CSE 446: Machine Learning

Natasha Jaques

UNIVERSITY of WASHINGTON

Course Website

https://courses.cs.washington.edu/courses/cse446/25sp/

Everything you need to know is there:

- Lecture slides (blank and annotated versions will be posted)
- Links to places to get help (Ed Discussion, staff email)
- Homeworks and due dates
- Past exams to study from
- Textbook and other references
- Sections, office hours
- Grading, FAQ, etc.

Course Staff - Instructor

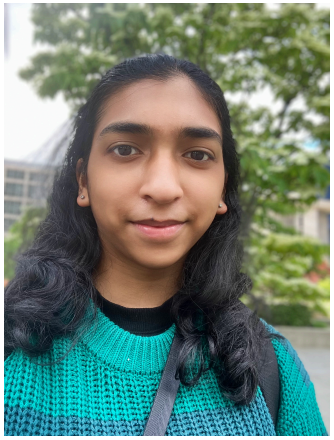
Natasha Jaques

Assistant Professor in CSE

Senior Research Scientist at Google DeepMind

Course Staff - Teaching Assistants

Donovan Clay
(Head TA)

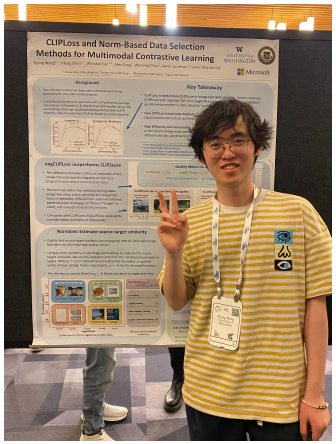


Cleah Taryn Winston

Emmanuel Azuh Mensah

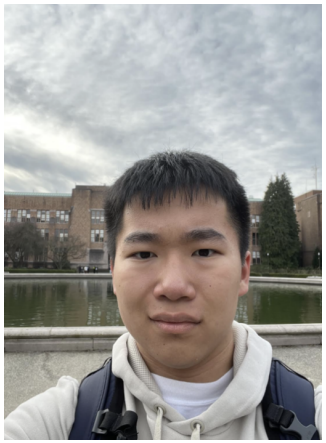
Sankar Vaishnav Harilal

Leo Maynard-Zhang



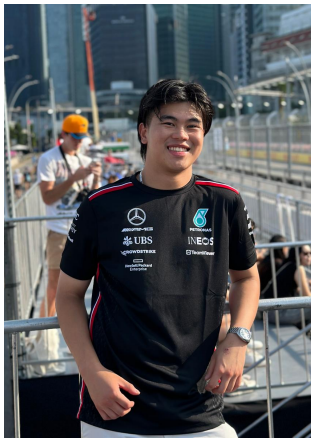
Yiping Wang

Saket Gollapudi



Anthony Xing

Varun Ananth

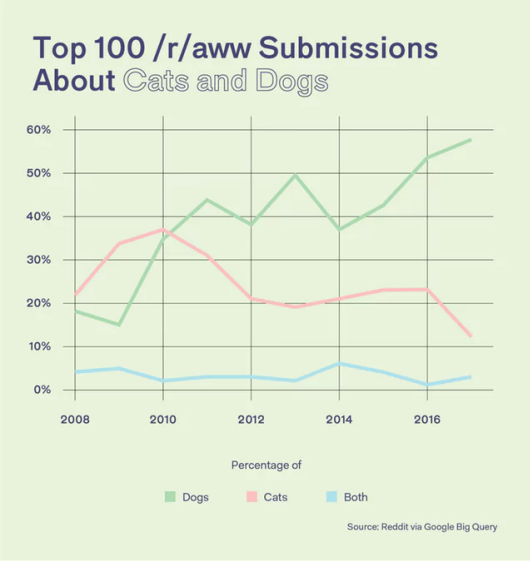


Marco Hendra Dani

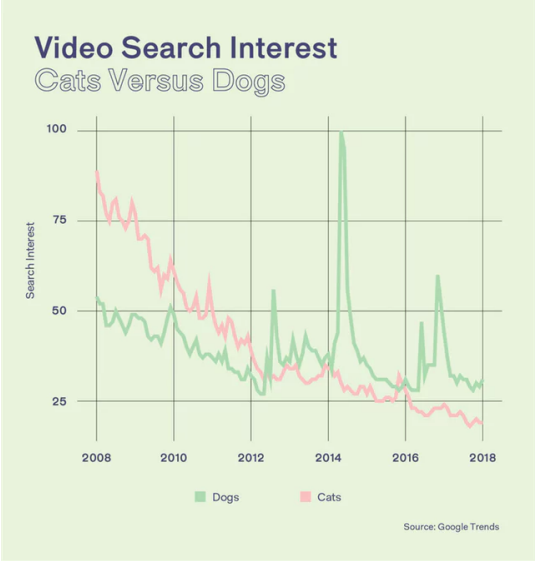
Traditional Algorithms

Social media mentions of Cats vs. Dogs

Reddit



Google



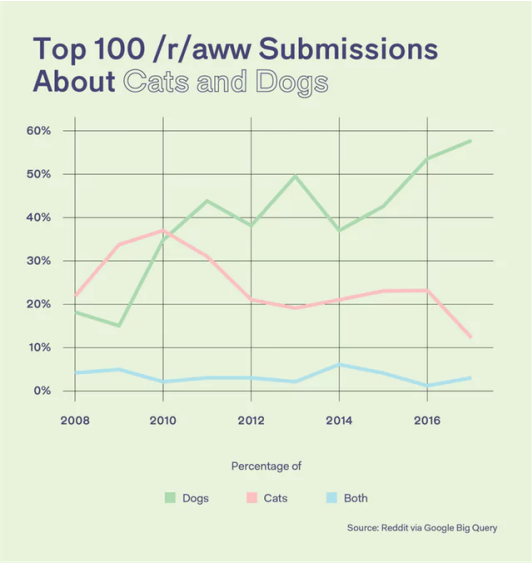
Twitter?

Graphics courtesy of https://theoutline.com/post/3128/dogs-cats-internet-popularity?zd=1

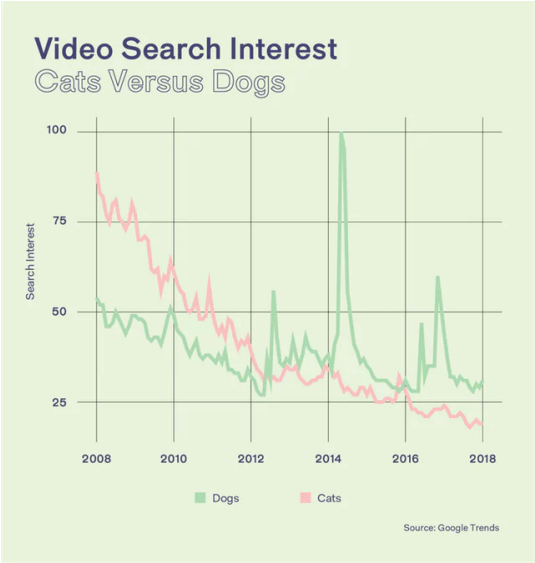
Traditional Algorithms

Social media mentions of Cats vs. Dogs

Reddit



Google



Twitter? 

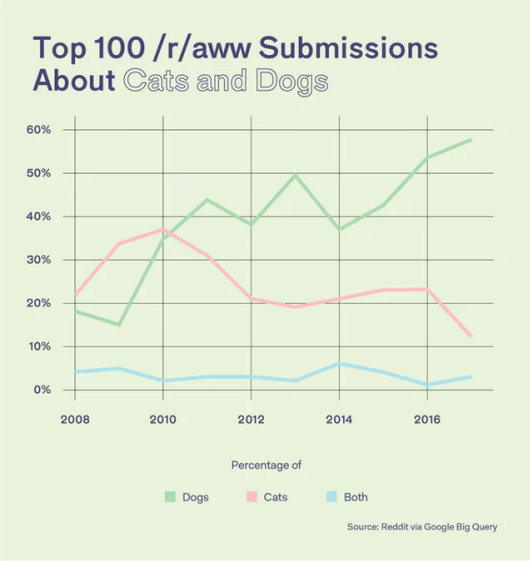
Write a program that sorts tweets into those containing "**cat**", "**dog**", or ***other***

Graphics courtesy of https://theoutline.com/post/3128/dogs-cats-internet-popularity?zd=1

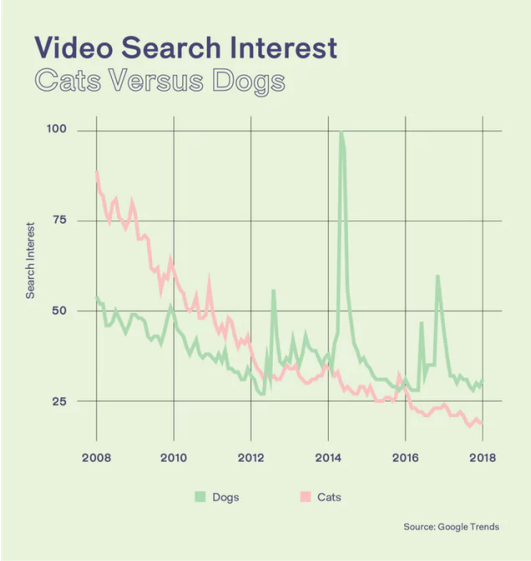
Traditional Algorithms

Social media mentions of Cats vs. Dogs

Reddit



Google



Twitter?

