

18-661 Introduction to Machine Learning

Linear Regression – I

Fall 2020

ECE – Carnegie Mellon University

Announcements

- The math quiz scores posted on Gradescope – Entry Code: 9E882Y
- HW1 due on this Friday, Sept 11

1. Recap of MLE/MAP

2. Linear Regression

Motivation

Algorithm

Univariate solution

Multivariate Solution

Probabilistic interpretation

Computational and numerical optimization

Recap of MLE/MAP

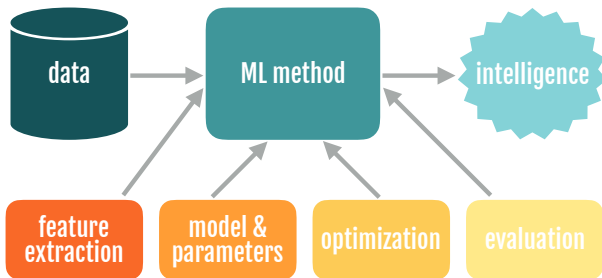
Dogecoin

- Scenario: You find a coin on the ground.
- *You ask yourself: 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 from this data?

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

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

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

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

- 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

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

- Suppose $\theta^* := 0.5$ and we observe: $D = \{H, H, T, T, T, T\}$
- Scenario 1: We assume $\theta \sim \text{Beta}(4, 4)$. Which is more accurate – θ_{MLE} or θ_{MAP} ?
 - $\theta_{MAP} = 5/12$, $\theta_{MLE} = 1/3$
- Scenario 2: We assume $\theta \sim \text{Beta}(1, 7)$. Which is more accurate – θ_{MLE} or θ_{MAP} ?
 - $\theta_{MAP} = 1/6$, $\theta_{MLE} = 1/3$

Linear Regression

Recap of MLE/MAP

Linear Regression

- Motivation

- Algorithm

- Univariate solution

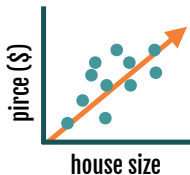
- Multivariate Solution

- Probabilistic interpretation

- Computational and numerical optimization

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

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.

Features used to predict



3620 South BUDLONG
Los Angeles, CA 90007
Status: Closed

\$1,510,000
Last Sold Price


14
Beds

6
Baths

4,418 Sq. Ft.
6342 / 86, Ft.

Built: 1956 Lot Size: 5,549 Sq. Ft. Sold On: JUL 26, 2013

[Overview](#)
[Property Details](#)
[Tour Insights](#)
[Property History](#)
[Public Records](#)
[Activity](#)
[Schools](#)



1 of 12

Five unit apartment complex within 2 blocks of USC campus, Gate #6. Great for students (most student leases have parents as guarantors). Most USC students live off campus, so housing units like this are always fully leased. Situated on a gated, corner lot, and across from an elementary school, this complex was recently renovated, and has in-unit laundry hook ups, wall-unit AC, and 12 parking spaces. It's within a DPS (Department of Public Safety) and Campus Cruiser patrolled area. This is a great income generating property, not to be missed!

Property Type: Multi-Family
Community: Downtown Los Angeles
MLS#: 22176741

Style: Two Level, Low Rise
County: [Los Angeles](#)

Property Details for 3620 South BUDLONG, Los Angeles, CA 90007

Details provided by iTech MLS and may not match the public record. [Learn More](#)

Interior Features

Kitchen Information

- Remodeled
- Oven, Range

Laundry Information

- Inside Laundry

Heating & Cooling

- Wall Cooling Unit(s)

Multi-Unit Information

Community Features

- Units in Complex (Total): 5

Multi-Family Information

- # Leased: 5
- # of Buildings: 1
- Owner Pays Water
- Tenant Pays Electricity, Tenant Pays Gas

Unit 1 Information

- # of Beds: 2
- # of Baths: 1
- Unfurnished
- Monthly Rent: \$1,700

Unit 2 Information

- # of Beds: 3
- # of Baths: 1
- Unfurnished
- Monthly Rent: \$2,250

Unit 3 Information

- Unfurnished

Unit 4 Information

- # of Beds: 3
- # of Baths: 1
- Unfurnished

Unit 5 Information

- Monthly Rent: \$2,350
- # of Beds: 3
- # of Baths: 2
- Unfurnished
- Monthly Rent: \$2,325
- # of Beds: 3
- # of Baths: 1
- Monthly Rent: \$2,250

Property / Lot Details

Property Features

- Automatic Gate, Card/Code Access

- Automatic Gate, Lawn, Sidewalks
- Corner Lot, Near Public Transit

- Tax Parcel Number: 5042017019

Lot Information

- Lot Size (Sq. Ft.): 5,549
- Lot Size (Acres): 0.2215
- Lot Size Source: Public Records

