature Scaling, Regularization, Cross Validation

Supervised learning

In a supervised learning problem, you have access to input variables (X) and outputs (Y), and the goal is to predict an output given an input

- Examples:
 - **Housing prices (Regression)**: predict the price of a house based on features (size, location, etc)
 - **Cat vs. Dog (Classification)**: predict whether a picture is of a cat or a dog

Predicting a continuous outcome variable:

- Predicting a company's future stock price using its profit and other financial info
- Predicting annual rainfall based on local flora and fauna
- Predicting distance from a traffic light using LIDAR measurements

Magnitude of the error matters:

- We can measure 'closeness' of prediction and labels, leading to different ways to evaluate prediction errors.
 - Predicting stock price: better to be off by 1\$ than by 20\$
 - Predicting distance from a traffic light: better to be off 1 m than by 10 m
- We should choose learning models and algorithms accordingly.

Predicting House Prices: Collecting Data


3620 South BUDLONG
 Los Angeles, CA 90007
 (Map icon) (Street View)

\$1,210,000
 Last Sold Price
 Built: 1980 Lot Size: 6,840 Sq. Ft. Sold On: Jul 26, 2013

14
 Beds
 6 Bath

4,418 sq. ft.
 3947 - 3971 sq. ft.
 Built On: Jul 26, 2013

[View Photos](#)
[Property Details](#)
[Floor Plans](#)
[Property History](#)
[Public Records](#)
[Auction](#)
[Alerts](#)




14 Photos

Five unit apartment complex within 2 blocks of USC campus. Gets ML. Great for students (most student leases have parents as guarantors). Almost USC students live off campus, so being unit is fine that are always in demand. Situated on a quiet, tree-lined lot, and bordered from an apartment complex. This complex was recently renovated, and has great security (multiple units up, and not ML), and 12 parking spaces. It is within a mile (Department of Public Safety) and Caltrans (College parking area). This is a great income generating property, not to be missed.

Property Type: **Multi-Family**
 Community: **Downtown Los Angeles**
 MLSPID: **20174741**

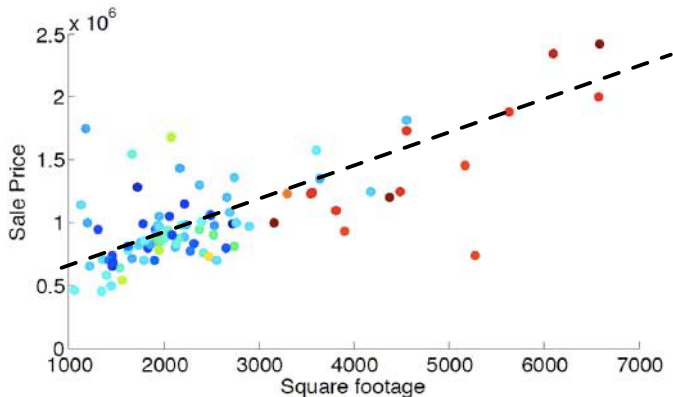
Style: **Two Level, Low Rise**
 Security: **See Remarks**

Property Details for 3625 South SUDLONG, Los Angeles, CA 90007

Don't miss out on this! Visit www.pearsoncmg.com today!

