25: Linear Regression and Gradient Ascent

Lisa Yan and Jerry Cain

November 9, 2020

Quick slide reference

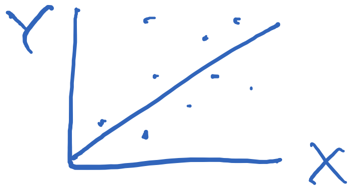
3	Linear Regression	25a_linreg
7	Linear Regression: MSE	25b_linreg_mse
12	Linear Regression: MLE	25c_linreg_mle
19	Gradient Ascent	25d_gradient_ascent
24	Linear Regression with Gradient Ascent	LIVE
*	Extra: Derivations	25f_extra_derivations

Linear Regression

Today's goals

We are going to learn linear regression.

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



The reason we cover this topic is to teach us <u>important skills</u> that will help us design and understand more complicated ML algorithms:

1. How to model likelihood of training data $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)$
2. What rules of argmax/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:

prediction $\hat{Y} = g(\boldsymbol{X})$,

for some parametric function g

Linear Regression

Assume linear model (and $\boldsymbol{X}$ is 1-D): $\boldsymbol{X} = \langle X_1 \rangle = X$

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

Training

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

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

Two approaches:

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

24b_linreg_mse

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 this lecture)

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!

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

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

(Derivation included at the end of this lecture)

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

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):

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

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. $\mathbb{E}[(Y-\hat{Y})^2]$

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 i.i.d. random variables $X_1, X_2, \dots, X_n$.

$X_i \sim Poi(\lambda)$

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

Likelihood question:

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

Likelihood function, $L(\theta)$:

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

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 i.i.d. 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:

$$\theta_{MLE} = \underset{\theta}{\arg\max} \prod_{i=1}^{n} f\left(x^{(i)}, y^{(i)} | \theta\right)$$

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

(This derivation is included at the end of this video)

(difficult)

Linear Regression, MLE

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

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

2. Define maximum likelihood estimator:

$$\theta_{MLE} = \arg\max_{\theta} \sum_{i=1}^{n} \underbrace{\log f\left(y^{(i)} \mid x^{(i)}, \theta\right)}_{\text{what is the conditional distribution of } Y \mid X, \theta\text{?}}$$

⚠️ Issue: We have a model of the <u>prediction</u> $\hat{Y}$ (and not Y)

- Remember MSE approach, where we minimize the squared error between $\hat{Y}$ and Y?

$$E[(Y - \hat{Y})^2]$$

- Now, we model this error directly!

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

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

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

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

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

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:

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

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.**

24d_gradient_ascent

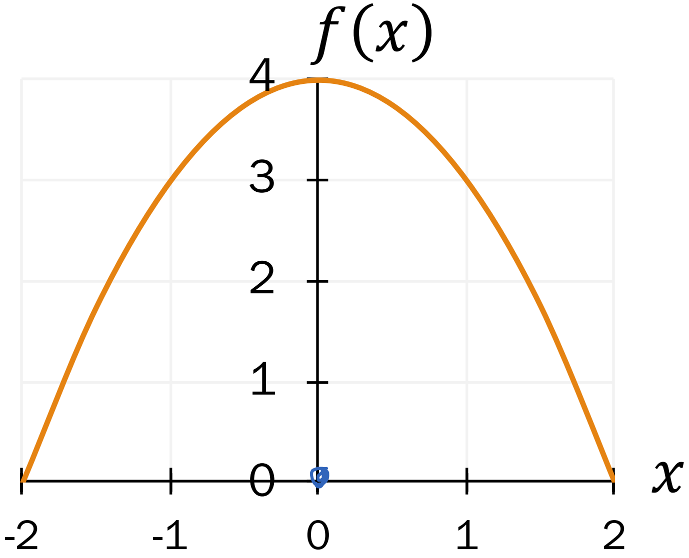
Gradient Ascent

Multiple ways to calculate argmax

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

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

A. Graph and guess

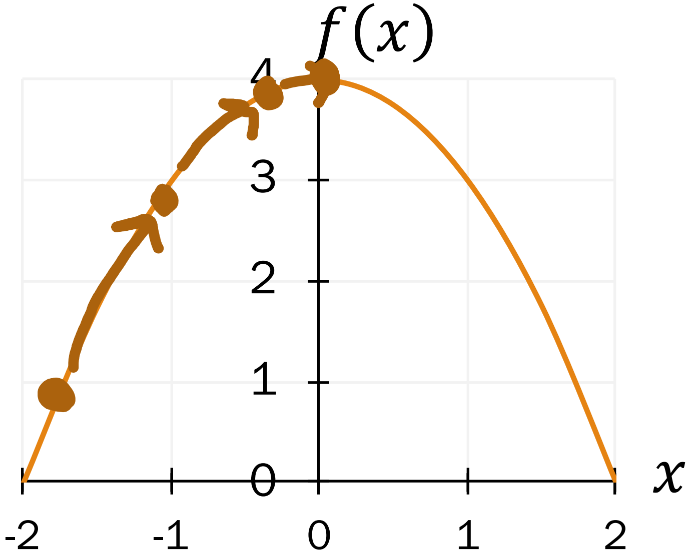


B. Differentiate, set to 0, and solve

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

$$x = 0$$

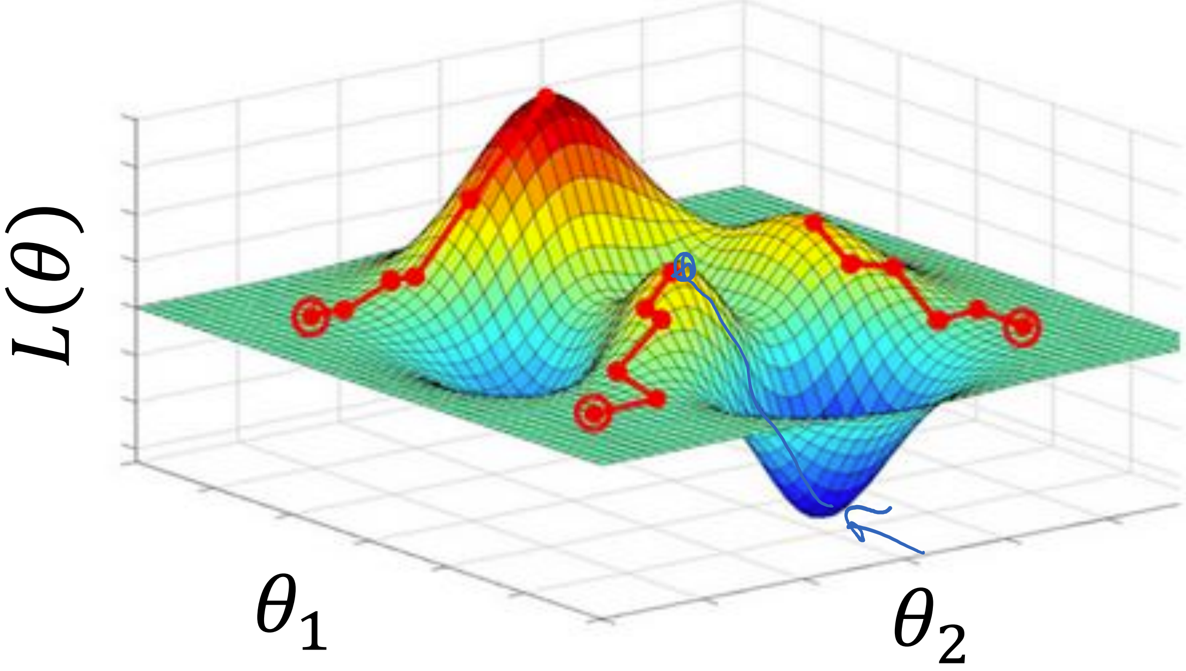
C. Gradient ascent: educated guess & check