Property Information

- Updated/Remodeled
- Square Footage Source: Public Records

Parking / Garage, Exterior Features, Utilities & Financing

Parking Information

- # of Parking Spaces (Total): 12
- Parking Space
- Garage

Utility Information

- Green Certification Rating: 0.00
- Green Location: Transportation, Walkability
- Green Walk Score: 0
- Green Year Certified: 0

Financial Information

- Capitalization Rate (%): 6.25
- Actual Annual Gross Rent: \$128,331
- Gross Rent Multiplier: 11.29

Building Information

- Total Floors: 2

Location Details, Misc. Information & Listing Information

Location Information

- Gross Streets: W 26th Pl

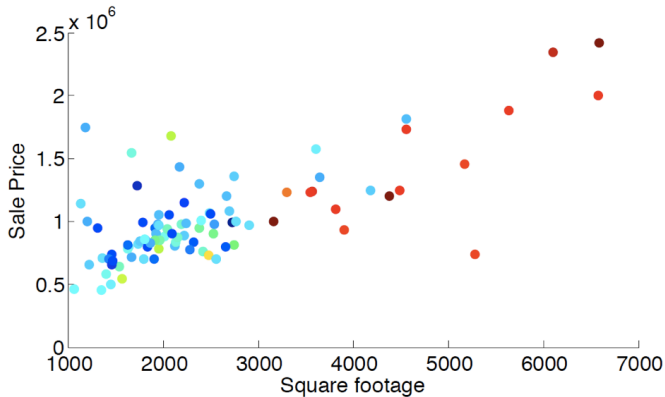
Expense Information

- Operating: \$37,664

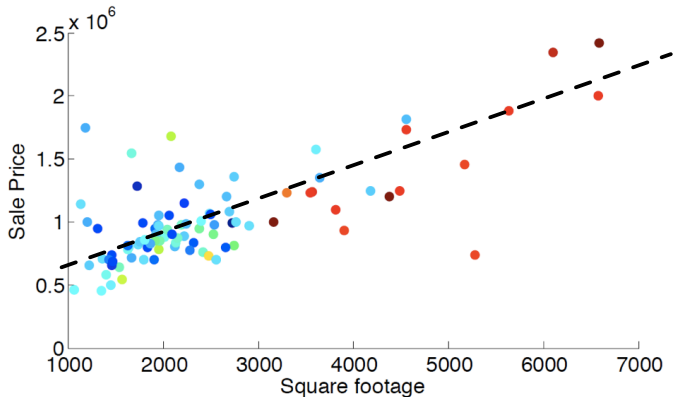
Listing Information

- Listing Terms: Cash, Cash To Existing Loan
- Buyer Financing: Cash

Correlation between square footage and sale price

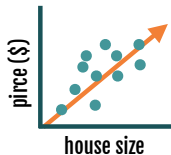


Roughly linear relationship



$$\text{Sale price} \approx \text{price_per_sqft} \times \text{square_footage} + \text{fixed_expense}$$

Data Can be Compactly Represented by Matrices



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

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

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

- Can represent compactly in matrix notation

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

Some Concepts That You Should Know

- Invertibility of Matrices and Computing Inverses
- Vector Norms – L2, Frobenius etc., Inner Products
- Eigenvalues and Eigen-vectors
- Singular Value Decomposition
- Covariance Matrices and Positive Semi-definite-ness

Excellent Resources:

- Essence of Linear Algebra YouTube Series
- Prof. Gilbert Strang's course at MIT
- More resources posted on Piazza

- Let us solve the house-price prediction problem

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

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

Matrix Inverse

- Let us solve the house-price prediction problem

$$\begin{bmatrix} 1000 & 1 \\ 2000 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_0 \end{bmatrix} = \begin{bmatrix} 200,000 \\ 350,000 \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} w_1 \\ w_0 \end{bmatrix} = \left(\begin{bmatrix} 1000 & 1 \\ 2000 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 200,000 \\ 350,000 \end{bmatrix} \quad (4)$$

$$= \frac{1}{-1000} \begin{bmatrix} 1 & -1 \\ -2000 & 1000 \end{bmatrix} \begin{bmatrix} 200,000 \\ 350,000 \end{bmatrix} \quad (5)$$

$$= \frac{1}{-1000} \begin{bmatrix} 150,000 \\ -5 \times 10^7 \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} w_1 \\ w_0 \end{bmatrix} = \begin{bmatrix} 150 \\ 50,000 \end{bmatrix} \quad (7)$$

You could have data from many houses

- Sale_price =
price_per_sqft \times square_footage + fixed_expense + unexplainable_stuff
- Want to learn the price_per_sqft and fixed_expense
- Training data: past sales record.

sqft	sale price
2000	800K
2100	907K
1100	312K
5500	2,600K
...	...