Interior Features			
Kitchen Information		Laundry Information	
<ul style="list-style-type: none"> • Pantry • Dish Range 		<ul style="list-style-type: none"> • Stack Laundry 	
Multi-Unit Information		Heating & Cooling	
		<ul style="list-style-type: none"> • Walk Cooling (3/10/0) 	
Community Features		Unit's Information	
<ul style="list-style-type: none"> • Unit's Calendar (10/1/0) 		<ul style="list-style-type: none"> • Heating Rent: \$1,320 	
Multi-Family Information		Unit's Details	
<ul style="list-style-type: none"> • # Locked: 0 • # of Units: 1 • Group Type: Water • Tenant Pays Electricity, Tenant Pays Gas 		<ul style="list-style-type: none"> • # of Bath: 1 • # of Kitchen: 1 • Monthly Rent: \$2,200 	
Unit's Information		Unit's Information	
<ul style="list-style-type: none"> • # of Bath: 0 • # of Kitchen: 1 • Unit's Status: 0 • Monthly Rent: \$1,200 		<ul style="list-style-type: none"> • Unimproved • Unit's Status: 0 • # of Bath: 1 • Unit's Status: 0 	
Property / Lot Details		Monthly Rent: \$2,200	
Property Features		Property Information	
<ul style="list-style-type: none"> • Automatic Gate, Car Wash, Pool 		<ul style="list-style-type: none"> • Automatic Gate, Lawn, Scheduling • Owner Lot Area: 10,000 Sq. Ft. 	
Lot Information		Property Information	
<ul style="list-style-type: none"> • Lot Size (Sq. Ft.): 0.50 • Lot Size (Acres): 0.0119 • Lot Size (Square Feet): 0.0000 		<ul style="list-style-type: none"> • Lot's Status: 0 • Square Footage: 0.0000 	
Parking / Storage, Exterior Features, Utilities & Financing			
Parking Information		Utility Information	
<ul style="list-style-type: none"> • # of Parking Spaces (Total): 10 • # of Parking Spaces (Per Unit): 1 • # of Parking Spaces (Per Unit): 1 		<ul style="list-style-type: none"> • Sewer: 0.0000 • Water: 0.0000 • Gas: 0.0000 • Electric: 0.0000 • Sewer: 0.0000 • Water: 0.0000 • Gas: 0.0000 • Electric: 0.0000 	
Building Information		Financial Information	
<ul style="list-style-type: none"> • Total Floor: 2 		<ul style="list-style-type: none"> • Capital Cost: \$1,000,000 • Annual Rental Income: \$1,000,000 • Gross Rent Multiplier: 11.20 	
Location Details, Map, Interiors & Listing Information			
Location Information		Expense Information	
<ul style="list-style-type: none"> • Cross Street: 10000 		<ul style="list-style-type: none"> • Operating: \$10,000 	
Listing Information		Listing Information	
<ul style="list-style-type: none"> • Listing Title: Call, Call To Listing: Call • Buyer: Listing: Call 		<ul style="list-style-type: none"> • Listing Title: Call, Call To Listing: Call • Buyer: Listing: Call 	

Correlation between Square Footage and Sale Price



- Sale price \approx price_per_sqft \times square_footage + fixed_expense
- Learn parameters (w_0 , w_1) of the dotted line $y = w_1x + w_0$

Reduce Prediction Error

How to measure prediction errors?

sqft	sale price	prediction	abs error	squared error
2000	810K	720K	90K	8100
2100	907K	800K	107K	107^2
1100	312K	350K	38K	38^2
5500	2,600K	2,600K	0	0
...	...			

- **absolute** difference (ℓ_1 norm): $|\text{prediction} - \text{sale price}|$.
- **squared** difference (ℓ_2 norm): $(\text{prediction} - \text{sale price})^2$
[differentiable!].

Minimize Squared Errors

Our model:

Sale_price =

price_per_sqft \times square_footage + fixed_expense + unexplainable_stuff

Training data:

sqft	sale price	prediction	error	squared error
2000	810K	720K	90K	8100
2100	907K	800K	107K	107^2
1100	312K	350K	38K	38^2
5500	2,600K	2,600K	0	0
...	...			
Total				$8100 + 107^2 + 38^2 + 0 + \dots$

Aim:

Adjust price_per_sqft and fixed_expense such that the sum of the squared error is minimized — i.e., the unexplainable_stuff is minimized.

Linear Regression

Setup:

- **Input:** $\mathbf{x} \in \mathbb{R}^D$ (covariates, predictors, features, etc)
- **Output:** $y \in \mathbb{R}$ (responses, targets, outcomes, outputs, etc)
- **Model:** $f: \mathbf{x} \rightarrow y$, with $f(\mathbf{x}) = w_0 + \sum_{d=1}^D w_d x_d = w_0 + \mathbf{w}^\top \mathbf{x}$.
 - $\mathbf{w} = [w_1 \ w_2 \ \cdots \ w_D]^\top$: *weights, parameters, or parameter vector*
 - w_0 is called *bias*.
 - Sometimes, we also call $\tilde{\mathbf{w}} = [w_0 \ w_1 \ w_2 \ \cdots \ w_D]^\top$ parameters.
- **Training data:** $\mathcal{D} = \{(\mathbf{x}_n, y_n), n = 1, 2, \dots, N\}$

