25: Linear Regression and Gradient Ascent

Jerry Cain

May 23, 2022

Table of Contents

2 Linear Regression: Intro
6 Linear Regression: MSE
11 Linear Regression: MLE
18 Gradient Ascent
46 Extra: Derivations

Linear Regression

Today’s goals

We are going to learn linear regression.

- Informally known as "fitting data to a straight line"
- Linear models, however, are too simple for more complex datasets.
- Furthermore, many tasks in CS deal with classification (categorical data), not regression.

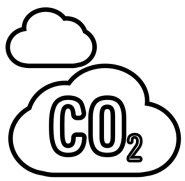
We cover this topic anyway so we can learn ***important techniques*** that will help us design and understand more complicated ML algorithms:

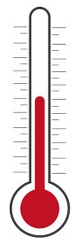
1. How to model likelihood of training data $(\boldsymbol{x}^{(i)}, y^{(i)})$
2. What rules of argmax and calculus are important to remember
3. What **gradient ascent** is and why it is useful

Regression: Predicting real numbers

Review

Training data: $(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})$





Global Land-Ocean temperature

	CO2 levels	Output
Year 1	338.8	0.26
Year 2	340.0	0.32
…		⋮
Year n	340.76	0.14

$\boldsymbol{X} = (X_1)$
(assume one feature)

$Y \in \mathbb{R}$

Model:
$\hat{Y} = g(\boldsymbol{X})$,
for some parametric function g

Linear Regression

Assume linear model (and $\boldsymbol{X}$ is 1-D):

$$\hat{Y} = g(\boldsymbol{X}) = aX + b$$

Training

Training data: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$

Learn parameters $\theta = (a, b)$

Two approaches:

- Analytical solution via mean squared error
- Iterative solution via MLE and gradient ascent

Linear Regression: MSE

Mean Squared Error (MSE)

For regression tasks, we usually want a $g(X)$ that minimizes MSE:

$$\theta_{MSE} = \arg\min_{\theta} E\left[\left(Y - \hat{Y}\right)^2\right] = \arg\min_{\theta} E\left[\left(Y - g(X)\right)^2\right]$$

- Y and $\hat{Y} = g(X)$ are both random variables
- Intuitively: Choose parameter θ that minimizes the expected squared deviation ("error") of your prediction $\hat{Y}$ from the true Y

For linear regression, where $\theta = (a, b)$ and $\hat{Y} = aX + b$:

$$E[(Y - aX - b)^2]$$

Don’t make me get non-linear!

$$\theta_{MSE} = \underset{\theta=(a,b)}{\arg\min}\, E[(Y - aX - b)^2]$$

$$a_{MSE} = \rho(X,Y)\,\frac{\sigma_Y}{\sigma_X}, \qquad b_{MSE} = \mu_Y - a_{MSE}\,\mu_X$$

(Derivation included at the end of slides)

Can we find these statistics on X and Y from our training data?

Training data: $\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \ldots, \left(x^{(n)}, y^{(n)}\right)$

Not exactly, but *we can estimate* them!

Don't make me get non-linear!

$$\theta_{MSE} = \underset{\theta=(a,b)}{\arg\min}\, E[(Y - aX - b)^2]$$

$$a_{MSE} = \rho(X,Y)\,\frac{\sigma_Y}{\sigma_X}, \qquad b_{MSE} = \mu_Y - a_{MSE}\,\mu_X$$

(Derivation included at the end of slides)

Can we find these statistics on X and Y from our training data?

Training data: $\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \ldots, \left(x^{(n)}, y^{(n)}\right)$

Estimate parameters based on observed training data:

$$\hat{a}_{MSE} = \hat{\rho}(X,Y)\,\frac{S_Y}{S_X}, \qquad \hat{b}_{MSE} = \bar{Y} - \hat{a}_{MSE}\bar{X}$$

$\hat{\rho}(X,Y)$: Sample correlation (Wikipedia)

Linear Regression

Review

Assume linear model (and $\boldsymbol{X}$ is 1-D):

$$\hat{Y} = g(\boldsymbol{X}) = aX + b$$

Training

Training data: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)})$, …, $(x^{(n)}, y^{(n)})$

Learn parameters $\theta = (a, b)$

If we want to minimize the mean squared error of our prediction,

$$\hat{a}_{MSE} = \hat{\rho}(X, Y)\frac{S_Y}{S_X}, \qquad \hat{b}_{MSE} = \bar{Y} - \hat{a}_{MSE}\bar{X}$$

Linear Regression: MLE

Linear Regression

Review

Assume linear model (and $\boldsymbol{X}$ is 1-D):

$$\hat{Y} = g(\boldsymbol{X}) = aX + b$$

Training

Learn parameters $\theta = (a, b)$

Training data: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$

We've seen which parameters minimize mean squared error.