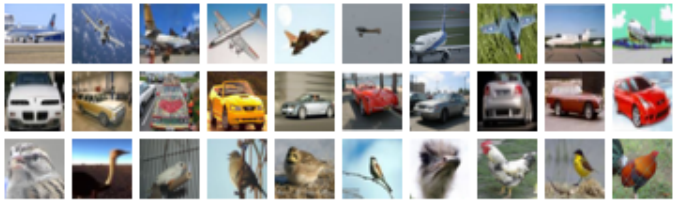
```
cats = []
dogs = []
other = []
for tweet in tweets:
    if "cat" in tweet:
        cats.append(tweet)
    elif "dog" in tweet:
        dogs.append(tweet)
    else:
        other.append(tweet)
return cats, dogs, other
```

Write a program that sorts tweets into those containing "**cat**", "**dog**", or ***other***

Graphics courtesy of https://theoutline.com/post/3128/dogs-cats-internet-popularity?zd=1

Machine Learning Algorithms

Write a program that sorts images into those containing "**birds**", "**airplanes**", or ***other.***



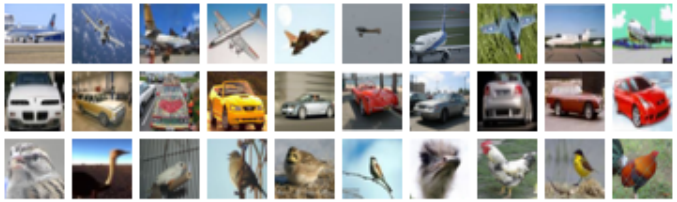
airplane

other

bird

Machine Learning Algorithms

Write a program that sorts images into those containing "**birds**", "**airplanes**", or ***other.***



airplane

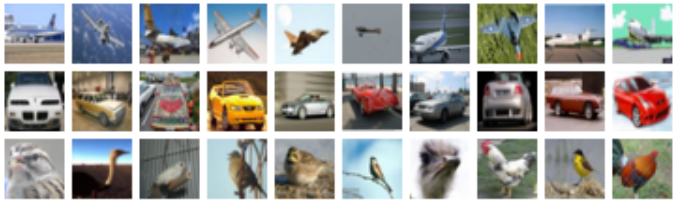
other

bird

```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elif plane in image:
        planes.append(image)
    else:
        other.append(image)
return birds, planes, other
```

Machine Learning Algorithms

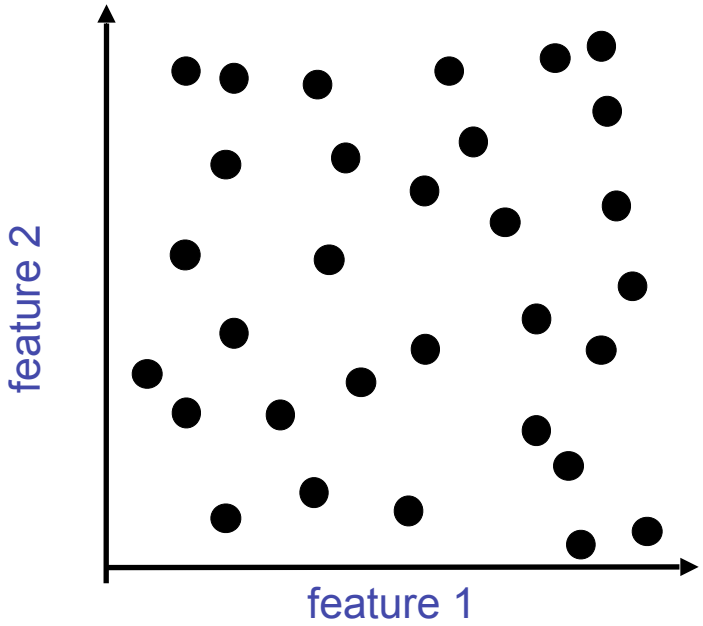
Write a program that sorts images into those containing "**birds**", "**airplanes**", or ***other.***



airplane

other

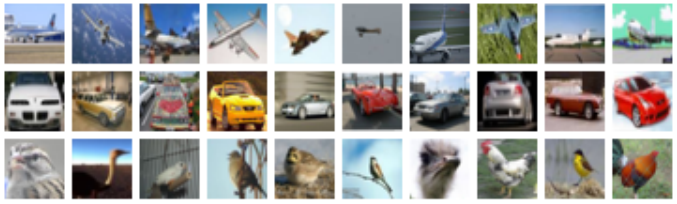
bird



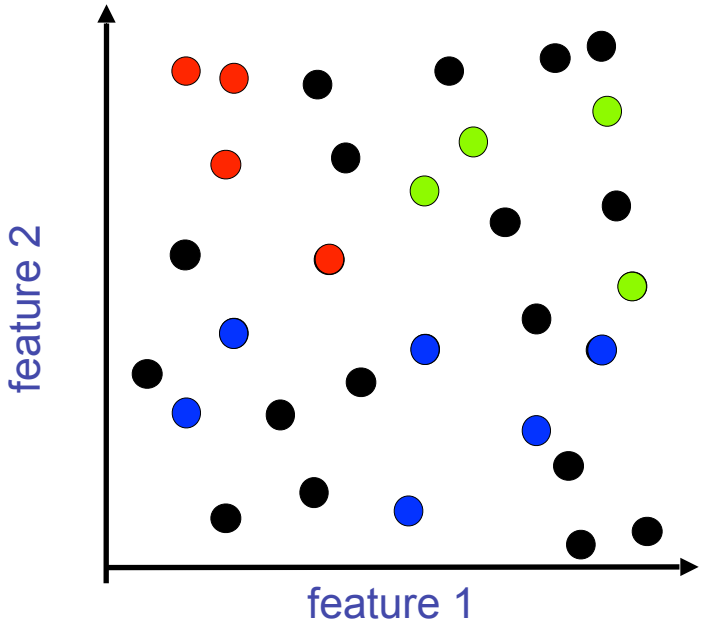

```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elif plane in image:
        planes.append(image)
    else:
        other.append(image)
return birds, planes, other
```

Machine Learning Algorithms

Write a program that sorts images into those containing "**birds**", "**airplanes**", or ***other.***



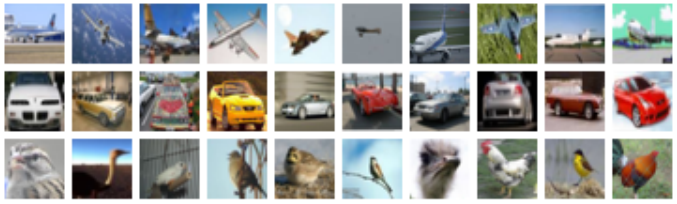
airplane
other
bird



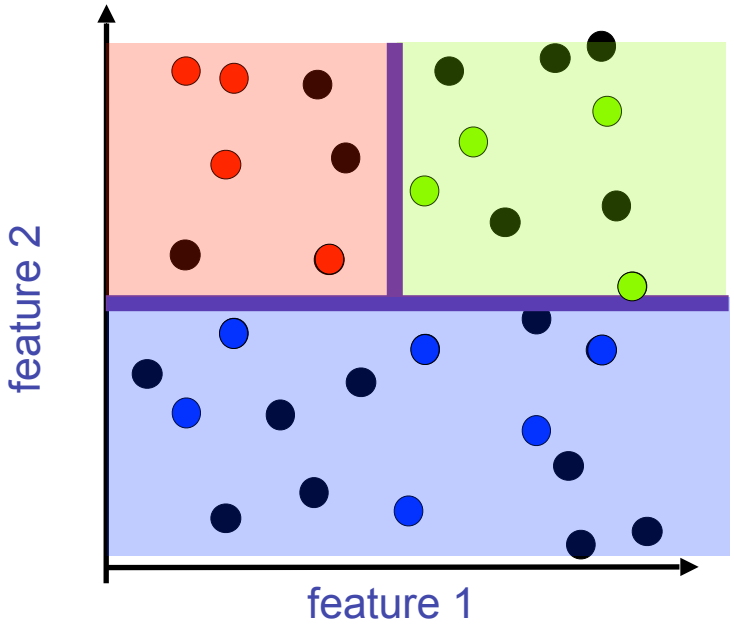

```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elif plane in image:
        planes.append(image)
    else:
        other.append(image)
return birds, planes, other
```

Machine Learning Algorithms

Write a program that sorts images into those containing "**birds**", "**airplanes**", or ***other.***



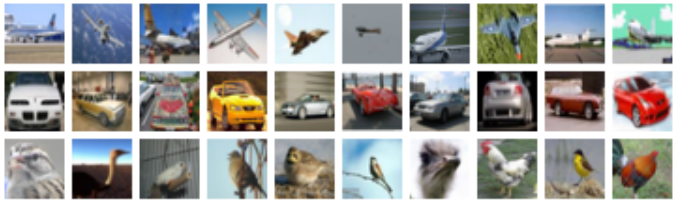
airplane
other
bird




```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elif plane in image:
        planes.append(image)
    else:
        other.append(image)
return birds, planes, other
```