Minimize the Residual Sum of Squares:

$$RSS(\tilde{\mathbf{w}}) = \sum_{n=1}^N [y_n - f(\mathbf{x}_n)]^2 = \sum_{n=1}^N [y_n - (w_0 + \sum_{d=1}^D w_d x_{nd})]^2$$

Recap of MLE/MAP

Linear Algebra Review

Linear Regression

Formulation

Univariate Solution

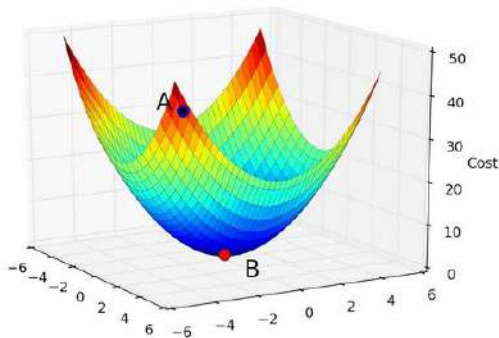
Multivariate Solution

Probabilistic Interpretation

A Simple Case: x Is One-dimensional ($D=1$)

Residual sum of squares:

$$RSS(\tilde{\mathbf{w}}) = \sum_n [y_n - f(\mathbf{x}_n)]^2 = \sum_n [y_n - (w_0 + w_1 x_n)]^2$$



What kind of function is this? CONVEX (has a unique global minimum)

A Simple Case: x Is One-dimensional ($D=1$)

Residual sum of squares:

$$RSS(\mathbf{w}) = \sum_n [y_n - f(\mathbf{x}_n)]^2 = \sum_n [y_n - (w_0 + w_1 x_n)]^2$$

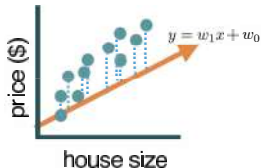


Figure 2: RSS is the sum of squares of the dotted lines

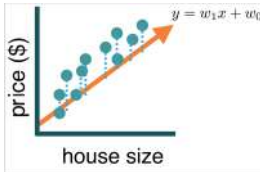


Figure 3: Adjust (w_0, w_1) to reduce RSS



Figure 4: RSS minimized at (w_0^*, w_1^*)

A Simple Case: x Is One-dimensional ($D=1$)

Residual sum of squares:

$$RSS(\tilde{\mathbf{w}}) = \sum_n [y_n - f(\mathbf{x}_n)]^2 = \sum_n [y_n - (w_0 + w_1 x_n)]^2$$

Stationary points:

Take derivative with respect to parameters and set it to zero

$$\frac{\partial RSS(\tilde{\mathbf{w}})}{\partial w_0} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] = 0,$$

$$\frac{\partial RSS(\tilde{\mathbf{w}})}{\partial w_1} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] x_n = 0.$$

A Simple Case: x Is One-dimensional ($D=1$)

$$\frac{\partial RSS(\tilde{\mathbf{w}})}{\partial w_0} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] = 0$$

$$\frac{\partial RSS(\tilde{\mathbf{w}})}{\partial w_1} = 0 \Rightarrow -2 \sum_n [y_n - (w_0 + w_1 x_n)] x_n = 0$$

Simplify these expressions to get the “Normal Equations”:

$$\sum y_n = N w_0 + w_1 \sum x_n$$

$$\sum x_n y_n = w_0 \sum x_n + w_1 \sum x_n^2$$

Solving the system we obtain the **least squares coefficient estimates**:

$$w_1 = \frac{\sum (x_n - \bar{x})(y_n - \bar{y})}{\sum (x_i - \bar{x})^2} \quad \text{and} \quad w_0 = \bar{y} - w_1 \bar{x}$$

where $\bar{x} = \frac{1}{N} \sum_n x_n$ and $\bar{y} = \frac{1}{N} \sum_n y_n$.