What if we want parameters that maximize the likelihood of the training data?

Note: Maximizing likelihood is typically an objective for classification models.

Likelihood, it's been a minute

Review

Consider a sample of n iid random variables $X_1, X_2, \ldots, X_n$.

- X_i was drawn from a distribution with density function $f(X_i|\theta)$.
  or mass
- Observed data: $(X_1, X_2, \ldots, X_n)$

Likelihood question:

How likely is the observed data $(X_1, X_2, \ldots, X_n)$ given parameter θ?

Likelihood function, $L(\theta)$:

$$L(\theta) = f(X_1, X_2, \ldots, X_n|\theta) = \prod_{i=1}^{n} f(X_i|\theta)$$

This is just a product, since X_i are i.i.d.

Likelihood of the training data

Training data (n datapoints): (shorthand)

- $\left(x^{(i)}, y^{(i)}\right)$ drawn iid from a distribution $f\left(X = x^{(i)}, Y = y^{(i)} | \theta\right) = f\left(x^{(i)}, y^{(i)} | \theta\right)$
- $\hat{Y} = g(X)$, where $g(\cdot)$ is a function with parameter θ

We can show that θ_{MLE} maximizes the log conditional likelihood function:

$$\theta_{MLE} = \arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} | \, x^{(i)}, \theta\right)$$

Linear Regression, MLE

1. Assume linear model (and $\boldsymbol{X}$ is 1-D):

$$\hat{Y} = g(\boldsymbol{X}) = aX + b$$

2. Define maximum likelihood estimator:

$$\theta_{MLE} = \arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

⚠️ Issue: We have a model for $\hat{Y}$, not Y

- Remember the MSE approach, where we minimize the squared **error** between $\hat{Y}$ and Y?
- Now, we **model this error** directly!

$$\begin{aligned} Y &= \hat{Y} + Z \\ &= aX + b + Z \end{aligned}$$

error/noise
(also random)

Comparison: MSE vs MLE

$$\hat{Y} = g(\boldsymbol{X}) = aX + b$$

Minimum Mean Squared Error

$$\theta_{MSE} = \arg\min_{\theta} E\left[\left(Y - g(X)\right)^2\right]$$

- Do not directly model Y (nor error)
- Parameters are estimates of statistics from training data:

$$\hat{a}_{MSE} = \hat{\rho}(X, Y)\frac{S_Y}{S_X}$$
$$\hat{b}_{MSE} = \bar{Y} - \hat{a}_{MSE}\bar{X}$$

Maximum Likelihood Estimation

$$\theta_{MLE} = \arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

- Directly model error between predicted $\hat{Y}$ and Y

$$Y = \hat{Y} + Z = aX + b + Z$$

If we assume error $Z \sim \mathcal{N}(0, \sigma^2)$, then these two estimators are **equivalent.**

$$\theta_{MSE} = \theta_{MLE}!$$

Linear Regression, MLE (next steps)

1. Assume linear model (and $\boldsymbol{X}$ is 1-D):

$$\hat{Y} = g(\boldsymbol{X}) = aX + b$$

2. Define maximum likelihood estimator:

$$\theta_{MLE} = \arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

3. Model error, Z:

$Y = aX + b + Z$, where $Z \sim \mathcal{N}(0, \sigma^2)$

4. Pick $\theta = (a, b)$ that maximizes likelihood of training data

We will not analytically find a solution. Instead, we are going to use **gradient ascent**, an **iterative optimization algorithm.**

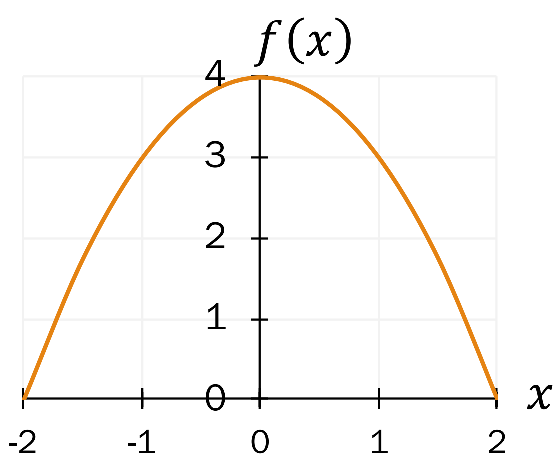
Gradient Ascent

Multiple ways to calculate argmax

Let $f(x) = -x^2 + 4$,
where $-2 < x < 2$.

