https://www.cs.washington.edu/education/courses/cse546/18au/

Machine Learning
CSE546

Kevin Jamieson
University of Washington

September 27, 2018

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?

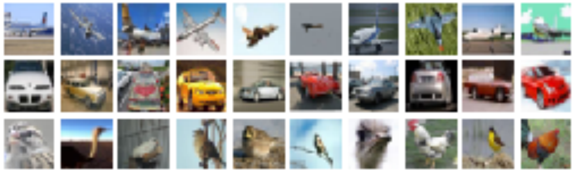
```
cats = []
dogs = []
other = []
for tweet in tweets:
    if "cat" in tweet:
        cats.append(tweet)
    elseif "dog" in tweet:
        dogs.append(tweet)
    else:
        other.append(tweet)
return cats, dogs, other
```

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

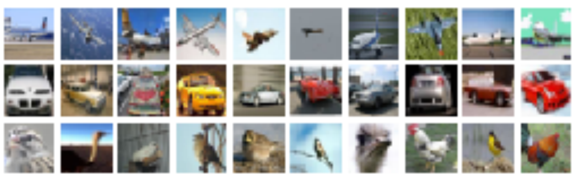
Write a program that sorts images into those containing "**birds**", "**airplanes**", or ***other.***



airplane
other
bird

Machine learning algorithms

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

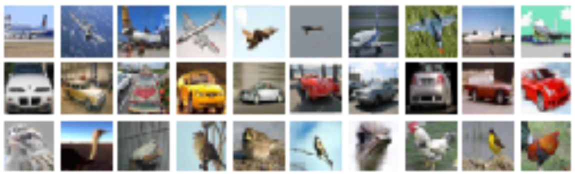


airplane
other
bird

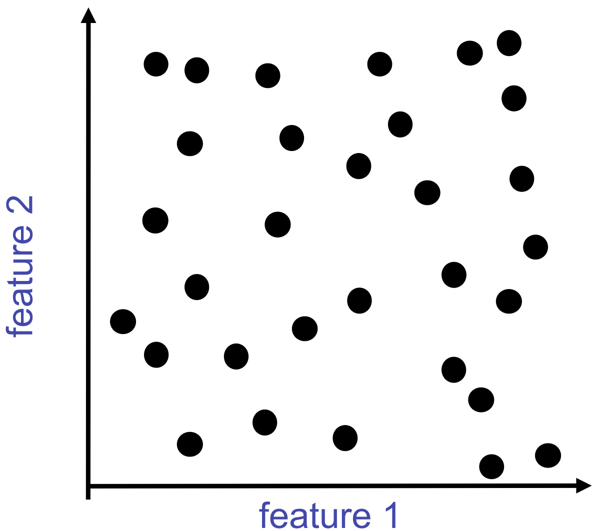
```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elseif plane in image:
        planes.append(image)
    else:
        other.append(tweet)
return birds, planes, other
```

Machine learning algorithms

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



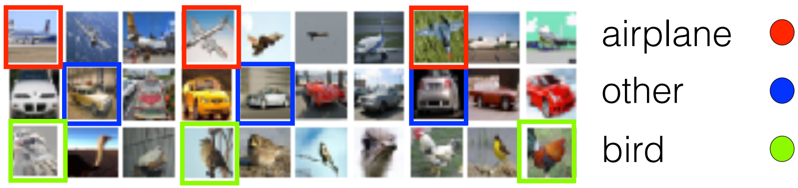
airplane
other
bird

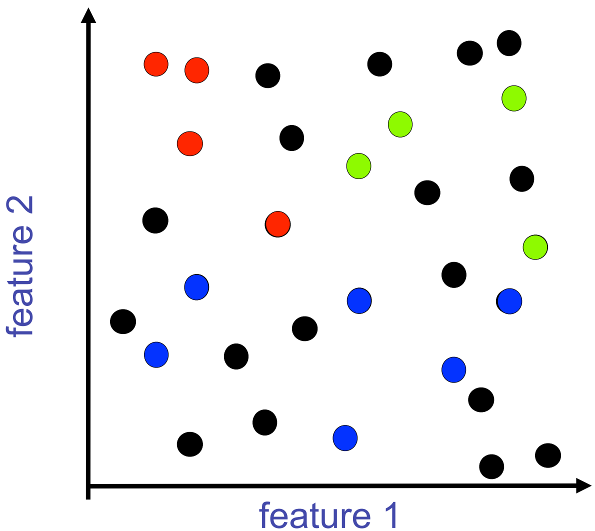



```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elseif plane in image:
        planes.append(image)
    else:
        other.append(tweet)
return birds, planes, other
```