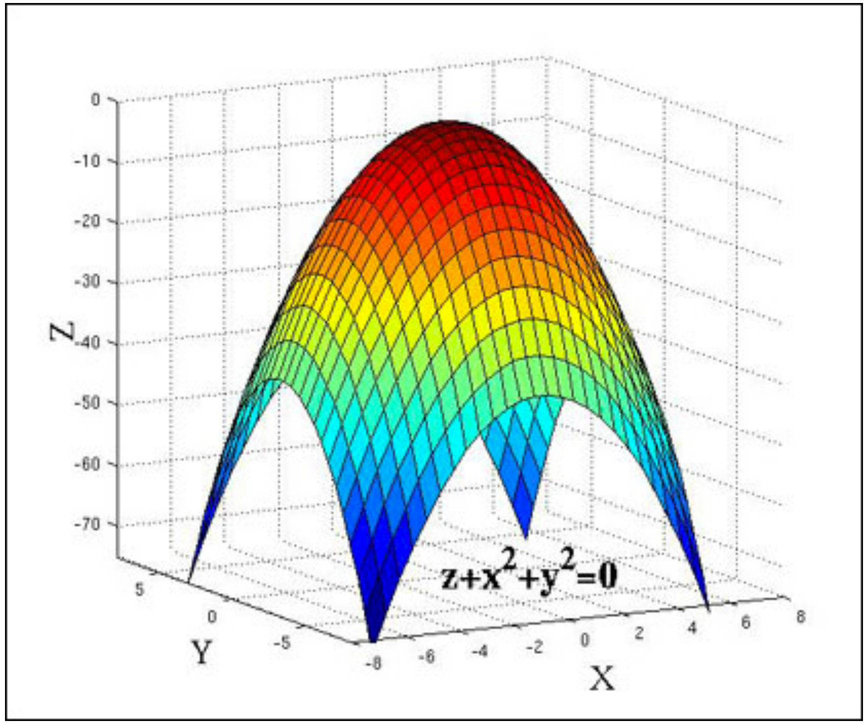


Gradient ascent

CS109
- $L(\theta)$ are concave
- $LL(\theta)$ are also concave

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



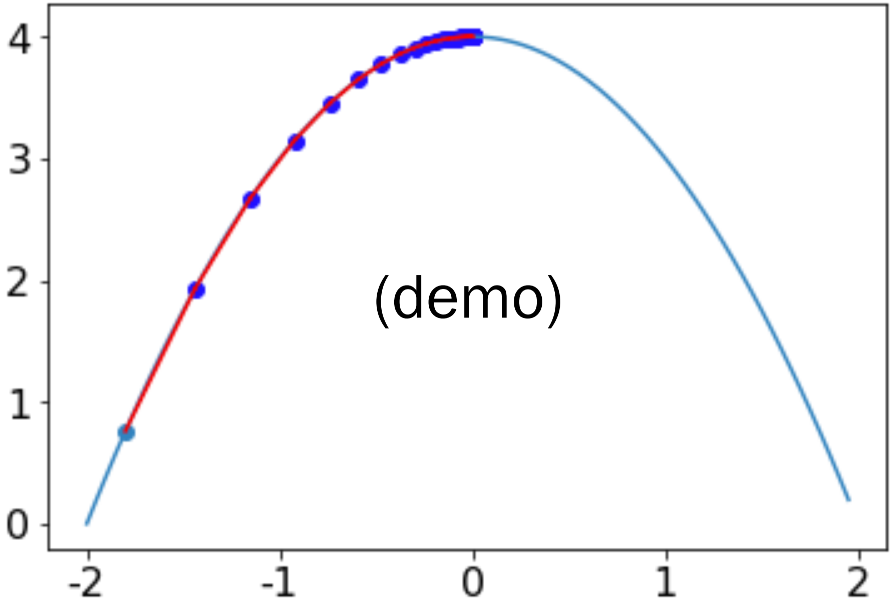


If your function is concave,
Local maxima = global maxima

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 += $\boxed{\eta}$ * gradient
```

learning rate

Computing the MLE

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!

(live)

25: Linear Regression and Gradient Ascent

Lisa Yan and Jerry Cain

November 9, 2020

Three goals today

$$\theta_{MLE} = \arg\max_{\theta} LL(\theta)$$

$$\log f(x^{(i)}, y^{(i)} | \theta)$$

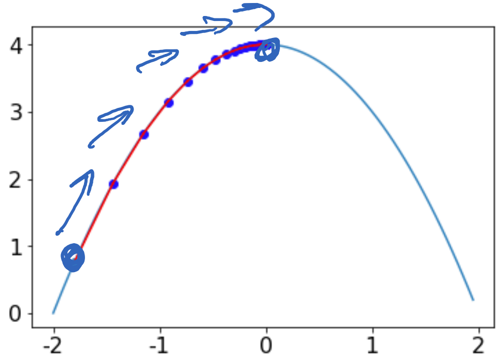
- ✅ How to model likelihood of training data $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)$

(θ_{MLE} also maximizes log conditional likelihood)

$$\log f(y^{(i)} | x^{(i)}, \theta)$$

- What gradient ascent is, why it is useful, and how to use it

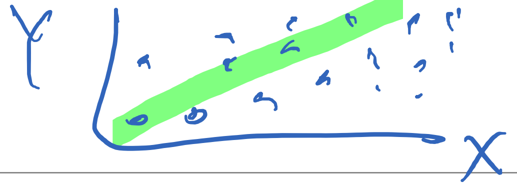
(an iterative optimization algorithm)



- Use properties of argmax/calculus

(to review)

Linear Regression, MLE (so far)



Review

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

$$\theta_{MLE} = \arg\max_{\theta} LL(\theta)$$

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

(θ_{MLE} also maximizes log conditional likelihood)

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

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

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

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 (~~simultaneous~~) equations

(computer)
Gradient Ascent

1. Determine formula for log conditional likelihood

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

$Y = aX + b + Z$

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

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

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

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

Breakout Rooms

Check out the questions on the next slide (Slide 30). Post any clarifications here!