What is $\underset{x}{\arg\max}\ \underbrace{f(x)}_{\text{objective function}}$?

A. Graph and guess

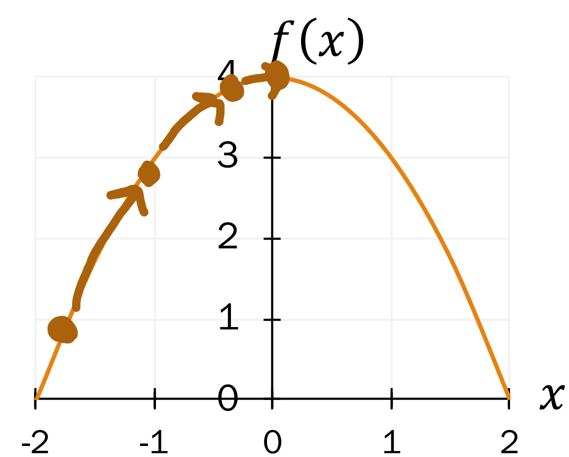


B. Differentiate, set to 0, and solve

$$\frac{df}{dx} = -2x = 0$$

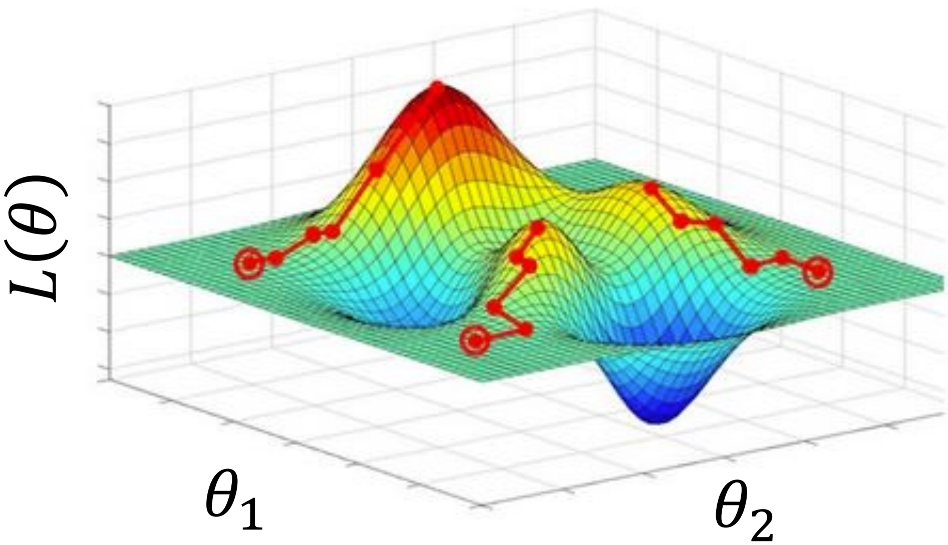
$$x = 0$$

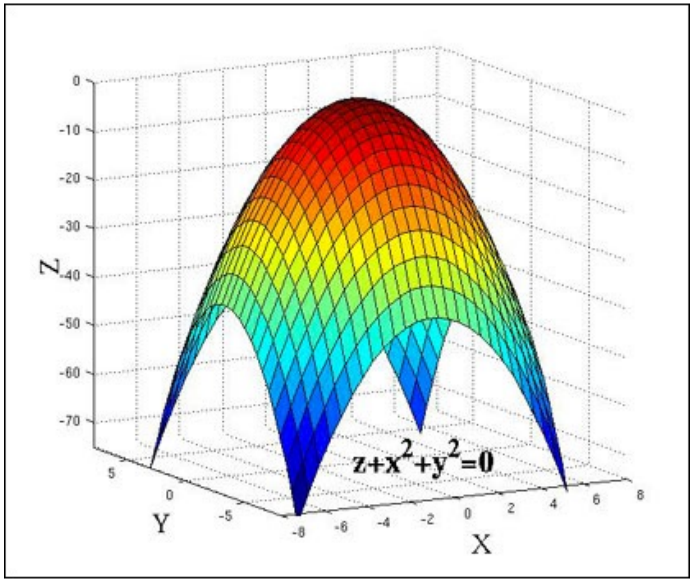
C. Gradient ascent: educated guess & check



Gradient ascent

Walk uphill and you will find a local maxima
(if your step is small enough).