Problem: there isn't a $\mathbf{w} = [w_1, w_0]^T$ that will satisfy all equations

Want to predict the best price_per_sqft and fixed_expense

- Sale_price =
price_per_sqft \times square_footage + fixed_expense + unexplainable_stuff
- Want to learn the price_per_sqft and fixed_expense
- **Training data:** past sales record.

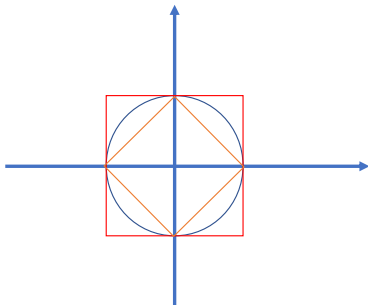
sqft	sale price	prediction
2000	810K	720K
2100	907K	800K
1100	312K	350K
5500	2,600K	2,600K
...

How to measure errors?

Need a way to map the vector of error (the difference between the prediction and sale price) 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 $a \in \mathbb{R}$
 - triangle inequality: $f(x + y) \leq f(x) + f(y)$
- e.g., ℓ_2 norm: $\|x\|_2 = \sqrt{x^\top x} = \sqrt{\sum_{i=1}^n x_i^2}$
- e.g., ℓ_1 norm: $\|x\|_1 = \sum_{i=1}^n |x_i|$
- e.g., ℓ_∞ norm: $\|x\|_\infty = \max |x_i|$



from inside to outside: ℓ_1 , ℓ_2 , ℓ_∞ norm ball.

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?

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

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

Reduce prediction error

How to measure 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: $|\text{prediction} - \text{sale price}|$.
- **squared** difference: $(\text{prediction} - \text{sale price})^2$ [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.

Recap of MLE/MAP

Linear Regression

Motivation

Algorithm

Univariate solution

Multivariate Solution

Probabilistic interpretation

Computational and numerical optimization

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 Regression

Motivation

Algorithm

Univariate solution

Multivariate Solution

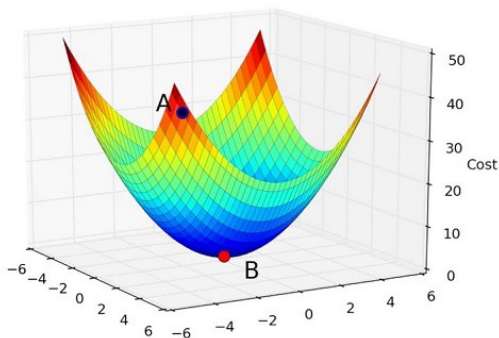
Probabilistic interpretation

Computational and numerical optimization

A simple case: \mathbf{x} is just 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 just one-dimensional ($D=1$)

Residual sum of squares:

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

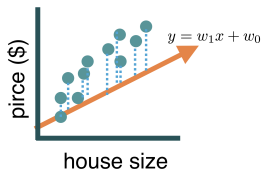


Figure 2: RSS is the sum of squares of the dotted lines

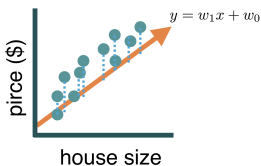


Figure 3: Adjust (w_0, w_1) to reduce RSS

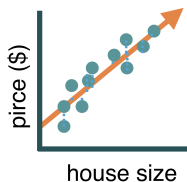


Figure 4: RSS minimized at (w_0^*, w_1^*)

A simple case: \mathbf{x} is just 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 just 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”:

$$\begin{aligned}\sum y_n &= Nw_0 + w_1 \sum x_n \\ \sum x_n y_n &= w_0 \sum x_n + w_1 \sum x_n^2\end{aligned}$$

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 $\bar{y} = \frac{1}{N} \sum_n y_n$.

Example

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

Residual sum of squares:

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

The w_1 and w_0 that minimize this are given by:

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

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

Example

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

Residual sum of squares:

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

The w_1 and w_0 that minimize this are given by:

$$w_1 \approx 1.6$$

$$w_0 \approx 0.45$$

Recap of MLE/MAP

Linear Regression

Motivation

Algorithm

Univariate solution

Multivariate Solution

Probabilistic interpretation

Computational and numerical optimization

Least Mean Squares when \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)

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

Least Mean Squares when \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:

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

. 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 Regression

Motivation

Algorithm

Univariate solution

Multivariate Solution

Probabilistic interpretation

Computational and numerical optimization

Why is minimizing RSS sensible?

Probabilistic interpretation

- Noisy observation model:

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

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

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 $s = \sigma^2$:

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

How does this probabilistic interpretation help us?

- 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.

Recap of MLE/MAP

Linear Regression

Motivation

Algorithm

Univariate solution

Multivariate Solution

Probabilistic interpretation

Computational and numerical optimization

Computational complexity of the Least Squares Solution

Bottleneck of computing the solution?