Machine Learning Algorithms

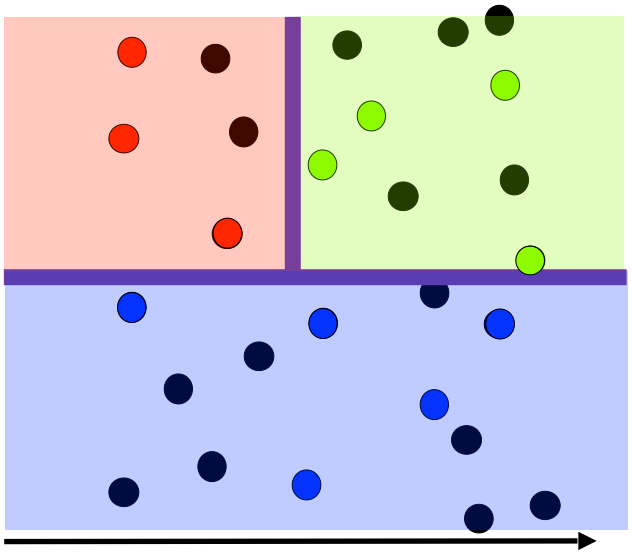
Write a program that sorts images into those containing "**birds**", "**airplanes**", or ***other.***



airplane

other

bird



```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elif plane in image:
        planes.append(image)
    else: …
```

The decision rule of

`if "cat" in tweet:`

is **hard coded by expert.**

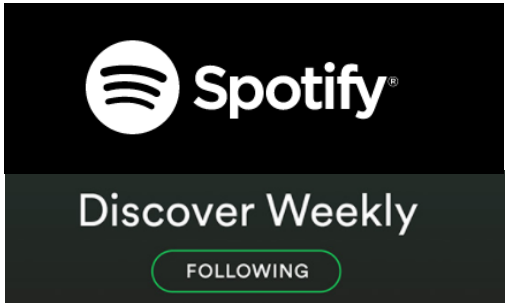
The decision rule of

`if bird in image:`

is **LEARNED using DATA**

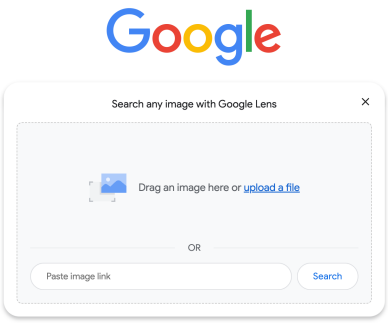
Machine Learning Ingredients

- **Data**: past observations
- **Hypotheses/Models**: devised to capture the patterns in data
- **Prediction**: apply model to forecast future observations

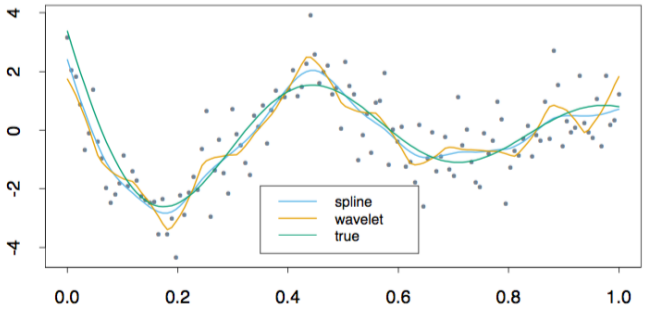


You may also like…

Even before ChatGPT, you were already interacting with ML algorithms every day

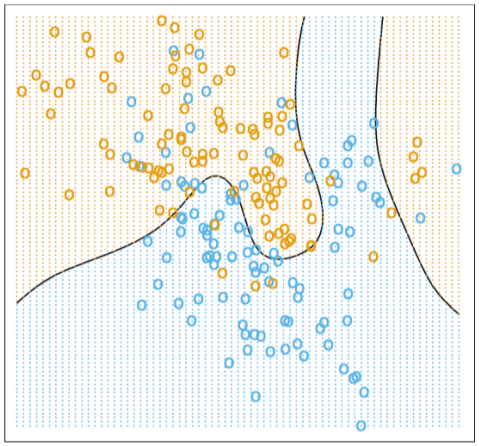


Flavors of ML



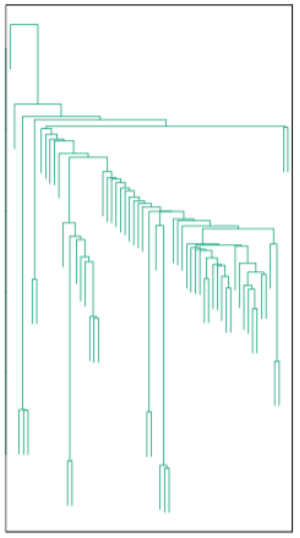
Regression

Predict continuous value:
ex: stock market, credit score, temperature, Netflix rating



Classification

Predict categorical value:
loan or not? spam or not? what disease is this?



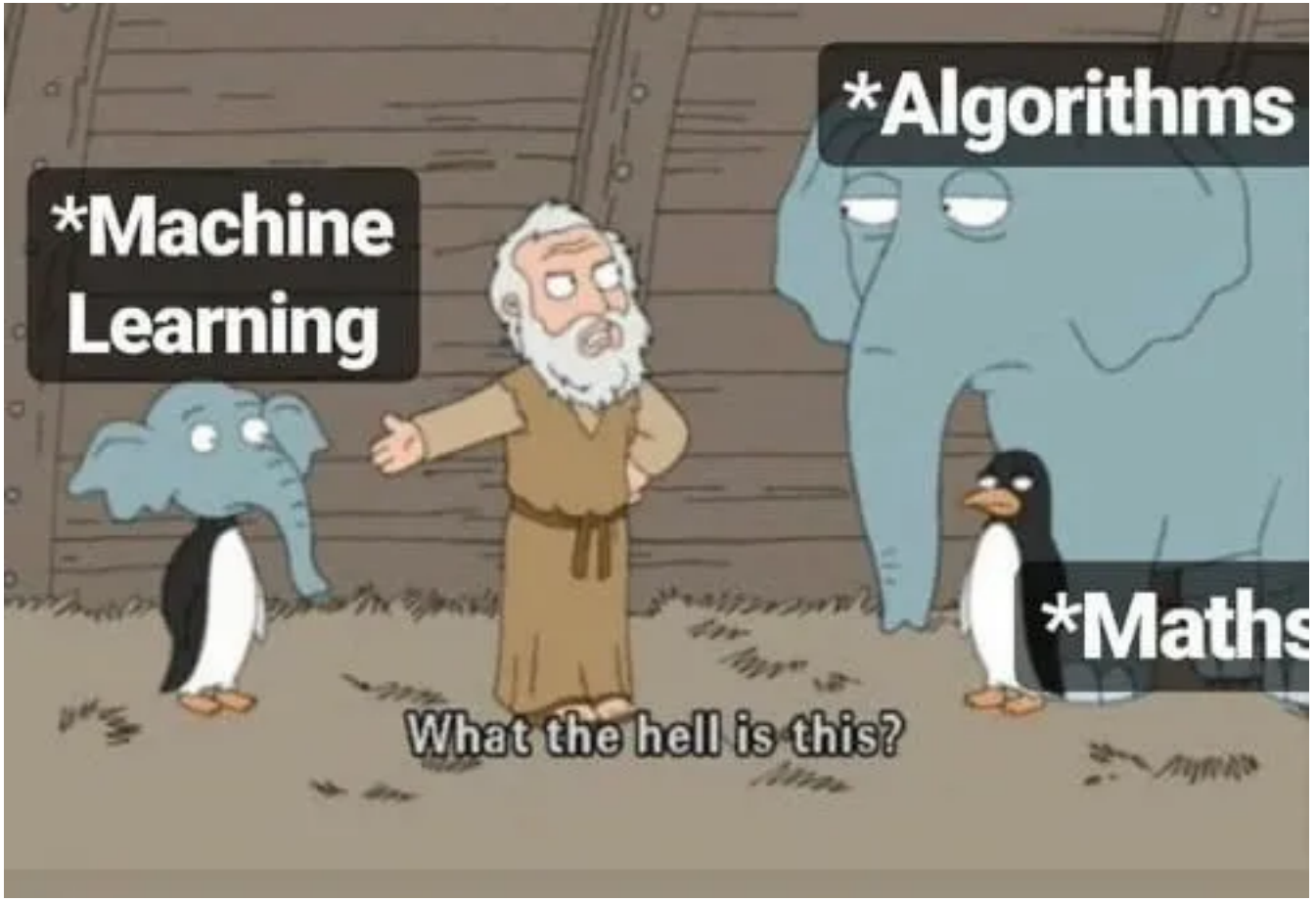
Unsupervised Learning

Predict structure:
tree of life from DNA, find similar images, community detection

Self-supervised:
Model distribution of large-scale data, like text or images (large vision and language models)

Mix of statistics (conceptual) and algorithms (programming)

Mix of statistics (conceptual) and algorithms (programming)



What this class is:

- **Fundamentals of ML:** bias/variance tradeoff, overfitting, optimization and computational tradeoffs, supervised learning (e.g., linear, boosting, deep learning), unsupervised models (e.g. k-means, EM, PCA)
- **Preparation for further learning:** the field is fast-moving, you will learn the foundations of ML to understand the latest results

What this class is not:

- **Survey course:** laundry list of algorithms, how to win Kaggle
- **An easy course:** familiarity with intro linear algebra and probability are assumed, homework will be time-consuming

Prerequisites

- Familiarity with:
  - Linear algebra
    - linear dependence, rank, linear equations
  - Multivariate calculus
  - Probability and statistics
    - Distributions, densities, marginalization, moments
  - Algorithms
    - Basic data structures, complexity

- Use HW0 as a refresher on the skills you will need
- **See assigned reading and website for additional review materials!**