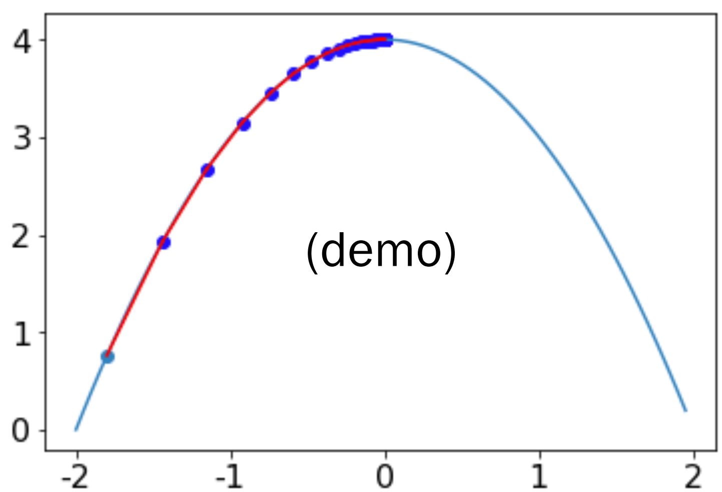


If your function is concave,
Local maxima = global maxima

Gradient ascent algorithm

Walk uphill and you will find a local maxima
(if your step is small enough).

Let $f(x) = -x^2 + 4$,
where $-2 < x < 2$.



1. $$\frac{df}{dx} = -2x$$ Gradient at x

2. Gradient ascent algorithm:

```
initialize x
repeat many times:
  compute gradient
  x += η * gradient
```

Computing the MLE

Review

General approach for finding $\theta_{MLE} = \arg\max_\theta LL(\theta)$:

1. Determine formula for $LL(\theta)$

$$LL(\theta) = \sum_{i=1}^{n} \log f(X_i|\theta)$$

2. Differentiate $LL(\theta)$ w.r.t. (each) θ

$$\frac{\partial LL(\theta)}{\partial \theta}$$

To maximize:

$$\frac{\partial LL(\theta)}{\partial \theta} = 0$$

3. Solve resulting (simultaneous) equations

(algebra or **computer**)

If algebra is intractable, we can still find a maximum using gradient ascent!

Linear Regression, MLE (so far)

Assume linear model (and $\boldsymbol{X}$ is 1-D):

$$\hat{Y} = g(\boldsymbol{X}) = aX + b$$

Model error, Z:

$$Y = aX + b + Z, \text{ where } Z \sim \mathcal{N}(0, \sigma^2)$$

Pick $\theta = (a, b)$ that maximizes likelihood of training data

$$\begin{aligned}
\theta_{MLE} &= \arg\max_{\theta} LL(\theta) \\
&= \arg\max_{\theta} \sum_{i=1}^{n} \log f\left(x^{(i)}, y^{(i)}, |\theta\right) \\
&= \arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} | \, x^{(i)}, \theta\right)
\end{aligned}$$

(θ_{MLE} also maximizes log conditional likelihood)

Computing the MLE with gradient ascent

General approach for finding θ_{MLE} , the MLE of θ:

1. Determine formula for $LL(\theta)$

   log conditional likelihood

$$\sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

2. Differentiate $LL(\theta)$ w.r.t. (each) θ

$$\frac{\partial}{\partial \theta_j} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

3. Solve resulting equations

   (computer) Gradient Ascent

1. Determine formula for log conditional likelihood

Model: $\theta = (a, b)$
$Y = aX + b + Z$
$Z \sim \mathcal{N}(0, \sigma^2)$

Optimization problem: $\arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$

Over the next few slides, we will show that our MLE linear regression θ_{MLE} reduces to

$$\arg\max_{\theta} \underbrace{\left[-\sum_{i=1}^{n}\left(y^{(i)} - ax^{(i)} - b\right)^2\right]}_{\text{objective function}}$$

