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 \((x^{(i)}, y^{(i)})\)
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

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

<table>
<thead>
<tr>
<th>Year</th>
<th>CO2 levels</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>338.8</td>
<td>0.26</td>
</tr>
<tr>
<td>2</td>
<td>340.0</td>
<td>0.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>\vdots</td>
</tr>
<tr>
<td>(n)</td>
<td>340.76</td>
<td>0.14</td>
</tr>
</tbody>
</table>

\(X = (X_1)\)
(assume one feature)

Global Land-Ocean temperature

Model:
\[\hat{Y} = g(X),\]
for some parametric function \(g\)
Linear Regression

Assume linear model (and $X$ is 1-D):

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

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:

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

- $Y$ and $\hat{Y} = g(X)$ are both random variables
- Intuitively: Choose parameter $\theta$ 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!

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

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

(Derivation included at the end of slides)

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

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

Not exactly, but we can estimate them!
Don’t make me get non-linear!

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

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

(Derivation included at the end of slides)

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

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

**Estimate** parameters based on observed training data:

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

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

Assume linear model (and $X$ is 1-D):

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

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

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

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

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

Assume linear model (and $X$ is 1-D):

\[ \hat{Y} = g(X) = aX + b \]

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

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 \( f(X_i | \theta) \).
- 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 \( \theta \)?

**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):

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

We can show that $\theta_{MLE}$ maximizes the log conditional likelihood function:

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

1. Assume linear model (and \(X\) is 1-D):

\[ \hat{Y} = g(X) = aX + b \]

2. Define maximum likelihood estimator:

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

⚠ 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!

\[
Y = \hat{Y} + Z
\]

\[
= aX + b + Z
data/\text{noise (also random)}
Comparison: MSE vs MLE

\[ \hat{Y} = g(X) = aX + b \]

Minimum Mean Squared Error

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

- 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

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

- 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 $X$ is 1-D):
   \[
   \hat{Y} = g(X) = aX + b
   \]

2. Define maximum likelihood estimator:
   \[
   \theta_{MLE} = \arg \max_{\theta} \sum_{i=1}^{n} \log f(y^{(i)} | x^{(i)}, \theta)
   \]

3. Model error, $Z$:
   \[
   Y = aX + b + Z, \text{ where } Z \sim \mathcal{N}(0, \sigma^2)
   \]

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 $\text{arg max}_x f(x)$?

A. Graph and guess

B. Differentiate, set to 0, and solve

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

$$x = 0$$

C. Gradient ascent: educated guess & check

objective function
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
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
     - \( x += \eta \times \text{gradient} \)
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) $\theta$

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

3. Solve resulting (simultaneous) equations

To maximize:

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

If algebra is intractable, we can still find a maximum using gradient ascent!
Linear Regression, MLE (so far)

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

\[
\hat{Y} = g(\mathbf{X}) = aX + b
\]

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

\[
\theta_{\text{MLE}} = \arg \max_{\theta} LL(\theta)
\]

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

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

(\( \theta_{\text{MLE}} \) also maximizes log conditional likelihood)

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2022
Computing the MLE with gradient ascent

General approach for finding $\theta_{MLE}$, the MLE of $\theta$:

1. Determine formula for $LL(\theta)$
   
   \[ \sum_{i=1}^{n} \log f(y^{(i)} \mid x^{(i)}, \theta) \]

2. Differentiate $LL(\theta)$ w.r.t. (each) $\theta$
   
   \[ \frac{\partial}{\partial \theta_j} \sum_{i=1}^{n} \log f(y^{(i)} \mid x^{(i)}, \theta) \]

3. Solve resulting equations

   (computer) Gradient Ascent

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2022
1. Determine formula for log conditional likelihood

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

Optimization problem:  
\[ \arg \max_\theta \sum_{i=1}^{n} \log f(y^{(i)} | x^{(i)}, \theta) \]

Over the next few slides, we will show that our MLE linear regression \( \theta_{MLE} \) reduces to

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

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:  
\[
\arg \max_{\theta} \frac{1}{n} \sum_{i=1}^{n} \log f(y^{(i)} | x^{(i)}, \theta)
\]

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

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:**
\[ \arg \max_{\theta} \sum_{i=1}^{n} \log f(y^{(i)} | x^{(i)}, \theta) \]

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

\[ Y|X, \theta \sim \mathcal{N}(aX + b, \sigma^2) \]
\[ f(y^{(i)} | x^{(i)}, \theta) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(y^{(i)} - (ax^{(i)} + b))^2}{2\sigma^2}} \]