https://us.edstem.org/courses/2678/discussion/171555

Breakout rooms: 3 min

🤔

1. Determine formula for log conditional likelihood

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

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

goal

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

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:

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

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

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

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

$$Y \mid X = x, \theta = (a,b)$$
$$Y = ax + b + Z$$

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

2. Substitute 1. into objective fn.

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

using natural log

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

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.

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

(from previous slide)

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

Argmax refresher #1:
Invariant to additive constants

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

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!

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

🎉🎉🎊🎊

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

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

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

$\theta = a, b$

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:

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

3. Solve resulting (simultaneous) equations

(computer)
Gradient Ascent

Think

Slide 36 has two questions to go over by yourself.

Post any clarifications here!

https://us.edstem.org/courses/2678/discussion/153773

Think by yourself: 2 min

(by yourself)

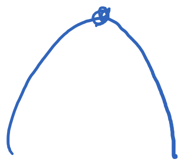
2. Compute gradient

$$\overset{a,b}{\theta_{MLE}} = \arg\max [h(\theta)]$$

$$\frac{\partial h(a_{MLE}, b_{MLE})}{\partial a} = 0$$

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

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



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

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

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

$$\frac{\partial h}{\partial b}$$

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

Calculus refresher #2:
Chain rule 🌟🌟🌟

$$\frac{d}{da} f(g(a)) = \frac{df(z)}{dz} \cdot \frac{dg(a)}{da}$$

🤔 (by yourself)

2. Compute gradient

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

$Y = aX + b + Z$

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

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

Calculus refresher #1:

Derivative(sum) = sum(derivative)

Calculus refresher #2:

Chain rule 🌟🌟🌟

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

#1

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

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

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

2. Compute gradient

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

Optimization problem:

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

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

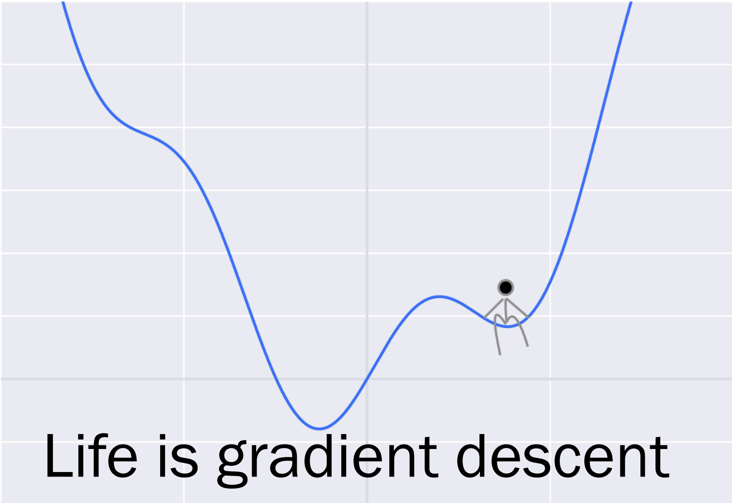
$$\sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b \right)\left(x^{(i)} \right) = 0$$

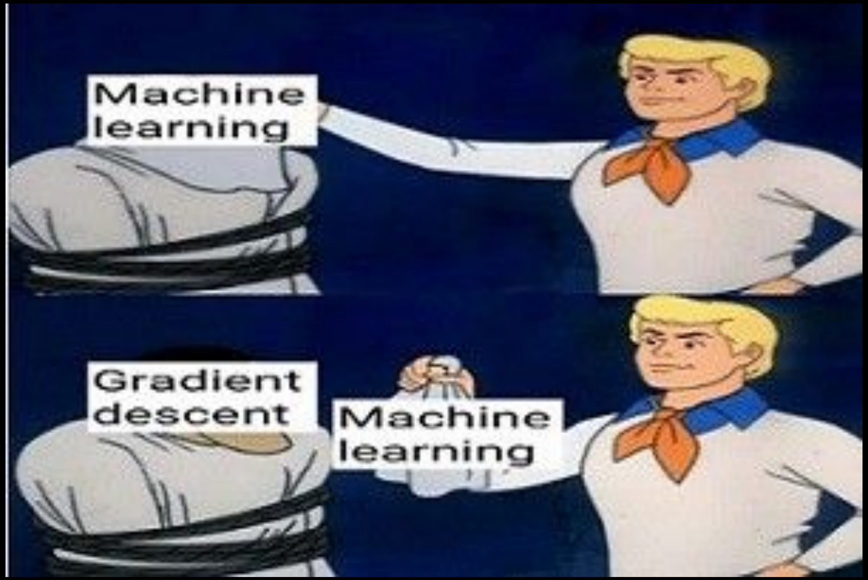
2. What is the derivative of the objective function w.r.t. b?

$$\sum_{i=1}^{n} 2\left(y^{(i)} - ax^{(i)} - b \right) = 0$$

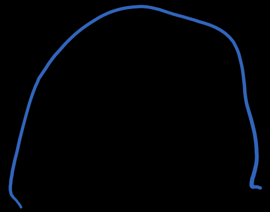
analytical solution for a_{MLE}, b_{MLE}: Set to 0 and solve simultaneous equations

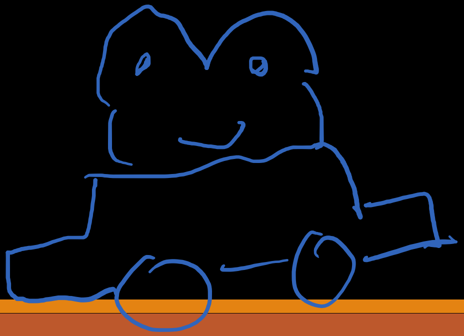
Next up: We will reach the same solution **computationally** with **gradient ascent.**



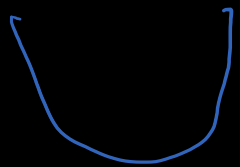


Interlude for jokes





toad
away



Note: gradient **de**scent finds local **min**imum

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

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 $\theta$ = a, b
repeat many times:
  compute gradient
  $\theta$ += $\eta$ * gradient
```