Machine learning algorithms

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

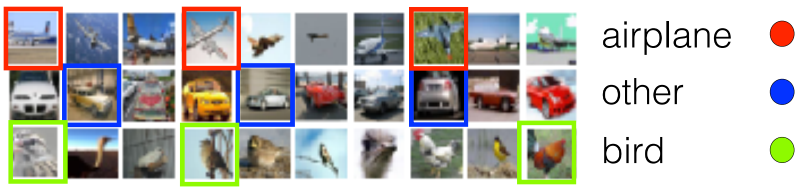


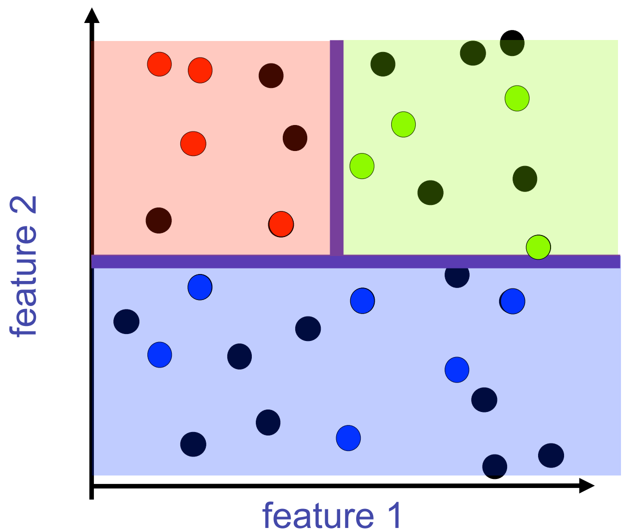



```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elseif plane in image:
        planes.append(image)
    else:
        other.append(tweet)
return birds, planes, other
```

Machine learning algorithms

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

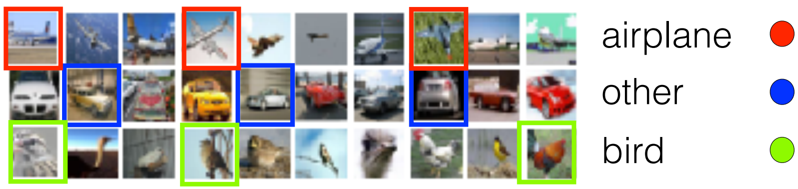


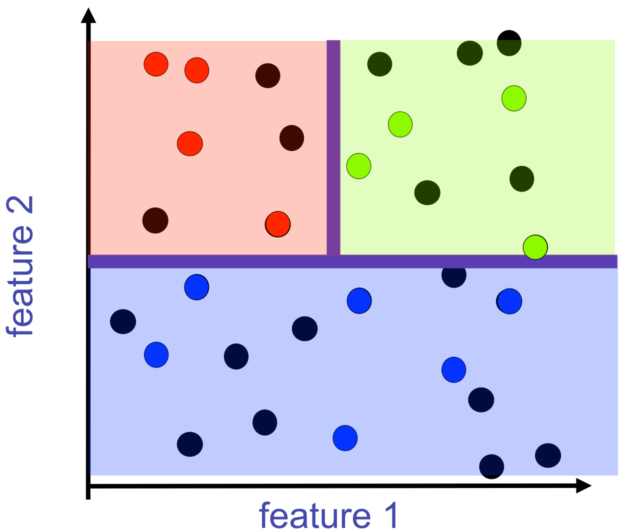



```
birds = []
planes = []
other = []
for image in images:
    if bird in image:
        birds.append(image)
    elseif plane in image:
        planes.append(image)
    else:
        other.append(tweet)
return birds, planes, other
```

Machine learning algorithms

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






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

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

NETFLIX

98% Match 2017 TV

Spotify

Discover Weekly

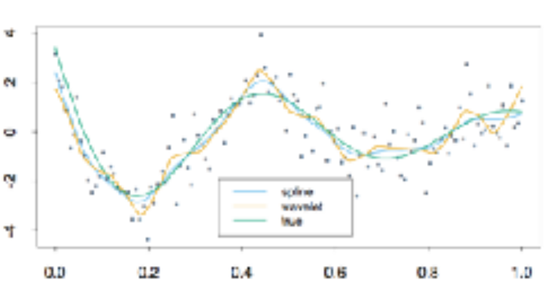
FOLLOWING

amazon prime

You may also like…

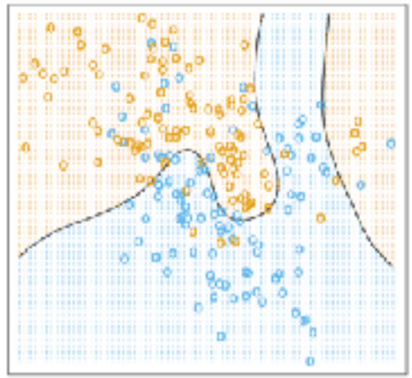
ML uses past data to make personalized predictions