2. Substitute 1. into objective fn.

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

using natural log:
\[ \arg \max_{\theta} \left[ \sum_{i=1}^{n} -\log \sqrt{2\pi\sigma} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y^{(i)} - ax^{(i)} - b)^2 \right] \]
1. Determine formula for log conditional likelihood

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

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

3. Use argmax properties to simplify objective fn.

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

**Argmax refresher #1:**
Invariant to additive constants

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

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

4. Celebrate!

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

General approach for finding $\theta_{MLE}$, the MLE of $\theta$:

1. Determine formula for $LL(\theta)$
   
   log conditional likelihood
   
   $$\sum_{i=1}^{n} \log f(y^{(i)} | x^{(i)}, \theta)$$

   \[ h(\theta) = -\sum_{i=1}^{n} (y^{(i)} - ax^{(i)} - b)^2 \]

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

   \[ \frac{\partial}{\partial \theta_j} \sum_{i=1}^{n} \log f(y^{(i)} | x^{(i)}, \theta) \]

   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

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2022
2. Compute gradient

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

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

1. What is the derivative of the objective function w.r.t. \( a \)?
\[
\frac{\partial}{\partial a} \left[ -\sum_{i=1}^{n} (y^{(i)} - ax^{(i)} - b)^2 \right] = 
\]

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

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

1. What is the derivative of the objective function w.r.t. \( a \)?
\[
\frac{\partial}{\partial a} \left[- \sum_{i=1}^{n} (y^{(i)} - ax^{(i)} - b)^2 \right] =
\]

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

1. What is the derivative of the objective function w.r.t. \( a \)?
\[
\sum_{i=1}^{n} 2(y(i) - ax(i) - b)(x(i))
\]

2. What is the derivative of the objective function w.r.t. \( b \)?
\[
\sum_{i=1}^{n} 2(y(i) - ax(i) - b)
\]

**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 $\theta_{MLE}$, the MLE of $\theta$:

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

   \[
   \sum_{i=1}^{n} \log f(y^{(i)} | x^{(i)}, \theta)
   \]

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

   \[
   \frac{\partial}{\partial \theta_j} \sum_{i=1}^{n} \log f(y^{(i)} | x^{(i)}, \theta)
   \]

3. Solve resulting (simultaneous) equations

   (computer)

   Gradient Ascent

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

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

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

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2022
3. Gradient ascent with multiple parameters (if time)

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

\[ = \arg \max_{\theta} h(\theta) \]

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

initialize \( \theta \)
repeat many times:
compute gradient
\( \theta += \eta \times \text{gradient} \)

How does this work for multiple parameters?
3. Gradient ascent with multiple parameters

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

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

\[
\begin{align*}
a, b &= 0, 0 \quad \# \text{initialize } \theta \\
\text{repeat many times:} & \\
\text{gradient}_a, \text{gradient}_b &= 0, 0 \quad \# \text{TODO: fill in} \\
a &= \eta \times \text{gradient}_a \quad \# \theta += \eta \times \text{gradient} \\
b &= \eta \times \text{gradient}_b
\end{align*}
\]

How do we pseudocode the gradients we derived?
3. Gradient ascent with multiple parameters

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

\[
= \arg \max_\theta h(\theta)
\]

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

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

\[
a, b = 0, 0 \quad # \text{initialize } \theta
\]

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 += \eta * gradient_a \quad # \theta += \eta * gradient
b += \eta * gradient_b
```

Finish computing gradient before updating any part of \( \theta \).

(Spring 2021 demo)
Global land-ocean temperature prediction

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

\[ \begin{align*}
\text{Year 1} & : 338.8 & \text{Output} & : 0.26 \\
\text{Year 2} & : 340.0 & \quad & : 0.32 \\
\text{…} & \quad & \quad & \quad \\
\text{Year n} & : 340.76 & \text{Output} & : 0.14 \\
\end{align*} \]

\(X = (X_1)\)  
(assume one feature)

\[ Y \in \mathbb{R} \]

Minimizing Mean Square Error

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

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

\[ a_{MSE} = 0.01452 \]

\[ b_{MSE} = 0.17511 \]
3b. Interpret

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

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

\( a, b = 0, 0 \) # initialize \( \theta \)

repeat many times:

for each training example \((x, y)\):

\[
\text{diff} = y - (a \times x + b) \\
\text{gradient}_a += 2 \times \text{diff} \times x \\
\text{gradient}_b += 2 \times \text{diff}
\]

\( a += \eta \times \text{gradient}_a \) # \( \theta += \eta \times \text{gradient} \)

\( b += \eta \times \text{gradient}_b \)

Updates to \( a \) and \( b \) should include information from all \( n \) training datapoints.
3b. Interpret

**Optimization problem:**

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

\[ = \arg \max_{\theta} h(\theta) \]

**Gradient:**

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

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

---

**Initialization:**

\[ a, b = 0, 0 \]

# initialize \( \theta \)

**Repeat many times:**