Recap of MLE/MAP

Linear Algebra Review

Linear Regression

Formulation

Univariate Solution

Multivariate Solution

Probabilistic Interpretation

Least Mean Squares: \mathbf{x} Is D -dimensional

sqft (1000's)	bedrooms	bathrooms	sale price (100k)
1	2	1	2
2	2	2	3.5
1.5	3	2	3
2.5	4	2.5	4.5

RSS($\tilde{\mathbf{w}}$) in matrix form:

$$RSS(\tilde{\mathbf{w}}) = \sum_n [y_n - (w_0 + \sum_d w_d x_{nd})]^2 = \sum_n [y_n - \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_n]^2,$$

where we have redefined some variables (by augmenting)

$$\tilde{\mathbf{x}} \leftarrow [1 \ x_1 \ x_2 \ \dots \ x_D]^\top, \quad \tilde{\mathbf{w}} \leftarrow [w_0 \ w_1 \ w_2 \ \dots \ w_D]^\top$$

What is $\tilde{\mathbf{x}}$ for the first house? $[1, 1, 2, 1]^\top$

Least Mean Squares: \mathbf{x} Is D -dimensional

$RSS(\tilde{\mathbf{w}})$ in matrix form:

$$RSS(\tilde{\mathbf{w}}) = \sum_n [y_n - (w_0 + \sum_d w_d x_{nd})]^2 = \sum_n [y_n - \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_n]^2,$$

where we have redefined some variables (by augmenting)

$$\tilde{\mathbf{x}} \leftarrow [1 \ x_1 \ x_2 \ \dots \ x_D]^\top, \quad \tilde{\mathbf{w}} \leftarrow [w_0 \ w_1 \ w_2 \ \dots \ w_D]^\top$$

which leads to

$$\begin{aligned} RSS(\tilde{\mathbf{w}}) &= \sum_n (y_n - \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_n)(y_n - \tilde{\mathbf{x}}_n^\top \tilde{\mathbf{w}}) \\ &= \sum_n \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_n \tilde{\mathbf{x}}_n^\top \tilde{\mathbf{w}} - 2y_n \tilde{\mathbf{x}}_n^\top \tilde{\mathbf{w}} + \text{const.} \\ &= \left\{ \tilde{\mathbf{w}}^\top \left(\sum_n \tilde{\mathbf{x}}_n \tilde{\mathbf{x}}_n^\top \right) \tilde{\mathbf{w}} - 2 \left(\sum_n y_n \tilde{\mathbf{x}}_n^\top \right) \tilde{\mathbf{w}} \right\} + \text{const.} \end{aligned}$$

RSS($\tilde{\mathbf{w}}$) in New Notations

From previous slide:

$$RSS(\tilde{\mathbf{w}}) = \left\{ \tilde{\mathbf{w}}^\top \left(\sum_n \tilde{\mathbf{x}}_n \tilde{\mathbf{x}}_n^\top \right) \tilde{\mathbf{w}} - 2 \left(\sum_n y_n \tilde{\mathbf{x}}_n^\top \right) \tilde{\mathbf{w}} \right\} + \text{const.}$$

Design matrix and target vector:

$$\tilde{\mathbf{X}} = \begin{pmatrix} \tilde{\mathbf{x}}_1^\top \\ \tilde{\mathbf{x}}_2^\top \\ \vdots \\ \tilde{\mathbf{x}}_N^\top \end{pmatrix} \in \mathbb{R}^{N \times (D+1)}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \in \mathbb{R}^N$$

Compact expression:

$$RSS(\tilde{\mathbf{w}}) = \|\tilde{\mathbf{X}}\tilde{\mathbf{w}} - \mathbf{y}\|_2^2 = \left\{ \tilde{\mathbf{w}}^\top \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \tilde{\mathbf{w}} - 2 \left(\tilde{\mathbf{X}}^\top \mathbf{y} \right)^\top \tilde{\mathbf{w}} \right\} + \text{const}$$

Example: $RSS(\tilde{\mathbf{w}})$ in Compact Form

sqft (1000's)	bedrooms	bathrooms	sale price (100k)
1	2	1	2
2	2	2	3.5
1.5	3	2	3
2.5	4	2.5	4.5