Course Registration

- As of today, all enrollment restrictions have been dropped.
- If there are open spaces, any UW student may register for those spaces.
- All CSE course registration processes are managed centrally by CSE. **Do not email instructors, we cannot help.**

Resources:

- https://www.cs.washington.edu/academics/ugrad/advising/
- https://www.cs.washington.edu/academics/ugrad/courses/petition
- https://www.cs.washington.edu/students/ugrad/non-major-registration

Lectures

- Will be broadcast on Zoom.
- Will be recorded and posted shortly after class.
  - Find Zoom links and videos in Canvas—>Zoom.
- **In-person attendance is strongly encouraged.**
  - Material can be tough, and I can teach better if I know where you get lost
  - **Please ask lots of questions!**
  - We can afford to slow down
- **For exams, in-person attendance is mandatory.**

Grading

- 5 homeworks (60%)
  - Each contain both theoretical questions and will have programming.
  - Collaboration okay. You must write, submit, and understand your answers and code (which we may run).
  - **READ COLLABORATION POLICY ON WEBSITE**
  - If you get someone (or ChatGPT) to do your homework, did you actually learn it?
    - This stuff is foundational, if you want to work in this area need to actually understand
  - HW0 (8%), HW1 (13%), HW2 (13%), HW3 (13%), HW4 (13%)
- Midterm (20%) and Final (20%)

Homeworks

- **HW 0** is out (**Due next Wednesday 4/9 @ 11:59pm**)
  - Should be review (but being rusty is expected)
  - Work individually, treat as guide for what to brush up on
- **HW 1,2,3,4**. They are not easy or short. Start early.
- **Submit** to Gradescope.
- **Regrade requests** on Gradescope.
- **Late days:** 5 days total over the quarter; no more than 2 per assignment. If an assignment is submitted late and this exceeds your 5 late days, that assignment will receive 0 credit.
  - **Do not need to email us about using a late day**
- **Assignments due at 11:59pm**, submit early and often (do not email us at 12:05).

1. All code must be written in Python
2. All written work must be typeset (e.g., LaTeX)

See course website for tutorials and references.

Communication Channels

- **Announcements, questions about class, homework help -> EdStem**
  - "I think there is a typo in the homework?"
  - "What does this notation mean?"
  - "Is this an accurate description of how this works?
- **Personal concerns ->** cse446-staff@cs.washington.edu
  - "Was in hospital…", "Laptop was stolen…"
  - Do not email to ask if you can use one of your allowed late days
  - Do not email professor directly
- **Office hours**
  - "How do I get started on problem 2?"
  - "Am I on the right track?"
  - "I have this problem at work—can you point me in the right direction?"
  - We will not be able to give detailed code debugging help
- **Regrade requests**
  - Directly submit on Gradescope
- **Anonymous feedback** (https://feedback.cs.washington.edu/)
  - "Your real-world example X lacked nuance. I would like you to…"

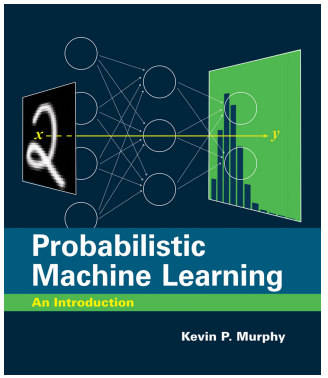
Textbooks

- **Free PDF** Textbook we will assign most reading from:

Probabilistic Machine Learning: An Introduction

Kevin Murphy

PDF, also in print



- So many more resources on the website!

The textbook for the course is:

- **[Murphy] Probabilistic Machine Learning: An Introduction, Kevin Murphy, 2022 (Note, this is not the 2012 edition used in past years). Follow the link to obtain a free, PDF pre-print as well as options to purchase a hard copy.**

For a gentler introduction to machine learning the following text is available for free online:

- [CIML] A Course in Machine Learning by Hal Daume III.

The following three texts are also excellent general machine learning texts and their PDFs are available for **free** online.

- [B] Pattern Recognition and Machine Learning, Christopher Bishop.
- [HTF] The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani, Jerome Friedman.
- [EH] Computer Age Statistical Inference: Algorithms, Evidence and Data Science, Bradley Efron, Trevor Hastie.
- [ZLLS] Dive into Deep Learning, Aston Zhang, Zach Lipton, Mu Li, Alex Smola.

You may also find these reference materials useful throughout the quarter.

- **Machine Learning (and related topics)**
  - Understanding Machine Learning: From Theory to Algorithms, Shai Shalev-Shwartz, Shai Ben-David. An introduction to theoretical machine learning.
  - Fairness and Machine Learning, Solon Barocas, Moritz Hardt, Arvind Narayanan
  - Foundations of Data Science, by Avrim Blum, John Hopcroft and Ravi Kannan. This freely available pdf has nice chapters on machine learning (chapter 5), clustering (chapter 7) and SVD (chapter 3).
  - Crib sheet of math for ML by Iain Murray
- **Linear Algebra and Matrix Analysis**
  - These wonderful videos by 3blue1brown provide a gentle and highly intuitive overview of linear algebra. (The same person created most of the videos on multivariable calculus on Khan Academy -- also excellent).
  - Linear Algebra Review and Reference by Zico Kolter and Chuong Do (free). Light refresher for linear algebra and matrix calculus if you're a bit rusty.
  - Linear Algebra, David Cherney, Tom Denton, Rohit Thomas and Andrew Waldron (free). Introductory linear algebra text.
  - Matrix Analysis, Horn and Johnson. A great reference from elementary to advanced material.
  - The Matrix Cookbook, Pederson and Pederson. Contains many useful matrix manipulation identities, including derivatives
- **Probability and Statistics**
  - 20su offering of CSE312 materials and video lectures by Alex Tsun. Slides, full notes, and short video lecture snippets on basic probabilty and statistics. This course was recently redesigned in part to better prepare students for CSE 446, making it an excellent resource for review.
  - Probability Review by Arian Maleki and Tom Do. (From Andrew Ng's machine learning class.)
  - Section notes from Anna Karlin's 18au offering of 312:
    Counting, Combinatorics + intro probability, Conditional probability; Random variables & linearity of expectation, Variance and discrete r.v.s, Conditional expectation, Joint distributions, Continuous random variables, CLT, tail bounds and MLE.
  - All of Statistics, Larry Wasserman. Chapters 1-5 are a great probability refresher and the book is a good reference for statistics.
  - A First Course in Probability, Sheldon Ross. Elementary concepts (previous editions are a couple bucks on Amazon)
- **Optimization**
  - Numerical Optimization, Nocedal, Wright. Practical algorithms and advice for general optimization problems.
  - Convex Optimization: Algorithms and Complexity, Sébastien Bubeck. Elegant proofs for the most popular optimization procedures used in machine learning.
- **Python**
  - www.learnpython.org "Whether you are an experienced programmer or not, this website is intended for everyone who wishes to learn the Python programming language."
  - NumPy for Matlab users
- **Latex**
  - Learn Latex in 30 minutes
  - Overleaf. An online Latex editor.
  - Standalone Latex editor on your local machine
  - Latex Math symbols
  - Detexify LaTeX handwritten symbol recognition

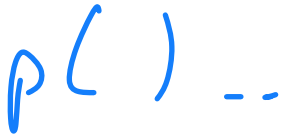
Enjoy!

- ML is:
  - Becoming ubiquitous in science, engineering and beyond
  - Transforming the world
- This class should give you a basic foundation for understanding and applying ML

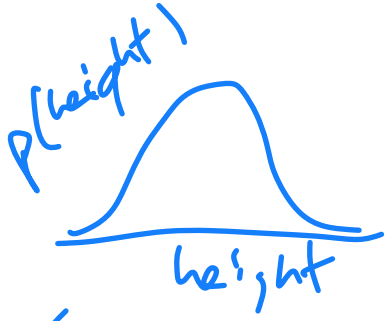
Probability review

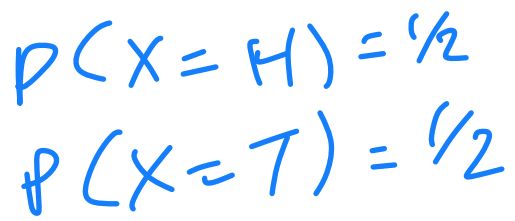
Definitions

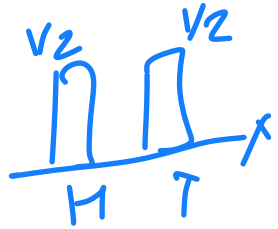
- **Random Variable**: A variable that takes on different values determined randomly.
  - Example: The height of a person from the US.



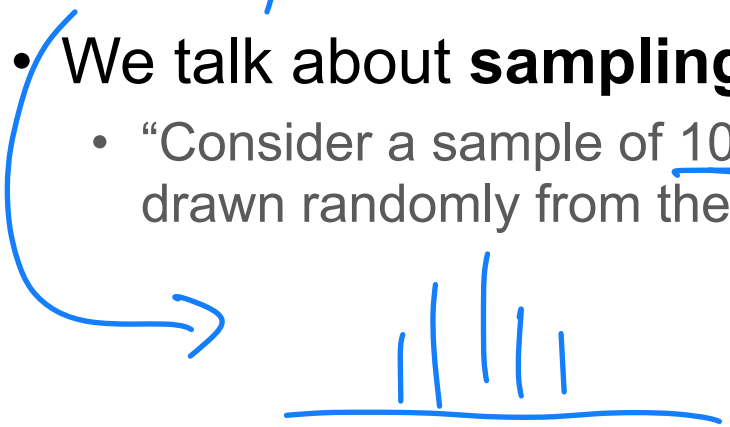
- **Distribution**: The different values a random variable can take on along with the probability of that value.