Flavors of ML



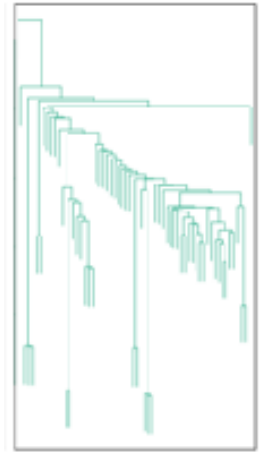
Regression

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

Mix of statistics (theory) and algorithms (programming)

CSE546: Machine Learning

Lecture: Tuesday, Thursday 11:30-12:50 Room: KNE 220

Instructor: Kevin Jamieson

Contact: cse546-instructors@cs.washington.edu

Website: https://courses.cs.washington.edu/courses/cse546/18au/

What this class is:

- **Fundamentals of ML:** bias/variance tradeoff, overfitting, parametric models (e.g. linear), non-parametric models (e.g. kNN, trees), optimization and computational tradeoffs, unsupervised models (e.g. PCA), reinforcement learning
- **Preparation for learning:** the field is fast-moving, you will be able to apply the basics and teach yourself the latest
- **Homeworks and project:** use your research project for the class

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

- Formally:
  - CSE 312, STAT 341, STAT 391 or equivalent
- Topics
  - Linear algebra
    - eigenvalues, orthogonal matrices, quadratic forms
  - Multivariate calculus
  - Probability and statistics
    - Distributions, densities, marginalization, moments
  - Algorithms
    - Basic data structures, complexity
- “Can I learn these topics concurrently?”
- Use HW0 and Optional Review to judge skills (more in a sec)
- **See website for review materials!**

Grading

- 5 homeworks (65%)
  - Each contains both theoretical questions and will have programming
  - Collaboration okay. You must write, submit, and understand your answers and code (which we may run)
  - Do not Google for answers.
- Final project (35%)
  - An ML project of your choice that uses real data

1. All code must be written in Python
2. All written work must be typeset using LaTeX

See course website for tutorials and references.

Homeworks

- HW 0 is out (10 points, **Due next Thursday Midnight**)
  - Short *review*, gets you using Python and LaTeX
  - Work individually, treat as barometer for readiness
- HW 1,2,3,4 (25 points each)
  - They are not easy or short. Start early.
- Grade is minimum of the summed points and 100 points.
- **There is no credit for late work, receives 0 points.**
- **You must turn in all 5 assignments (even if late for 0 points) or else you will not pass.**

Projects (35%)

- An opportunity/intro for research in machine learning
- Grading:
  - We seek some novel exploration.
  - If you write your own solvers, great. We takes this into account for grading.
  - You may use ML toolkits (e.g. TensorFlow, etc), but we expect more ambitious project (in terms of scope, data, etc).
  - If you use simpler/smaller datasets, then we expect a more involved analysis.
- Individually or groups of two or three.
  - If in a group, the expectations are higher
- Must involve real data
  - Must be data that you have available to you by the time of the project proposals
- It's encouraged to be related to your research, but must be something new you did this quarter
  - Not a project you worked on during the summer, last year, etc.
  - You also must have the data right now.

Optional Review

- Little rusty on linear algebra and probability?
- We will have a *review* to remind you of topics you once knew well. **This is not a bootcamp.**
- **Monday evening? See Mattermost for finding a date…**

Communication Channels

- **Mattermost** (secure, open-source Slack clone)
  - Announcements (office hour changes, due dates, etc.)
  - Questions (logistical or homework) - please participate and help others
  - All non-personal questions should go here

- E-mail instructors about personal issues and grading:
  - cse546-instructors@cs.washington.edu
- Office hours limited to knowledge based questions. Use email for all grading questions.