Design matrix and target vector:

$$\tilde{\mathbf{X}} = \begin{pmatrix} \tilde{\mathbf{x}}_1^\top \\ \tilde{\mathbf{x}}_2^\top \\ \vdots \\ \tilde{\mathbf{x}}_N^\top \end{pmatrix} = \begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 1.5 & 3 & 2 \\ 1 & 2.5 & 4 & 2.5 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 2 \\ 3.5 \\ 3 \\ 4.5 \end{bmatrix}$$

. Compact expression:

$$RSS(\tilde{\mathbf{w}}) = \|\tilde{\mathbf{X}}\tilde{\mathbf{w}} - \mathbf{y}\|_2^2 = \left\{ \tilde{\mathbf{w}}^\top \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \tilde{\mathbf{w}} - 2 \left(\tilde{\mathbf{X}}^\top \mathbf{y} \right)^\top \tilde{\mathbf{w}} \right\} + \text{const}$$

Solution in Matrix Form

Compact expression

$$RSS(\tilde{\mathbf{w}}) = ||\tilde{\mathbf{X}}\tilde{\mathbf{w}} - \mathbf{y}||_2^2 = \left\{ \tilde{\mathbf{w}}^\top \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \tilde{\mathbf{w}} - 2 \left(\tilde{\mathbf{X}}^\top \mathbf{y} \right)^\top \tilde{\mathbf{w}} \right\} + \text{const}$$

Gradients of Linear and Quadratic Functions

- $\nabla_{\mathbf{x}}(\mathbf{b}^\top \mathbf{x}) = \mathbf{b}$
- $\nabla_{\mathbf{x}}(\mathbf{x}^\top \mathbf{A} \mathbf{x}) = 2\mathbf{A}\mathbf{x}$ (symmetric \mathbf{A})

Normal equation

$$\nabla_{\tilde{\mathbf{w}}} RSS(\tilde{\mathbf{w}}) = 2\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \tilde{\mathbf{w}} - 2\tilde{\mathbf{X}}^\top \mathbf{y} = 0$$

This leads to the **least-mean-squares** (LMS) solution

$$\tilde{\mathbf{w}}^{LMS} = \left(\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \right)^{-1} \tilde{\mathbf{X}}^\top \mathbf{y}$$

Example: $RSS(\tilde{\mathbf{w}})$ in Compact Form

sqft (1000's)	bedrooms	bathrooms	sale price (100k)
1	2	1	2
2	2	2	3.5
1.5	3	2	3
2.5	4	2.5	4.5

Write the **least-mean-squares** (LMS) solution

$$\tilde{\mathbf{w}}^{LMS} = \left(\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} \right)^{-1} \tilde{\mathbf{X}}^T \mathbf{y}$$

Can use solvers in Matlab, Python etc., to compute this for any given $\tilde{\mathbf{X}}$ and \mathbf{y} .

Exercise: $RSS(\tilde{\mathbf{w}})$ in Compact Form

Using the general **least-mean-squares** (LMS) solution

$$\tilde{\mathbf{w}}^{LMS} = \left(\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \right)^{-1} \tilde{\mathbf{X}}^\top \mathbf{y}$$

recover the uni-variate solution that we had computed earlier:

$$w_1 = \frac{\sum (x_n - \bar{x})(y_n - \bar{y})}{\sum (x_i - \bar{x})^2} \quad \text{and} \quad w_0 = \bar{y} - w_1 \bar{x}$$

where $\bar{x} = \frac{1}{N} \sum_n x_n$ and $\bar{y} = \frac{1}{N} \sum_n y_n$.