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

Matrix multiply of $\mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{(D+1) \times (D+1)}$

Inverting the matrix $\mathbf{X}^\top \mathbf{X}$

How many operations do we need?

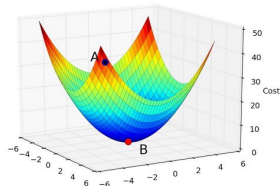
- $O(ND^2)$ for matrix multiplication $\mathbf{X}^\top \mathbf{X}$
- $O(D^3)$ (e.g., using Gauss-Jordan elimination) or $O(D^{2.373})$ (recent theoretical advances) for matrix inversion of $\mathbf{X}^\top \mathbf{X}$
- $O(ND)$ for matrix multiplication $\mathbf{X}^\top \mathbf{y}$
- $O(D^2)$ for $\left(\mathbf{X}^\top \mathbf{X} \right)^{-1}$ times $\mathbf{X}^\top \mathbf{y}$

$O(ND^2) + O(D^3)$ – Impractical for very large D or N

Alternative method: Batch Gradient Descent

(Batch) Gradient descent

- Initialize \mathbf{w} to $\mathbf{w}^{(0)}$ (e.g., randomly);
set $t = 0$; choose $\eta > 0$
- Loop *until convergence*
 1. Compute the gradient
$$\nabla \text{RSS}(\mathbf{w}) = \mathbf{X}^\top (\mathbf{X}\mathbf{w}^{(t)} - \mathbf{y})$$
 2. Update the parameters
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla \text{RSS}(\mathbf{w})$$
 3. $t \leftarrow t + 1$



What is the complexity of each iteration?

$O(\text{ND})$

Why would this work?

If gradient descent converges, it will converge to the same solution as using matrix inversion.

This is because $RSS(\mathbf{w})$ is a convex function in its parameters \mathbf{w}

Hessian of RSS

$$\begin{aligned} RSS(\mathbf{w}) &= \mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w} - 2 (\mathbf{X}^\top \mathbf{y})^\top \mathbf{w} + \text{const} \\ \Rightarrow \frac{\partial^2 RSS(\mathbf{w})}{\partial \mathbf{w} \mathbf{w}^\top} &= 2 \mathbf{X}^\top \mathbf{X} \end{aligned}$$

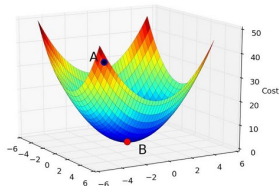
$\mathbf{X}^\top \mathbf{X}$ is positive semidefinite, because for any \mathbf{v}

$$\mathbf{v}^\top \mathbf{X}^\top \mathbf{X} \mathbf{v} = \|\mathbf{X}^\top \mathbf{v}\|_2^2 \geq 0$$

Alternative method: Batch Gradient Descent

(Batch) Gradient descent

- Initialize \mathbf{w} to $\mathbf{w}^{(0)}$ (e.g., randomly);
set $t = 0$; choose $\eta > 0$
- Loop *until convergence*
 1. Compute the gradient
$$\nabla \text{RSS}(\mathbf{w}) = \mathbf{X}^\top \mathbf{X} \mathbf{w}^{(t)} - \mathbf{X}^\top \mathbf{y}$$
 2. Update the parameters
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla \text{RSS}(\mathbf{w})$$
 3. $t \leftarrow t + 1$



What is the complexity of each iteration?

$O(ND)$

Stochastic gradient descent (SGD)

Widrow-Hoff rule: update parameters using one example at a time

- Initialize \mathbf{w} to some $\mathbf{w}^{(0)}$; set $t = 0$; choose $\eta > 0$
- Loop *until convergence*
 1. random choose a training a sample \mathbf{x}_t
 2. Compute its contribution to the gradient

$$\mathbf{g}_t = (\mathbf{x}_t^\top \mathbf{w}^{(t)} - y_t) \mathbf{x}_t$$

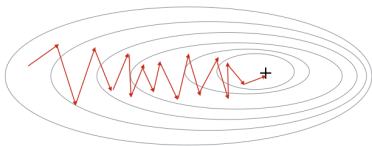
3. Update the parameters
$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \mathbf{g}_t$$
4. $t \leftarrow t + 1$

How does the complexity per iteration compare with gradient descent?

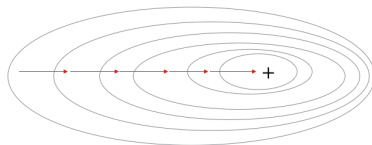
- $O(ND)$ for gradient descent versus $O(D)$ for SGD

SGD versus Batch GD

Stochastic Gradient Descent



Gradient Descent



- SGD reduces per-iteration complexity from $O(ND)$ to $O(D)$
- But it is noisier and can take longer to converge

Mini-Summary

- 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
- Gradient Descent and mini-batch SGD can overcome computational issues