- We talk about **sampling** from a distribution:
  - "Consider a sample of 100 different heights of people from the US drawn randomly from the distribution of all heights."



Independence

Let X and Y be **random variables**

Ex. X is the outcome of the first roll of a 6-sided dice, Y is the outcome of the second roll of the dice

(X and Y take values in {1,2,3,4,5,6} each with equal probability)

An **event** is statement about the world that holds or not:

Define events $A = \{X \in \{3,4\}\}$,

$B = \{X = 1\}$,

$C = \{Y \in \{3,4\}\}$

Every event is assigned a **probability**:

$P(A) = P(X \in \{3,4\}) =$ 2/6 = 1/3

Independence

Let X and Y be **random variables**

Ex. X is the outcome of the first roll of a 6-sided dice, Y is the outcome of the second roll of the dice

(X and Y take values in {1,2,3,4,5,6} each with equal probability)

An **event** is statement about the world that holds or not:

Define events $A = \{X \in \{3,4\}\}$,

$B = \{X = 1\}$,

$C = \{Y \in \{3,4\}\}$

Every event is assigned a **probability**:

$P(A) = P(X \in \{3,4\}) = 1/3$

For any events U, V we have $P(U \cup V) = P(U) + P(V) - P(U \cap V)$

Independence

Let X and Y be **random variables**

Ex. X is the outcome of the first roll of a 6-sided dice, Y is the outcome of the second roll of the dice

(X and Y take values in {1,2,3,4,5,6} each with equal probability)

An **event** is statement about the world that holds or not:

Define events $A = \{X \in \{3,4\}\}$,

$B = \{X = 1\}$,

$C = \{Y \in \{3,4\}\}$

Any events U, V are **independent** if $P(U \cap V) = P(U)P(V)$

Are A, B independent?

B, C?

A, C?

$P(A) = 1/3$ $P(B) = 1/6$

$P(A \cap B) = 0$

Independence

Let X and Y be **random variables**

Ex. X is the outcome of the first roll of a 6-sided dice, Y is the outcome of the second roll of the dice

(X and Y take values in {1,2,3,4,5,6} each with equal probability)

An **event** is statement about the world that holds or not:

Define events $A = \{X \in \{3,4\}\}$,

$B = \{X = 1\}$,

$C = \{Y \in \{3,4\}\}$

Any events U, V are **independent** if $P(U \cap V) = P(U)P(V)$

Are A, B independent? (no)

B, C? (yes)

A, C? (yes)

Independence

Let X and Y be **random variables**

Ex. X is the outcome of the first roll of a 6-sided dice, Y is the outcome of the second roll of the dice

(X and Y take values in {1,2,3,4,5,6} each with equal probability)

An **event** is statement about the world that holds or not:

Define events $A = \{X \in \{3,4\}\}$,

$B = \{X = 1\}$,

$C = \{Y \in \{3,4\}\}$

Any events U, V are **independent** if $P(U \cap V) = P(U)P(V)$

We define the **conditional probability** of event U given V as

$$P(U \mid V) = \frac{P(U \cap V)}{P(V)}$$

What is $P(X \leq 4 \mid X \geq 3)$?

$$= \frac{P(3 \leq X \leq 4)}{P(X \geq 3)} = \frac{1/3}{4/6} = 1/2$$

Independence

Let X and Y be **random variables**

Ex. X is the outcome of the first roll of a 6-sided dice, Y is the outcome of the second roll of the dice

(X and Y take values in {1,2,3,4,5,6} each with equal probability)

An **event** is statement about the world that holds or not:

Define events $A = \{X \in \{3,4\}\}$,

$B = \{X = 1\}$,

$C = \{Y \in \{3,4\}\}$

Any events U, V are **independent** if $P(U \cap V) = P(U)P(V)$

We define the **conditional probability** of event U given V as

$$P(U \mid V) = \frac{P(U \cap V)}{P(V)}$$

What is $P(X \leq 4 \mid X \geq 3) = \frac{P(3 \leq X \leq 4)}{P(X \geq 3)} = \frac{1/3}{2/3} = 1/2$

Independence

Let X and Y be **random variables**

Ex. X is the outcome of the first roll of a 6-sided dice, Y is the outcome of the second roll of the dice

(X and Y take values in {1,2,3,4,5,6} each with equal probability)

An **event** is statement about the world that holds or not:

Define events $A = \{X \in \{3,4\}\}$,

$B = \{X = 1\}$,

$C = \{Y \in \{3,4\}\}$

Any events U, V are **independent** if $P(U \cap V) = P(U)P(V)$

We define the **conditional probability** of event U given V as

$$P(U \mid V) = \frac{P(U \cap V)}{P(V)}$$

if ind then $\frac{P(U)P(V)}{P(V)} \rightarrow P(U)$

Observe: if U, V are independent then $P(U \mid V) = P(U)$.

In words: if independent, V tells you nothing about U (and vice versa)

Mean, variance

Mean $\mathbb{E}[X], \mu$

The expected value of X, each value is weighted by the probability of seeing it.

$$\mathbb{E}[X] = \sum_x P(X = x)x$$

Variance $\mathrm{Var}(X), \sigma^2$

The expected squared deviation of X from its mean.

$$\mathbb{E}[(X - \mathbb{E}[X])^2]$$

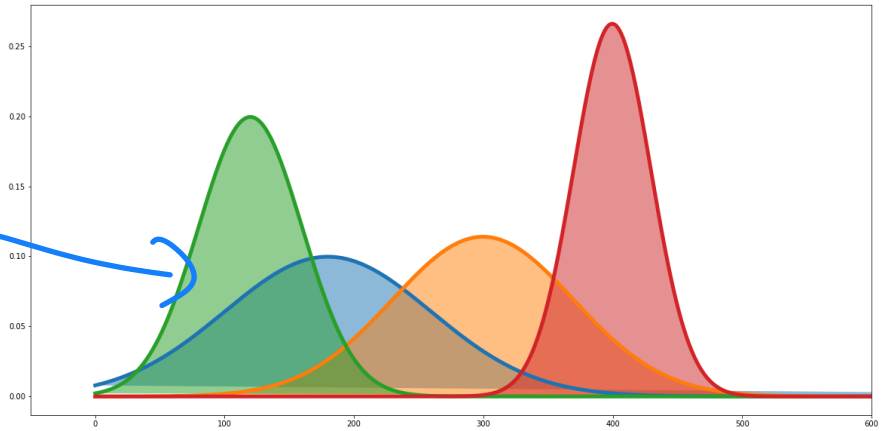
Median M

The value of X that is separating the higher half of it's range from the lower half.

$$P(X \leq M) = .5$$

Mean, variance

The mean is a prediction of the value of the random variable. Answers the question "What do I expect the height of a random person to be?"



The variance captures the spread in your data. Also captures the error in the prediction using the mean. "How much do people's heights deviate?"

$$\mathbb{E}[(X - \mathbb{E}[X])^2]$$

Maximum Likelihood Estimation

Your first consulting job

- *Client*: I have special coin, if I flip it, what's the probability it will be heads?
- *You (a machine learner)*: I need to collect **data.**

H H

- *You*: The probability is: 100% heads!

Your first consulting job

- *Client*: Uhhhh…. You sure about that? I just got a tails.
- *You (a machine learner)*: I need to collect **more data.**
  - *flips coin 5 times, get HHTHT

- *You*: The probability is: 60% Heads, 40% Tails!

Your first consulting job

- *Client*: Uhhhh…. You sure about that? I just got a tails.
- *You (a machine learner)*: I need to collect **more data.**
  - *flips coin 10000 times, it comes up Heads 70% of the time

- *You*: The probability is: 70% Heads, 30% Tails!

- *Client*: **Why should I believe you?**

- *You (a machine learner)*: Let's do some math!

Coin – Bernoulli Distribution

- **Data**: sequence *D= (HHTHT…)*, **k heads** out of **n flips**
- **Hypothesis:** P(Heads) = θ, P(Tails) = 1-θ
  - Flips are i.i.d.:
    - Independent events
    - Identically distributed according to Bernoulli distribution

- $P(\mathcal{D}|\theta) =$ $P(HHTHT|\theta)$ ↘ s.t. indep
  $= P(H) \cdot P(H) \cdot P(T) \cdot P(H) \cdot P(T)$
  $= \theta^{k}(1-\theta)^{n-k}$

Maximum Likelihood Estimation

- **Data**: sequence *D= (HHTHT…)*, **k heads** out of **n flips**
- **Hypothesis:** P(Heads) = θ, P(Tails) = 1-θ
- **Likelihood:**

$$P(\mathcal{D}|\theta) = \theta^k(1-\theta)^{n-k}$$

likelihood

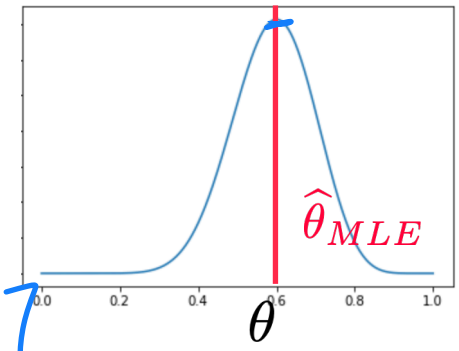
- **Maximum likelihood estimation (MLE):** Choose θ that maximizes the probability of observed data:

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

$$= \arg\max_\theta \log P(\mathcal{D}|\theta)$$

log bc easy

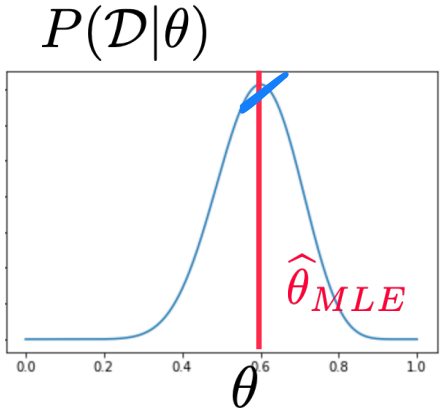
$= \arg\max_\theta \log [\theta^k (1-\theta)^{1-k}]$



MLE: Your first learning algorithm

$$\widehat{\theta}_{MLE} = \arg\max_{\theta} \log P(\mathcal{D}|\theta)$$

$$= \arg\max_{\theta} \log \theta^{k}(1-\theta)^{n-k}$$



- How do we find θ that maximizes likelihood?
- Use the fact that derivative is zero at maxima (also at minima)
- Set derivative to zero, and find θ satisfying:

$$\frac{d}{d\theta} \log P(\mathcal{D}|\theta) = 0$$

→ MLE

MLE: Your first learning algorithm

- **First manipulate the log likelihood to make it easy to work with:**

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \log P(\mathcal{D}|\theta)$$

$$= \arg\max_{\theta} \log \theta^k (1-\theta)^{n-k}$$

$$= \arg\max_{\theta} k \log\theta + (n-k)\log(1-\theta)$$

$\frac{d}{dx}\log x = \frac{1}{x}$

- **Then set derivative to 0, and find θ satisfying:** $\frac{d}{d\theta}\log P(\mathcal{D}|\theta) = 0$

$$\frac{d}{d\theta}\left[\frac{k}{\theta} - \frac{(n-k)}{(1-\theta)}\right] = 0$$

$$k - k\theta = n\theta - k\theta$$

$$\hat{\theta}_{MLE} = \frac{k}{n} \rightarrow \frac{3}{5} \rightarrow 60\%$$

How good is MLE? Well, it's unbiased

- We treat MLE $\hat{\theta}_{\text{MLE}}$ as a random variable, where there is a ground truth parameter θ^* that generates the data $\mathcal{D} = (HHTTH\ldots)$ of a fixed size n

$\hat{\theta}_{MLE} = \frac{k}{n}$

random variable

- What can we say about this random variable $\hat{\theta}_{\text{MLE}}$?
- First good property of MLE for Binomial: **unbiased**
  - Definition: **bias** of our MLE is

"true predictor"

$$\text{Bias}(\hat{\theta}_{\text{MLE}}) := \mathbb{E}_{\mathcal{D}\sim P_{\theta^*}}[\hat{\theta}_{\text{MLE}}] - \theta^* = \mathbb{E}\left[\frac{k}{n}\right] - \theta^* \to 0$$

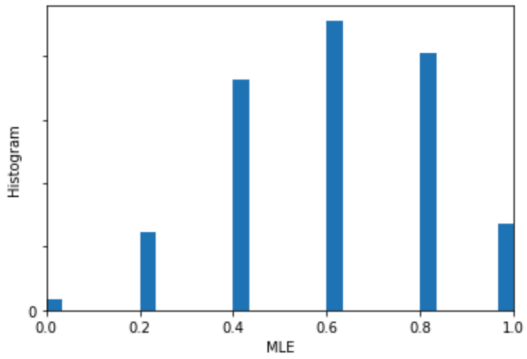
$$= \theta^* - \theta^* \to 0$$

- **Expectation** describes how the estimator behaves *on average*

How many flips do I need?

- Consider running many experiments with $\theta^* = \frac{3}{5}$, and observe many instances of the random variable

$$\widehat{\theta}_{MLE} = \frac{k}{n}$$

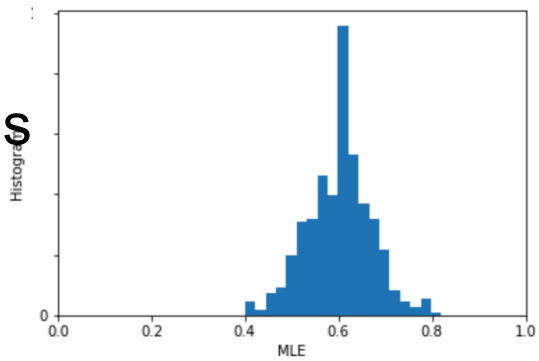


- *Client*: I flipped the coin 5 times and got 2 heads.

$$\widehat{\theta}_{MLE} =$$

- *Client*: I flipped the coin 50 times and got 30 heads

$$\widehat{\theta}_{MLE} =$$

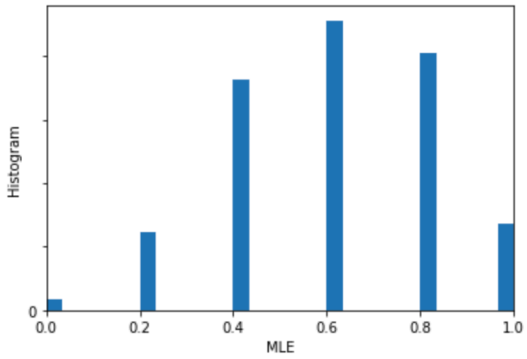


- *Client:* they are both unbiased, which one is right? Why?
- Variance goes down with larger n $\sqrt{\text{Var}(\widehat{\theta}_{\text{MLE}})} = \sqrt{\frac{\theta^*(1-\theta^*)}{n}}$

How many flips do I need?

- Consider running many experiments with $\theta^* = \frac{3}{5}$, and observe many instances of the random variable

$$\widehat{\theta}_{MLE} = \frac{k}{n}$$

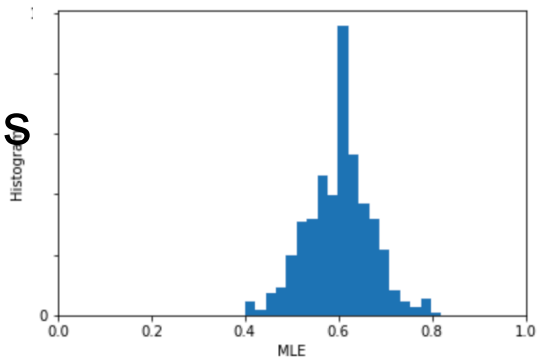


- *Client*: I flipped the coin 5 times and got 2 heads.

$$\widehat{\theta}_{MLE} =$$

- *Client*: I flipped the coin 50 times and got 30 heads

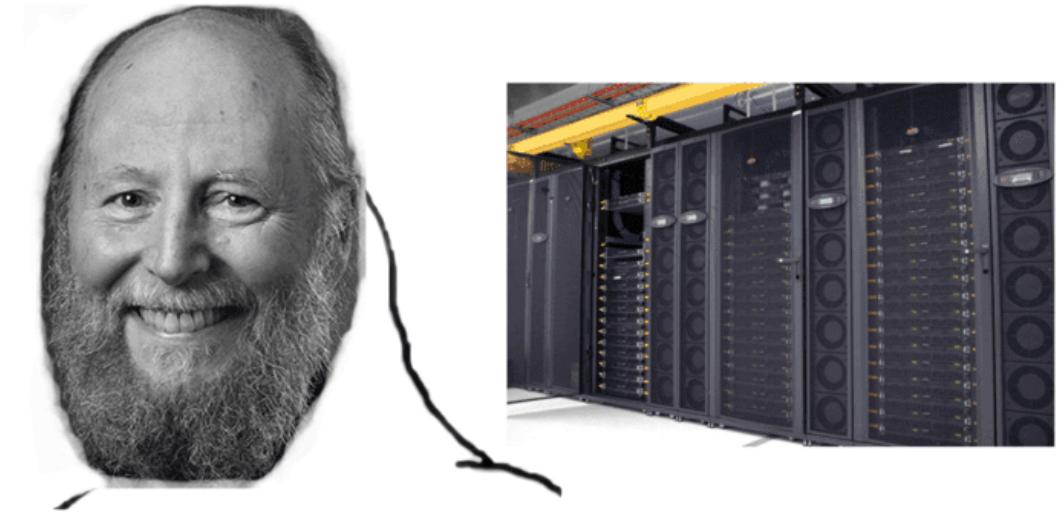
$$\widehat{\theta}_{MLE} =$$



- *Client:* they are both unbiased, which one is right? Why?
- Variance goes down with larger n $\sqrt{\text{Var}(\widehat{\theta}_{\text{MLE}})} = \sqrt{\frac{\theta^*(1-\theta^*)}{n}}$

Fundamental machine learning truth

- More data -> better performance
  - “The Bitter Lesson”
  - https://www.cs.utexas.edu/~eunsol/courses/data/bitter_lesson.pdf



Maximum Likelihood Estimation

Observe $X_1, X_2, \ldots, X_n$ drawn IID from $f(x;\theta)$ for some "true" $\theta = \theta_*$

Likelihood function $L_n(\theta) = \prod_{i=1}^{n} f(X_i;\theta)$

Log-Likelihood function $l_n(\theta) = \log(L_n(\theta)) = \sum_{i=1}^{n} \log(f(X_i;\theta))$