1. Determine formula for log conditional likelihood

Model: $\theta = (a, b)$
$Y = aX + b + Z$
$Z \sim \mathcal{N}(0, \sigma^2)$

Optimization problem:

$$\arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

$$\xrightarrow{\text{goal}} \arg\max_{\theta} \left[-\sum_{i=1}^{n} \left(y^{(i)} - ax^{(i)} - b\right)^2 \right]$$

1. What is the conditional distribution, $Y|X,\theta$?

2. Substitute 1. into objective fn.

3. Use argmax properties to simplify objective fn.

1. Determine formula for log conditional likelihood

Model: $\theta = (a, b)$
$Y = aX + b + Z$
$Z \sim \mathcal{N}(0, \sigma^2)$

Optimization problem: $\arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$

1. What is the conditional distribution, $Y|X,\theta$?

$$Y|X,\theta \sim \mathcal{N}(aX + b, \sigma^2)$$

$$f\left(y^{(i)} \mid x^{(i)}, \theta\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(y^{(i)} - \left(ax^{(i)} + b\right)\right)^2 / (2\sigma^2)}$$

2. Substitute 1. into objective fn.

$$\arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right) = \arg\max_{\theta} \sum_{i=1}^{n} \log\left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\left(y^{(i)} - ax^{(i)} - b\right)^2 / (2\sigma^2)}\right]$$

using natural log

$$= \arg\max_{\theta} \left[\sum_{i=1}^{n} -\log\sqrt{2\pi}\sigma - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y^{(i)} - ax^{(i)} - b\right)^2\right]$$

1. Determine formula for log conditional likelihood

Model: $\theta = (a, b)$

$Y = aX + b + Z$

$Z \sim \mathcal{N}(0, \sigma^2)$

Optimization problem: $\arg\max_\theta \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$

3. Use argmax properties to simplify objective fn.

$$\arg\max_\theta \left[\sum_{i=1}^{n} -\log\sqrt{2\pi}\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y^{(i)} - ax^{(i)} - b\right)^2\right]$$

(from previous slide)

$$= \arg\max_\theta \left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y^{(i)} - ax^{(i)} - b\right)^2\right]$$

Argmax refresher #1:
Invariant to additive constants

$$= \arg\max_\theta \left[-\sum_{i=1}^{n}\left(y^{(i)} - ax^{(i)} - b\right)^2\right]$$

Argmax refresher #2:
Invariant to positive constant scalars

1. Determine formula for log conditional likelihood

Model: $\theta = (a, b)$
$Y = aX + b + Z$
$Z \sim \mathcal{N}(0, \sigma^2)$

Optimization problem: $\arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$

4. Celebrate!

$$\arg\max_{\theta} \left[-\sum_{i=1}^{n} \left(y^{(i)} - a x^{(i)} - b \right)^2 \right]$$

🎉🎉🎊🎊

Computing the MLE with gradient ascent

General approach for finding θ_{MLE} , the MLE of θ:

1. Determine formula for ~~$LL(\theta)$~~

   log conditional likelihood

$$\sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

$$h(\theta) = -\sum_{i=1}^{n} \left(y^{(i)} - ax^{(i)} - b\right)^2$$

2. Differentiate $LL(\theta)$ w.r.t. (each) θ

$$\frac{\partial}{\partial \theta_j} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

2-D gradient:

$$\left(\frac{\partial h(\theta)}{\partial a}, \frac{\partial h(\theta)}{\partial b}\right)$$

3. Solve resulting (simultaneous) equations

(computer) Gradient Ascent

2. Compute gradient

Model: $\theta = (a, b)$
$Y = aX + b + Z$
$Z \sim \mathcal{N}(0, \sigma^2)$

Optimization problem: $\arg\max_{\theta} \left[-\sum_{i=1}^{n} \left(y^{(i)} - ax^{(i)} - b \right)^2 \right]$

1. What is the derivative of the objective function w.r.t. a?

$$\frac{\partial}{\partial a} \left[-\sum_{i=1}^{n} \left(y^{(i)} - ax^{(i)} - b \right)^2 \right] =$$

Calculus refresher #1:
Derivative(sum) = sum(derivative)

Calculus refresher #2:
Chain rule

2. What is the derivative of the objective function w.r.t. b?

2. Compute gradient

Model: $\theta = (a, b)$
$Y = aX + b + Z$
$Z \sim \mathcal{N}(0, \sigma^2)$

Optimization problem: $\arg\max_{\theta} \left[-\sum_{i=1}^{n} \left(y^{(i)} - a x^{(i)} - b \right)^2 \right]$