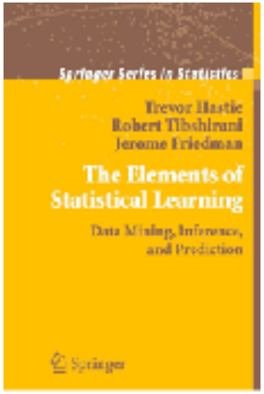
Staff

- Six Great TAs, lots of office hours (subject to change)
  - TA, **Jifan Zhang (jifan@uw)**, *Monday* 3:30-4:30 PM, CSE 4th floor breakout
  - TA, **An-Tsu Chen (atc22@uw)**, *Wednesday* 4:00-5:00 PM, CSE 220
  - TA, **Pascal Sturmfels (psturm@uw)**, *Wednesday* 9:00AM-10:00 AM, CSE 220
  - TA, **Beibin Li (beibin@uw)**, *Wednesday* 1:30-2:30 PM, CSE 220
  - TA, **Alon Milchgrub (alonmil@uw)**, *Thursday* 10:00-11:00AM, CSE 220
  - TA, **Kung-Hung (Henry) Lu (henrylu@uw)**, *Friday* 12:30-1:30 PM, CSE 007
  - Instructor, *Tuesday* 4:00-5:00 PM, CSE 666

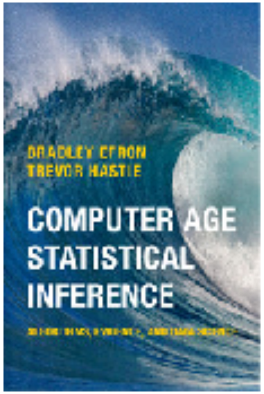
- Check website and Mattermost for changes and exceptions

Text Books

- Textbook (both ***free*** PDF):
  - ***The Elements of Statistical Learning: Data Mining, Inference, and Prediction;* Trevor Hastie, Robert Tibshirani, Jerome Friedman**

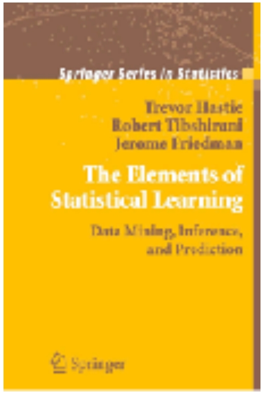


  - *Computer Age Statistical Inference: Algorithms, Evidence and Data Science,* Bradley Efron, Trevor Hastie



Text Books

- Textbook (both ***free*** PDF):
  - ***The Elements of Statistical Learning: Data Mining, Inference, and Prediction;* Trevor Hastie, Robert Tibshirani, Jerome Friedman**



  - *Computer Age Statistical Inference: Algorithms*

- *Not free, but more useful for this class*
  - ***All of Statistics,*** Larry Wasserman

Enjoy!

- ML is becoming ubiquitous in science, engineering and beyond
- It’s one of the hottest topics in industry today
- This class should give you the basic foundation for applying ML and developing new methods
- The fun begins…

Maximum Likelihood Estimation

Machine Learning – CSE546
Kevin Jamieson
University of Washington

September 27, 2018

Your first consulting job

- *Billionaire*: I have special coin, if I flip it, what's the probability it will be heads?
- *You*: Please flip it a few times: 5 times

HTHHT →

- *You*: The probability is: 3/5

- *Billionaire:* Why?

Coin – Binomial 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 Binomial distribution

$\mathbb{P}(X_2=H|X_1=T) = \mathbb{P}(X_2=H)$

$\mathbb{P}(A|B) = \mathbb{P}(A)$

$\mathbb{P}(HHTHT) = \mathbb{P}(H)\mathbb{P}(H)\mathbb{P}(T)\mathbb{P}(H)\mathbb{P}(T)$
$= \theta \cdot \theta \cdot (1-\theta)\theta(1-\theta)$
$= \theta^k (1-\theta)^{n-k}$

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

Maximum Likelihood Estimation

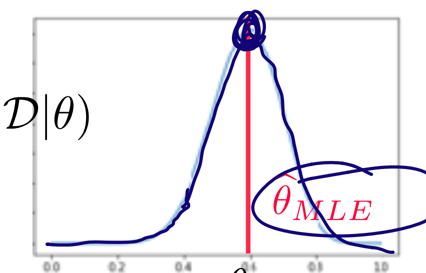
- **Data**: sequence *D= (HHTHT…)*, **k heads** out of **n flips**
- **Hypothesis:** P(Heads) = θ, P(Tails) = 1-θ