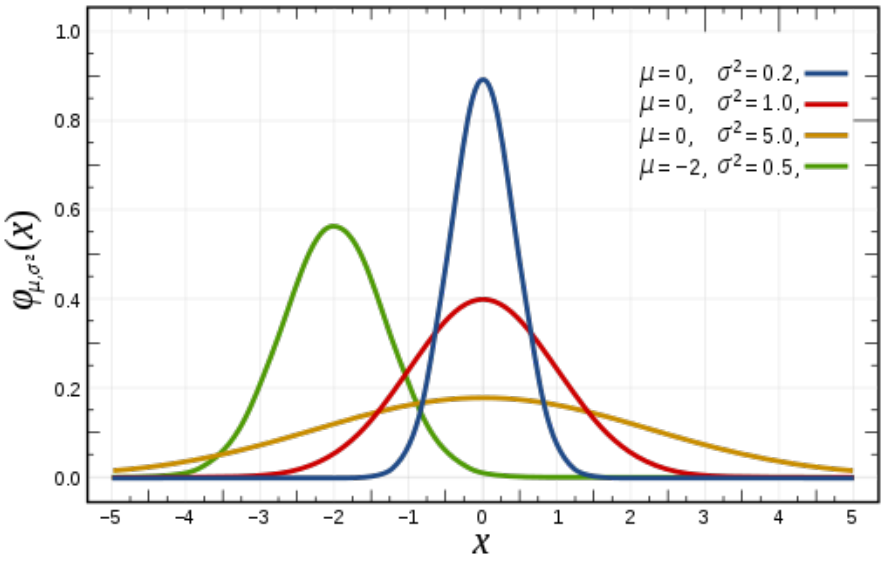
Maximum Likelihood Estimator (MLE) $\widehat{\theta}_{MLE} = \arg\max_{\theta} L_n(\theta)$

What about continuous variables?

- *Client*: What if I am measuring a **continuous variable**?
- ***You*: Let me tell you about Gaussians…**

$$P(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

PDF



given a set of iid samples from Gaussian fit $\theta = [\mu, \sigma]$

Some properties of Gaussians

- Affine transformation (multiplying by scalar and adding a constant)
  - $X \sim N(\mu, \sigma^2)$
  - $Y = aX + b \quad \rightarrow \quad Y \sim N(a\mu+b, a^2\sigma^2)$
- Sum of Gaussians
  - $X \sim N(\mu_X, \sigma^2_X)$
  - $Y \sim N(\mu_Y, \sigma^2_Y)$
  - $Z = X+Y \quad \rightarrow \quad Z \sim N(\mu_X+\mu_Y, \sigma^2_X+\sigma^2_Y)$

MLE for Gaussian

- Prob. of i.i.d. samples $D=\{x_1,\ldots,x_n\}$ (e.g., temperature):

$$P(\mathcal{D}|\mu,\sigma) = P(x_1,\ldots,x_n|\mu,\sigma)$$

$$= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \prod_{i=1}^{n} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Likelihood

by i.i.d

$P(x_1|\theta)P(x_2|\theta)\ldots P(x_n|\theta)\ldots$

- Log-likelihood of data:

$$\log P(\mathcal{D}|\mu,\sigma) = -n\log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2}$$

LL

- What is $\widehat{\theta}_{MLE}$ for $\theta=(\mu,\sigma^2)$? Draw a picture!

MLE for Gaussian

Generate $\mathcal{D} = \{x_1, ..., x_n\}$, where

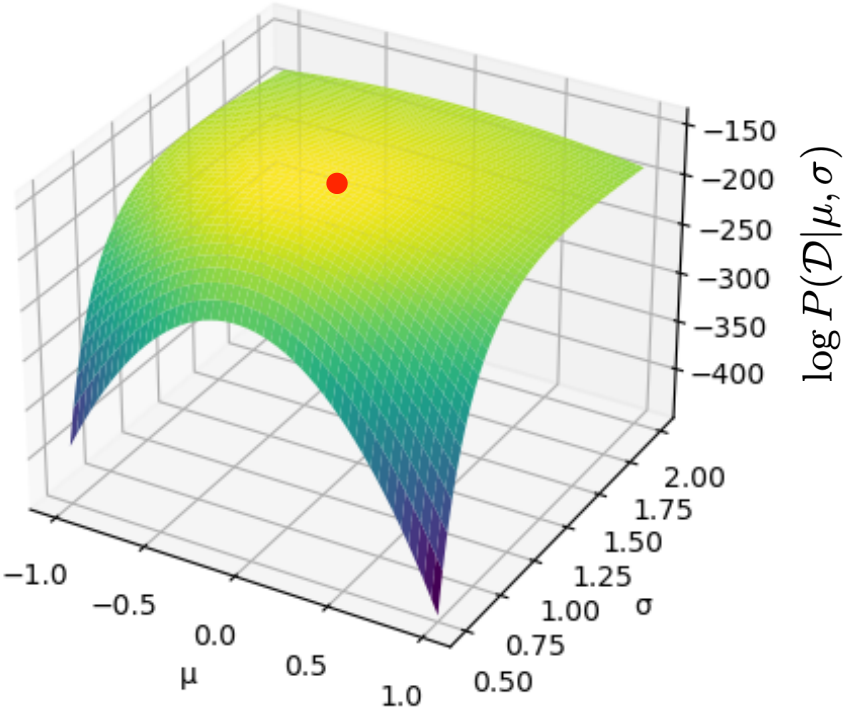
$$n = 100$$

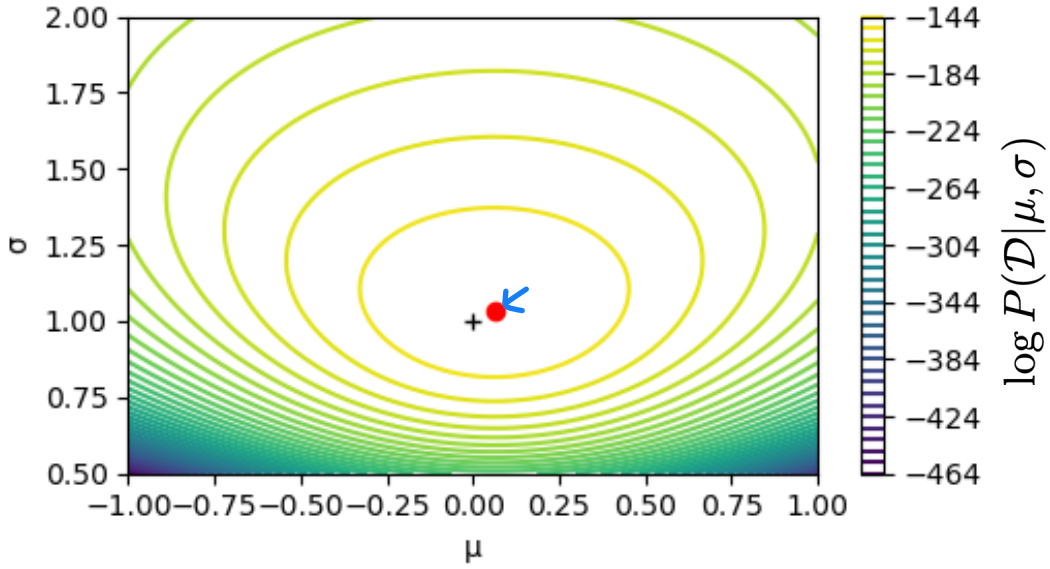
$$x_i \sim \mathcal{N}(\mu, \sigma^2)$$

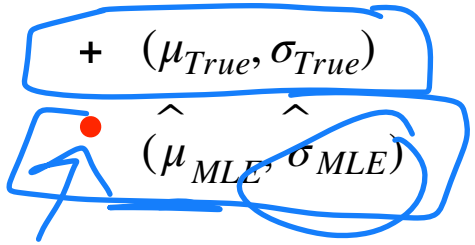
$$\mu = 0$$

$$\sigma^2 = 1$$

$$\log P(\mathcal{D}|\mu, \sigma) = -n \log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2\sigma^2}$$







Your second learning algorithm: MLE for mean of a Gaussian

- What's MLE for mean? Set **partial derivative** to zero.

$$\frac{\partial}{\partial \mu} \log P(\mathcal{D} \mid \mu, \sigma) = \frac{\partial}{\partial \mu} \left[-n \log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2\sigma^2} \right]$$

$$= -\sum_{i=1}^{n} \frac{2(x_i - \mu)}{2\sigma^2}$$

$$= -n\mu + \sum_{i=1}^{n} x_i = 0$$

$$\sigma^2$$

$$\hat{\mu}_{MLE} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

-> empirical mean

MLE for variance

- Again, set partial derivative to zero:

$$\frac{\partial}{\partial \sigma} \log P(\mathcal{D} \mid \mu, \sigma) = \frac{\partial}{\partial \sigma} \left[-n \log(\sigma\sqrt{2\pi}) - \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{2\sigma^2} \right]$$

$$= \frac{-n}{\sigma} + \sum_{i=1}^{n} \frac{(x_i - \mu)^2}{\sigma^3}$$

$$= -n\sigma^2 + \sum_{i=1}^{n} (x_i - \mu)^2 = 0$$

$$\sigma^2_{MLE} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu}_{MLE})^2$$

Learning Gaussian parameters

- MLE:

$$\widehat{\mu}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\widehat{\sigma^2}_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \widehat{\mu}_{MLE})^2$$

- MLE for the variance of a Gaussian is **biased**

$$\mathbb{E}[\widehat{\sigma^2}_{MLE}] \neq \sigma^2$$

- Unbiased variance estimator:

$$\widehat{\sigma^2}_{unbiased} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \widehat{\mu}_{MLE})^2$$