↑ learning rate

How does this work for multiple parameters?

3. Gradient ascent with multiple parameters

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

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

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
    # 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:

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

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

Gradient:

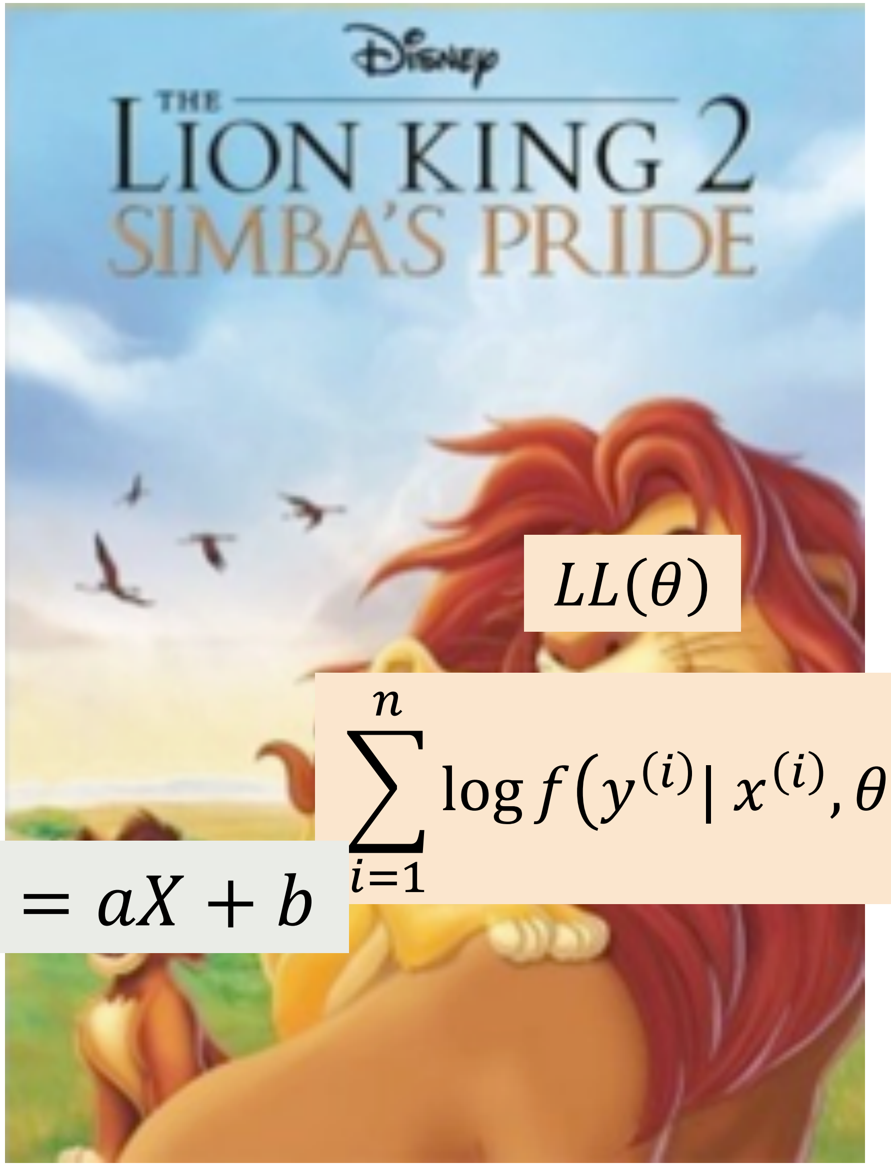
$$\frac{\partial h(\theta)}{\partial a} = \sum_{i=1}^{n} 2\left(y^{(i)} - a x^{(i)} - b \right) \underbrace{\left(x^{(i)} \right)}$$

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

```
a, b = 0, 0                      # initialize $\theta$
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            # $\theta$ += η * gradient
  b += η * gradient_b
```

Finish computing gradient before updating any part of θ.

Let’s try it out



$$LL(\theta)$$

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

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

(Fall 2020 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)

$Y \in \mathbb{R}$

Minimizing
Mean Square Error

Review

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

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

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

3b. Interpret

max likelihood of training data

$y^{(i)} = ax^{(i)} + b$

$(x^{(i)}, y^{(i)})$ $i = 1, \ldots, n$

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?

(by yourself)

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
```

$\hat{y}^{(i)} = ax^{(i)} + b$

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:

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

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

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
```

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

3b. Interpret

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

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

$\theta = (a, b)$

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 ∗ 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 $x^{(i)}$

$\eta = 1 \times 10^{-6}$

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

24f_extra_derivations

Extra: Derivations

$\hat{Y} = aX + b$

- $(a_{MSE}, b_{MSE}) = \theta_{MSE}$

  $= \underset{\theta}{\arg\min}\, \mathbb{E}[(Y - \hat{Y})^2]$

- $\theta_{MLE} = \underset{\theta}{\arg\max}\, L(\theta)$

Don't make me get non-linear!

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

$$\frac{d}{da}(f(a))^2 = 2f(a)\frac{df}{da}$$

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)

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

$$= -2E[XY] + 2aE[X^2] + 2bE[X] = 0$$

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

$$= -2E[Y] + 2aE[X] + 2b = 0$$

2. Solve resulting simultaneous equations

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

$$\frac{\text{Cov}(X,Y)}{\text{Var}(X)} = \frac{\rho(X,Y)\sigma_X\sigma_Y}{\sigma_X\sigma_X}$$

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

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:

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

<u>Proof</u>:

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

(θ_{MLE} also maximizes $LL(\theta)$)

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

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

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

($x^{(i)}$ indep. of θ)

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

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