Exercise: $RSS(\tilde{\mathbf{w}})$ in Compact Form

For the 1-D case, the **least-mean-squares** solution is

$$\begin{aligned}\tilde{\mathbf{w}}^{LMS} &= (\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^\top \mathbf{y} \\&= \left(\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \end{bmatrix} \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & \dots \\ 1 & x_N \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix} \\&= \left(\begin{bmatrix} N & N\bar{x} \\ N\bar{x} & \sum_n x_n^2 \end{bmatrix} \right)^{-1} \begin{bmatrix} \sum_n y_n \\ \sum_n x_n y_n \end{bmatrix} \\ \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} &= \frac{1}{\sum (x_i - \bar{x})^2} \begin{bmatrix} \bar{y} \sum (x_i - \bar{x})^2 - \bar{x} \sum (x_n - \bar{x})(y_n - \bar{y}) \\ \sum (x_n - \bar{x})(y_n - \bar{y}) \end{bmatrix}\end{aligned}$$

where $\bar{x} = \frac{1}{N} \sum_n x_n$ and $\bar{y} = \frac{1}{N} \sum_n y_n$.

Recap of MLE/MAP

Linear Algebra Review

Linear Regression

Formulation

Univariate Solution

Multivariate Solution

Probabilistic Interpretation

Why Minimize the RSS?

Probabilistic interpretation

- **Noisy observation model** for generating the dataset:

$$Y = w_0 + w_1 X + \eta$$

where $\eta \sim N(0, \sigma^2)$ is a Gaussian random variable

- Conditional likelihood of one training sample:

$$p(y_n|x_n) = N(w_0 + w_1 x_n, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{[y_n - (w_0 + w_1 x_n)]^2}{2\sigma^2}}$$

Probabilistic Interpretation (cont'd)

Log-likelihood of the training data \mathcal{D} (assuming i.i.d):

$$\begin{aligned}\log P(\mathcal{D}) &= \log \prod_{n=1}^N p(y_n|x_n) = \sum_n \log p(y_n|x_n) \\ &= \sum_n \left\{ -\frac{[y_n - (w_0 + w_1 x_n)]^2}{2\sigma^2} - \log \sqrt{2\pi}\sigma \right\} \\ &= -\frac{1}{2\sigma^2} \sum_n [y_n - (w_0 + w_1 x_n)]^2 - \frac{N}{2} \log \sigma^2 - N \log \sqrt{2\pi} \\ &= -\frac{1}{2} \left\{ \frac{1}{\sigma^2} \sum_n [y_n - (w_0 + w_1 x_n)]^2 + N \log \sigma^2 \right\} + \text{const}\end{aligned}$$

What is the relationship between minimizing RSS and maximizing the log-likelihood?

Maximum Likelihood Estimation

$$\log P(\mathcal{D}) = -\frac{1}{2} \left\{ \frac{1}{\sigma^2} \sum_n [y_n - (w_0 + w_1 x_n)]^2 + N \log \sigma^2 \right\} + \text{const}$$

Estimating σ , w_0 and w_1 can be done in two steps

- Maximize over w_0 and w_1 :

$$\max \log P(\mathcal{D}) \Leftrightarrow \min \sum_n [y_n - (w_0 + w_1 x_n)]^2 \leftarrow \text{This is RSS}(\tilde{\mathbf{w}})!$$

- Maximize over $s = \sigma^2$:

$$\begin{aligned} \frac{\partial \log P(\mathcal{D})}{\partial s} &= -\frac{1}{2} \left\{ -\frac{1}{s^2} \sum_n [y_n - (w_0 + w_1 x_n)]^2 + N \frac{1}{s} \right\} = 0 \\ \rightarrow \sigma^{*2} = s^* &= \frac{1}{N} \sum_n [y_n - (w_0 + w_1 x_n)]^2 \end{aligned}$$

Why Is This Interpretation Useful?

- It gives a solid footing to our intuition: minimizing $\text{RSS}(\tilde{\mathbf{w}})$ is a sensible thing based on reasonable modeling assumptions.
- Estimating σ^* tells us how much noise there is in our predictions. For example, it allows us to place confidence intervals around our predictions.

You Should Know

- Linear regression is the linear combination of features
 $f : \mathbf{x} \rightarrow y$, with $f(\mathbf{x}) = w_0 + \sum_d w_d x_d = w_0 + \mathbf{w}^\top \mathbf{x}$
- If we minimize residual sum of squares as our learning objective, we get a closed-form solution of parameters
- Probabilistic interpretation: maximum likelihood if assuming residual is Gaussian distributed