Maximum Likelihood Estimation

Observe $X_1, X_2, \ldots, X_n$ drawn IID from $f(x;\theta)$ for some "true" $\theta = \theta_*$

Likelihood function $L_n(\theta) = \prod_{i=1}^{n} f(X_i;\theta)$

Log-Likelihood function $l_n(\theta) = \log(L_n(\theta)) = \sum_{i=1}^{n} \log(f(X_i;\theta))$

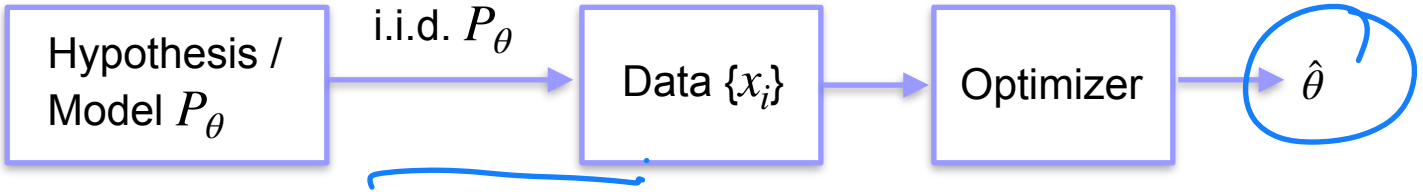
Maximum Likelihood Estimator (MLE) $\widehat{\theta}_{MLE} = \arg\max_{\theta} L_n(\theta)$

Under benign assumptions, as the number of observations $n \to \infty$ we have $\widehat{\theta}_{MLE} \to \theta_*$

The MLE is a "recipe" that begins with a *model* for data $f(x;\theta)$

Recap

- Learning is…
  - Collect some data
    - E.g., coin flips
  - Choose a hypothesis class or model
    - E.g., Bernoulli
  - Choose a loss function
    - E.g., data likelihood
  - Choose an optimization procedure
    - E.g., set derivative to zero to obtain MLE



Applications preview

Maximum Likelihood Estimation

Observe $X_1, X_2, \ldots, X_n$ drawn IID from $f(x;\theta)$ for some "true" $\theta = \theta_*$

Likelihood function $L_n(\theta) = \prod_{i=1}^{n} f(X_i;\theta)$

Log-Likelihood function $l_n(\theta) = \log(L_n(\theta)) = \sum_{i=1}^{n} \log(f(X_i;\theta))$

Maximum Likelihood Estimator (MLE) $\widehat{\theta}_{MLE} = \arg\max_{\theta} L_n(\theta)$

Under benign assumptions, as the number of observations $n \to \infty$ we have $\hat{\theta}_{MLE} \to \theta_*$

Why is it useful to recover the "true" parameters θ_* of a probabilistic model?

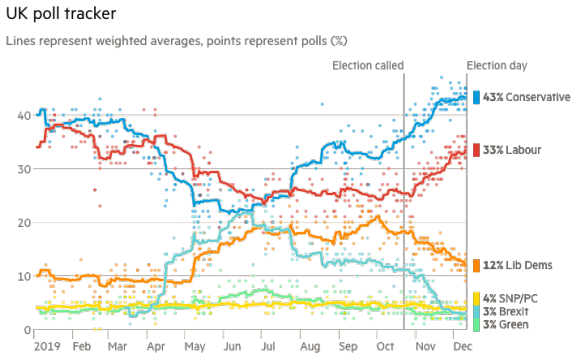
- **Estimation** of the parameters θ_* is the goal
- Help **interpret** or summarize large datasets
- Make **predictions** about future data
- **Generate** new data $X \sim f(\cdot\,; \hat{\theta}_{MLE})$

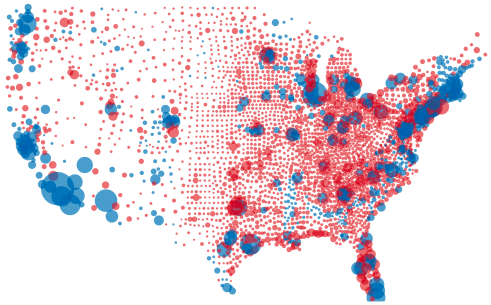
Estimation

Observe $X_1, X_2, \ldots, X_n$ drawn IID from $f(x;\theta)$ for some "true" $\theta = \theta_*$

Opinion polls
How does the greater population feel about an issue? Correct for over-sampling?

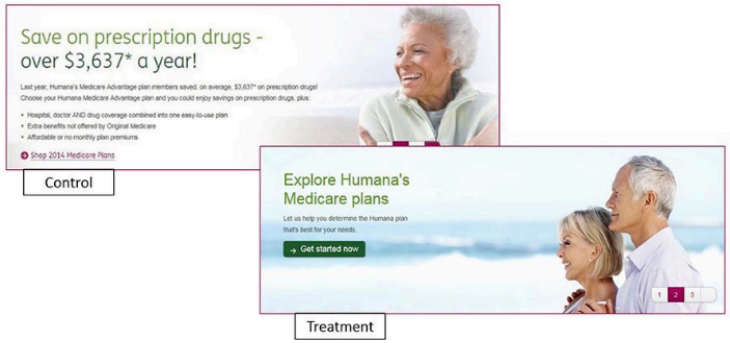
- θ_* is "true" average opinion
- $X_1, X_2, \ldots$ are sample calls





A/B testing
How do we figure out which ad results in more click-through?

- θ_* are the "true" average rates
- $X_1, X_2, \ldots$ are binary "clicks"



Interpret

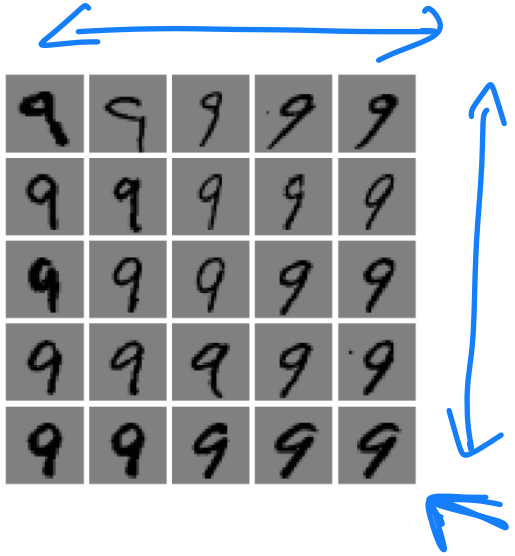
Observe $X_1, X_2, \ldots, X_n$ drawn IID from $f(x;\theta)$ for some "true" $\theta = \theta_*$

Customer segmentation / clustering
Can we identify distinct groups of customers by their behavior?

- θ_* describes "center" of distinct groups
- $X_1, X_2, \ldots$ are individual customers

Data exploration
What are the degrees of freedom of the dataset?

- θ_* describes the principle directions of variation
- $X_1, X_2, \ldots$ are the individual images

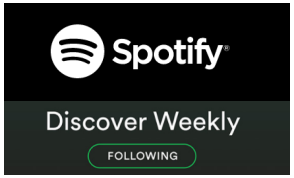


Predict

Observe $X_1, X_2, \ldots, X_n$ drawn IID from $f(x;\theta)$ for some "true" $\theta = \theta_*$

Content recommendation
Can we predict how much someone will like a movie based on past ratings?

- θ_* describes user's preferences
- $X_1, X_2, \ldots$ are (movie, rating) pairs



Object recognition / classification
Identify a flower given just its picture?

- θ_* describes the characteristics of each kind of flower
- $X_1, X_2, \ldots$ are the (image, label) pairs

(a)

(b)

(c)

Figure 1.1: Three types of Iris flowers: Setosa, Versicolor and Virginica. Used with kind permission of Dennis Kramb and SIGNA.

index	sl	sw	pl	pw	label
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
	...				
50	7.0	3.2	4.7	1.4	Versicolor
	...				
149	5.9	3.0	5.1	1.8	Virginica

iris

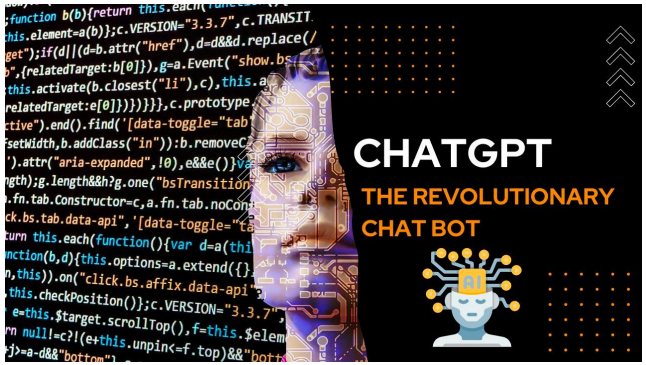
Generate

Observe $X_1, X_2, \ldots, X_n$ drawn IID from $f(x;\theta)$ for some "true" $\theta = \theta_*$

Text generation
Can AI generate text that could have been written like a human?

- θ_* describes language structure
- $X_1, X_2, \ldots$ are text snippets found online

"Kaia the dog wasn't a natural pick to go to mars. No one could have predicted she would..."



https://chat.openai.com/chat

Image to text generation
Can AI generate an image from a prompt?

- θ_* describes the coupled structure of images and text
- $X_1, X_2, \ldots$ are the (image, caption) pairs found online

"dog talking on cell phone under water, oil painting"

https://labs.openai.com/