1. What is the derivative of the objective function w.r.t. a?

$$\frac{\partial}{\partial a} \left[-\sum_{i=1}^{n} \left(y^{(i)} - a x^{(i)} - b \right)^2 \right] =$$

Calculus refresher #1:
Derivative(sum) = sum(derivative)

Calculus refresher #2:
Chain rule 🌟 🌟 🌟

2. Compute gradient

Model: $\theta = (a, b)$
$Y = aX + b + Z$
$Z \sim \mathcal{N}(0, \sigma^2)$

Optimization problem: $\arg\max_{\theta} \left[-\sum_{i=1}^{n} \left(y^{(i)} - ax^{(i)} - b \right)^2 \right]$

1. What is the derivative of the objective function w.r.t. a?

$$\sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)\left(x^{(i)}\right)$$

2. What is the derivative of the objective function w.r.t. b?

$$\sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)$$

analytical solution for a_{MLE}, b_{MLE}: Set to 0 and solve simultaneous equations

Next up: We will reach the same solution **computationally** with **gradient ascent.**

Computing the MLE with gradient ascent

General approach for finding θ_{MLE} , the MLE of θ:

1. Determine formula for $LL(\theta)$

   log conditional likelihood

   $$\sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

   $$h(\theta) = -\sum_{i=1}^{n}\left(y^{(i)} - ax^{(i)} - b\right)^2$$

2. Differentiate $LL(\theta)$ w.r.t. (each) θ

   $$\frac{\partial}{\partial \theta_j}\sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

   $$\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)\left(x^{(i)}\right)$$

   $$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)$$

3. Solve resulting (simultaneous) equations

   (computer)
   Gradient Ascent

3. Gradient ascent with multiple parameters (if time)

Optimization problem: $\arg\max_{\theta} \left[-\sum_{i=1}^{n} \left(y^{(i)} - ax^{(i)} - b \right)^2 \right]$

$$= \arg\max_{\theta} h(\theta)$$

Gradient:

$$\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)\left(x^{(i)}\right)$$

$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)$$

```
initialize θ
repeat many times:
  compute gradient
  θ += η * gradient
```

How does this work for multiple parameters?

3. Gradient ascent with multiple parameters

Optimization problem: $\arg\max_{\theta} \left[-\sum_{i=1}^{n} \left(y^{(i)} - ax^{(i)} - b \right)^2 \right]$

$= \arg\max_{\theta} h(\theta)$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)\left(x^{(i)}\right)$

$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)$

```
a, b = 0, 0                    # initialize θ
repeat many times:
   gradient_a, gradient_b = 0, 0
   # TODO: fill in



   a += η * gradient_a     # θ += η * gradient
   b += η * gradient_b
```

How do we pseudocode the gradients we derived?

3. Gradient ascent with multiple parameters

Optimization problem: $\arg\max_{\theta} \left[-\sum_{i=1}^{n} \left(y^{(i)} - ax^{(i)} - b \right)^2 \right]$

$= \arg\max_{\theta} h(\theta)$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)\left(x^{(i)}\right)$

$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)$

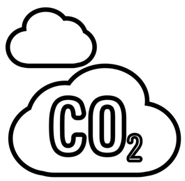
```
a, b = 0, 0                         # initialize θ
repeat many times:
  gradient_a, gradient_b = 0, 0
  for each training example (x, y):
    diff = y − (a ∗ x + b)
    gradient_a += 2 ∗ diff ∗ x
    gradient_b += 2 ∗ diff
  a += η ∗ gradient_a     # θ += η ∗ gradient
  b += η ∗ gradient_b
```

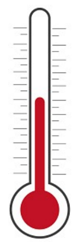
Finish computing gradient before updating any part of θ.