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

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

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

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



Your first learning algorithm

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

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

$$= k \log(\theta) + (n-k)\log(1-\theta)$$

- Set derivative to zero: $\boxed{\frac{d}{d\theta} \log P(\mathcal{D}|\theta) = 0}$

$$\frac{\partial}{\partial\theta}\left(k\log\theta + (n-k)\log(1-\theta)\right) = \frac{k}{\theta} + \frac{n-k}{1-\theta}(-1) = \frac{k(1-\theta)-(n-k)\theta}{\theta(1-\theta)}$$

$$= 0$$

$$k(1-\theta) = (n-k)\theta$$

$$k = n\theta \implies \boxed{\theta = \frac{k}{n}}$$

How many flips do I need?

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

- *You*: flip the coin 5 times. *Billionaire*: I got 3 heads.

$$\widehat{\theta}_{MLE} = 3/5$$

- *You*: flip the coin 50 times. *Billionaire*: I got 20 heads.

$$\widehat{\theta}_{MLE} = \frac{20}{50} = 2/5$$

- *Billionaire:* Which one is right? Why?

Simple bound (based on Hoeffding's inequality)

- For **n flips** and **k heads** the MLE is **unbiased** for true θ^*:

$$\widehat{\theta}_{MLE} = \frac{k}{n} \qquad \mathbb{E}[\widehat{\theta}_{MLE}] = \theta^*$$

- Hoeffding's inequality says that for any $\varepsilon>0$:

$$P(|\widehat{\theta}_{MLE} - \theta^*| \geq \epsilon) \leq 2e^{-2n\epsilon^2} = \delta$$

$$\Rightarrow \quad \varepsilon = \sqrt{\frac{\log(2/\delta)}{2n}}$$

$\Rightarrow$ With probability $\geq 1-\delta$, $|\widehat{\theta}_{MLE} - \theta_*| \leq \sqrt{\frac{\log(2/\delta)}{2n}}$

PAC Learning

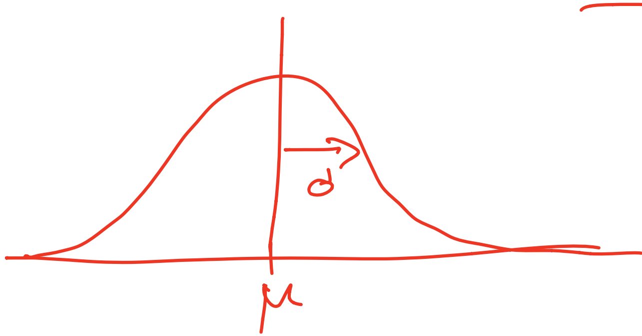
- PAC: Probably Approximate Correct
- *Billionaire*: I want to know the parameter θ^*, within $\varepsilon = 0.1$, with probability at least $1-\delta = 0.95$. How many flips?

$$P(|\widehat{\theta}_{MLE} - \theta^*| \geq \epsilon) \leq 2e^{-2n\epsilon^2}$$

What about continuous variables?

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

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



Some properties of Gaussians

- affine transformation (multiplying by scalar and adding a constant)
  - $X \sim N(\mu,\sigma^2)$
  - $Y = aX + b \quad \rightarrow \quad Y \sim N(a\mu+b,a^2\sigma^2)$

$\mathbb{E}[Y] = a\,\mathbb{E}[X] + b = a\mu + b$

- 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., exam scores):

$$
\begin{aligned}
P(\mathcal{D}|\mu,\sigma) &= P(x_1,\ldots,x_n|\mu,\sigma) \\
&= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \prod_{i=1}^n e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}
\end{aligned}
$$

- Log-likelihood of data:

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

Your second learning algorithm: MLE for mean of a Gaussian

- What's MLE for mean?

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

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

$$\sum_{i=1}^{n}(x_i-\mu) = 0 \longrightarrow \sum_{i=1}^{n}x_i = \sum \mu = n\mu$$

$$\Rightarrow \mu = \frac{1}{n}\sum_{i=1}^{n}x_i$$

MLE for variance

$\log(ab) = \log(a) + \log(b)$

- Again, set derivative to zero:

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

$$= -n\frac{1}{\sigma} + \sum_{i=1}^{n} \frac{(x_i-\mu)^2}{\sigma^3} = -n + \sum \frac{(x_i-\mu)^2}{\sigma^2} = 0$$

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

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

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 $X_1, X_2, \dots, X_n$ 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) $\widehat{\theta}_{MLE} = \arg\max_\theta L_n(\theta)$