```
gradients_a, gradients_b = 0, 0
for each training example (x, y):
  diff = y - (a * x + b)
  gradient_a += 2 * diff * x
  gradient_b += 2 * diff
```

\[ a += \eta \times \text{gradient}_a \]

# \( \theta \) += \( \eta \times \text{gradient}_a \)

\[ b += \eta \times \text{gradient}_b \]

---

How do we interpret the contribution of the i-th training datapoint?
3b. Interpret

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

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

\[ a, \ b = 0, 0 \quad \# \ initialize \ \theta \]
repeat many times:

gradient_a, gradient_b = 0, 0
for each training example (x, y):
\[ \text{diff} = y - (a \times x + b) \]
\[ \text{gradient}_a += 2 \times \text{diff} \times x \]
\[ \text{gradient}_b += 2 \times \text{diff} \]
\[ a += \eta \times \text{gradient}_a \quad \# \ \theta += \eta \times \text{gradient} \]
\[ b += \eta \times \text{gradient}_b \]

Prediction error!
\[ y^{(i)} - \hat{y}^{(i)} \]
3b. Interpret

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

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

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

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

a, b = 0, 0       # initialize \theta
repeat many times:

\[
\begin{align*}
\text{gradient}_a, \text{gradient}_b &= 0, 0 \\
\text{for each training example } (x, y): \quad \text{prediction}_{\text{error}} &= y - (a \times x + b) \\
\text{gradient}_a &=+ 2 \times \text{prediction}_{\text{error}} \times x \\
\text{gradient}_b &=+ 2 \times \text{prediction}_{\text{error}} \\
\end{align*}
\]

a += \eta \times \text{gradient}_a         # \theta += \eta \times \text{gradient}
b += \eta \times \text{gradient}_b
3b. Interpret

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

= \arg\max_\theta h(\theta)

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

\[
\begin{align*}
a, b &= 0, 0 \quad \# \text{initialize } \theta \\
\text{repeat many times:} \\
\text{gradient}_a, \text{gradient}_b &= 0, 0 \\
\text{for each training example } (x, y): \\
prediction\_error &= y - (a \times x + b) \\
\text{gradient}_a &=+ 2 \times \text{prediction}\_error \\
\text{gradient}_b &=+ 2 \times \text{prediction}\_error \\
a &=+ \eta \times \text{gradient}_a \quad \# \theta += \eta \times \text{gradient} \\
b &=+ \eta \times \text{gradient}_b
\end{align*}
\]

\[\hat{Y} = aX + b, \text{ so update to } a \text{ should also scale by } x^{(i)}\]
3b. Interpret

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

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

\( a, b = 0, 0 \) # initialize \( \theta \)
repeat many times:

gradient\_a, gradient\_b = 0, 0
for each training example \( (x, y) \):
prediction\_error = \( y - (a \times x + b) \)
gradient\_a += 2 \times prediction\_error \times x
gradient\_b += 2 \times prediction\_error \times 1

\( a += \eta \times \text{gradient\_a} \) # \( \theta += \eta \times \text{gradient} \)
\( b += \eta \times \text{gradient\_b} \)

\( \hat{Y} = ax + b \), 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) \( \theta \), set to 0

\[
\frac{\partial}{\partial a} E[(Y - aX - b)^2] = E \left[ \frac{\partial}{\partial a} (Y - aX - b)^2 \right] = E[-2(Y - aX - b)X] = -2E[XY] + 2aE[X^2] + 2bE[X]
\]

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

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

Show that $\theta_{MLE}$ maximizes the log conditional likelihood function:

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

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

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

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

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

Proof continued: (chain rule, log of product = sum of logs)

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

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

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

Lisa Yan, Chris Piech, Mehran Sahami, and Jerry Cain, CS109, Spring 2022