(Spring 2021 demo)

Global land-ocean temperature prediction

Training data: $(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)})$





	CO2 levels	Output
Year 1	338.8	0.26
Year 2	340.0	0.32
…		⋮
Year n	340.76	0.14

$\boldsymbol{X} = (X_1)$
(assume one feature)

$Y \in \mathbb{R}$

Review

Minimizing
Mean Square Error

$$\theta_{MSE} = \arg\min_{\theta} E\left[\left(Y - g(X)\right)^2\right]$$

$$\hat{Y} = \hat{\rho}(X, Y)\frac{S_Y}{S_X}(X - \bar{X}) + \bar{Y}$$

$$a_{MSE} = 0.01452$$
$$b_{MSE} = 0.17511$$

3b. Interpret

Optimization problem: $\arg\max_{\theta}\left[-\sum_{i=1}^{n}\left(y^{(i)}-ax^{(i)}-b\right)^{2}\right]$

$= \arg\max_{\theta} h(\theta)$

Gradient: $\frac{\partial h(\theta)}{\partial a}=\sum_{i=1}^{n}2\left(y^{(i)}-ax^{(i)}-b\right)\left(x^{(i)}\right)$

$\frac{\partial h(\theta)}{\partial b}=\sum_{i=1}^{n}2\left(y^{(i)}-ax^{(i)}-b\right)$

```
a, b = 0, 0                 # initialize θ
repeat many times:
  gradient_a, gradient_b = 0, 0
  for each training example (x, y):
    diff = y – (a * x + b)
    gradient_a += 2 * diff * x
    gradient_b += 2 * diff
  a += η * gradient_a       # θ += η * gradient
  b += η * gradient_b
```

Updates to a and b should include information from all n training datapoints

3b. Interpret

Optimization problem: $$\arg\max_{\theta}\left[-\sum_{i=1}^{n}\left(y^{(i)}-ax^{(i)}-b\right)^2\right]$$

$$= \arg\max_{\theta} h(\theta)$$

Gradient: $$\frac{\partial h(\theta)}{\partial a}=\sum_{i=1}^{n}2\left(y^{(i)}-ax^{(i)}-b\right)\left(x^{(i)}\right)$$

$$\frac{\partial h(\theta)}{\partial b}=\sum_{i=1}^{n}2\left(y^{(i)}-ax^{(i)}-b\right)$$

```
a, b = 0, 0                # initialize θ
repeat many times:
  gradient_a, gradient_b = 0, 0
  for each training example (x, y):
    diff = y – (a ∗ x + b)
    gradient_a += 2 ∗ diff ∗ x
    gradient_b += 2 ∗ diff
  a += η ∗ gradient_a      # θ += η ∗ gradient
  b += η ∗ gradient_b
```

How do we interpret the contribution of the i-th training datapoint?

3b. Interpret

Optimization problem: $\arg\max_{\theta} \left[-\sum_{i=1}^{n} \left(y^{(i)} - a x^{(i)} - b \right)^2 \right]$

$$= \arg\max_{\theta} h(\theta)$$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^{n} 2\left(y^{(i)} - a x^{(i)} - b\right)\left(x^{(i)}\right)$

$$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^{n} 2\left(y^{(i)} - a x^{(i)} - b\right)$$

```
a, b = 0, 0                    # initialize θ
repeat many times:
  gradient_a, gradient_b = 0, 0
  for each training example (x, y):
    diff = y − (a ∗ x + b)
    gradient_a += 2 ∗ diff ∗ x
    gradient_b += 2 ∗ diff
  a += η ∗ gradient_a     # θ += η ∗ gradient
  b += η ∗ gradient_b
```

Prediction error!

$y^{(i)} - \hat{y}^{(i)}$

3b. Interpret

Optimization problem: $\arg\max_{\theta}\left[-\sum_{i=1}^{n}\left(y^{(i)}-ax^{(i)}-b\right)^2\right]$

$= \arg\max_{\theta} h(\theta)$

Gradient: $\frac{\partial h(\theta)}{\partial a}=\sum_{i=1}^{n}2\left(y^{(i)}-ax^{(i)}-b\right)\left(x^{(i)}\right)$

$\frac{\partial h(\theta)}{\partial b}=\sum_{i=1}^{n}2\left(y^{(i)}-ax^{(i)}-b\right)$

```
a, b = 0, 0                       # initialize θ
repeat many times:
  gradient_a, gradient_b = 0, 0
  for each training example (x, y):
    prediction_error = y − (a ∗ x + b)
    gradient_a += 2 ∗ prediction_error ∗ x
    gradient_b += 2 ∗ prediction_error
  a += η ∗ gradient_a    # θ += η ∗ gradient
  b += η ∗ gradient_b
```

3b. Interpret

Optimization problem: $\arg\max_{\theta} \left[-\sum_{i=1}^{n} \left(y^{(i)} - a x^{(i)} - b \right)^2 \right]$

$= \arg\max_{\theta} h(\theta)$

Gradient: $\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^{n} 2\left(y^{(i)} - a x^{(i)} - b\right)\left(x^{(i)}\right)$

$\frac{\partial h(\theta)}{\partial b} = \sum_{i=1}^{n} 2\left(y^{(i)} - a x^{(i)} - b\right)$

```
a, b = 0, 0                  # initialize θ
repeat many times:
  gradient_a, gradient_b = 0, 0
  for each training example (x, y):
    prediction_error = y – (a * x + b)
    gradient_a += 2 * prediction_error * x
    gradient_b += 2 * prediction_error
  a += η * gradient_a     # θ += η * gradient
  b += η * gradient_b
```

$\hat{Y} = aX + b$, so update to a should also scale by $x^{(i)}$

3b. Interpret

Optimization problem: $\arg\max_{\theta} \left[-\sum_{i=1}^{n} \left(y^{(i)} - ax^{(i)} - b \right)^2 \right]$

$= \arg\max_{\theta} h(\theta)$

Gradient: $\dfrac{\partial h(\theta)}{\partial a} = \sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)\left(x^{(i)}\right)$

$\dfrac{\partial h(\theta)}{\partial b} = \sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b\right)$

```
a, b = 0, 0                    # initialize θ
repeat many times:
  gradient_a, gradient_b = 0, 0
  for each training example (x, y):
    prediction_error = y − (a ∗ x + b)
    gradient_a += 2 ∗ prediction_error ∗ x
    gradient_b += 2 ∗ prediction_error ∗ 1
  a += η ∗ gradient_a    # θ += η ∗ gradient
  b += η ∗ gradient_b
```

$\hat{Y} = aX + b$, so update to b just scales by 1, not $x^{(i)}$

Reflecting on today

We did a lot today!

- Learned gradient ascent
- Modeled likelihood of training dataset
- Thanked argmax for its convenience
- Remembered calculus
- Implemented gradient ascent with multiple parameters to optimize for

Next up, we will use all these skills and more to tackle the final prediction model of CS109:

Logistic Regression

Extra: Derivations

Don't make me get non-linear!

$$\theta_{MSE} = \arg\min_{\theta=(a,b)} E[(Y - aX - b)^2]$$

1. Differentiate w.r.t. (each) θ, set to 0

$$\frac{\partial}{\partial a} E[(Y - aX - b)^2] = E\left[\frac{\partial}{\partial a}(Y - aX - b)^2\right]$$

($E[\cdot]$ is a linear function w.r.t. a)

$$= E[-2(Y - aX - b)X]$$

$$= -2E[XY] + 2aE[X^2] + 2bE[X]$$

$$\frac{\partial}{\partial b} E[(Y - aX - b)^2] = E[-2(Y - aX - b)]$$

$$= -2E[Y] + 2aE[X] + 2b$$

2. Solve resulting simultaneous equations

$$a_{MSE} = \frac{E[XY] - E[X]E[Y]}{E[X^2] - (E[X])^2} = \frac{\text{Cov}(X,Y)}{\text{Var}(X)} = \rho(X,Y)\frac{\sigma_Y}{\sigma_X}$$

$$b_{MSE} = E[Y] - a_{MSE}E[X] = \mu_Y - \rho(X,Y)\frac{\sigma_Y}{\sigma_X}\mu_X$$

Log conditional likelihood, a derivation

$\hat{Y} = g(X)$, where $g(\cdot)$ is a function with parameter θ

Show that θ_{MLE} maximizes the **log conditional likelihood** function:

$$\theta_{MLE} = \arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

<u>Proof</u>:

$$\theta_{MLE} = \arg\max_{\theta} \prod_{i=1}^{n} f\left(x^{(i)}, y^{(i)} \mid \theta\right) \quad = \arg\max_{\theta} \sum_{i=1}^{n} \log f\left(x^{(i)}, y^{(i)} \mid \theta\right)$$

(θ_{MLE} also maximizes $LL(\theta)$)

$$= \arg\max_{\theta} \sum_{i=1}^{n} \log f\left(x^{(i)} \mid \theta\right) + \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

(chain rule,
log of product = sum of logs)

$$= \arg\max_{\theta} \sum_{i=1}^{n} \log f\left(x^{(i)}\right) + \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

($x^{(i)}$ indep. of θ)

$$= \arg\max_{\theta} \sum_{i=1}^{n} \log f\left(y^{(i)} \mid x^{(i)}, \theta\right)$$

($f\left(x^{(i)}\right)$ constant w.r.t. θ)