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

Properties (under benign regularity conditions—smoothness, identifiability, etc.):

- Asymptotically consistent and normal: $\frac{\widehat{\theta}_{MLE} - \theta_*}{\widehat{se}} \sim \mathcal{N}(0,1)$
- Asymptotic Optimality, minimum variance (see Cramer-Rao lower bound)

Recap

- Learning is…
  - Collect some data
    - E.g., coin flips
  - Choose a hypothesis class or model
    - E.g., binomial
  - Choose a loss function
    - E.g., data likelihood
  - Choose an optimization procedure
    - E.g., set derivative to zero to obtain MLE
  - Justifying the accuracy of the estimate
    - E.g., Hoeffding’s inequality

House i attributes = {sq. feet, dist. to lake, ...}

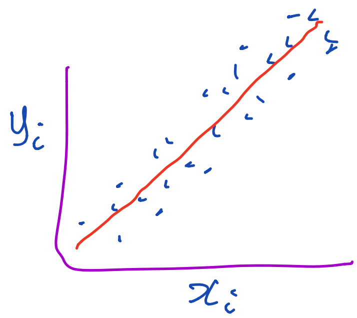
We know $x_i \in \mathbb{R}^d$ (d known attributes)

Observe final sale price $y_i \in \mathbb{R}$

Last month observations $\{(x_i, y_i)\}_{i=1}^n$

Now we see houses $\{x_j\}_{j=1}^m$

Linear model $\cancel{\hat{y}_i = x_i^T\theta}$ for some $\theta \in \mathbb{R}^d$.



$$\boxed{y_i = x_i^T\theta + \varepsilon_i,} \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$

$$f(y \mid x_i, \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - x_i^T\theta)^2}{2\sigma^2}\right)$$

$$L_n(\theta) = (2\pi\sigma^2)^{-n/2} \prod_{i=1}^n \exp\left(-\frac{(y - x_i^T\theta)^2}{2\sigma^2}\right)$$

$$= (2\pi\sigma^2)^{-n/2} \exp\left(\sum -\frac{(y - x_i^T\theta)^2}{2\sigma^2}\right)$$

$$\ell_n(\theta) = -\frac{n}{2}\log(2\pi\sigma^2) - \sum_{i=1}^n \frac{(y_i - x_i^T\theta)^2}{2\sigma^2}$$

$$\nabla_\theta \ell_n(\theta) = \sum_{i=1}^n \frac{(y_i - x_i^T\theta)x_i}{\sigma^2} = 0$$

$$\sum_{i=1}^n x_i y_i = \sum_{i=1}^n x_i x_i^T \theta = \left(\sum_{i=1}^n x_i x_i^T\right)\theta$$

$$\hat{\theta}_{MLE} = \left(\sum_{i=1}^n x_i x_i^T\right)^{-1}\left(\sum x_i y_i\right)$$

$$X = (x_1, \dots, x_n)^T = \begin{bmatrix} - & x_1^T & - \\ & \vdots & \\ - & x_n^T & - \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$\boxed{\underset{n\times 1}{Y} = \underset{n\times d}{X}\,\underset{d\times 1}{\theta_*} + \underset{n\times 1}{\varepsilon}} \qquad \|w\|_2^2 := \sum_{i=1}^{n} |w_i|^2$$

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmin}} \sum (y_i - x_i^T\theta)^2 = \underset{\theta}{\operatorname{argmin}} \|\underbrace{y - X\theta}\|_2^2$$

$$\rightarrow \nabla_\theta \|y - X\theta\|_2^2 = -X^T \cdot 2(y - X\theta) = 0$$

$$X^T y = X^T X \theta \rightarrow \hat{\theta}_{MLE} = (X^T X)^{-1} X^T y$$

$$\begin{aligned}
\hat{\theta}_{MLE} &= (X^T X)^{-1} X^T y \\
&= (X^T X)^{-1} X^T (X\theta_* + \varepsilon) \\
&= \underline{(X^T X)^{-1}}\, \underline{X^T} X \theta_* + (X^T X)^{-1} X^T \varepsilon \\
&= \theta_* \underline{+ (X^T} X)^{-1} X^T \varepsilon
\end{aligned}$$

$$\underline{\mathbb{E}[\hat{\theta}}_{MLE}] = \theta_* + (X^T X)^{-1} X^T \mathbb{E}[\varepsilon] = \theta_*$$

$$\mathbb{E}\left[\left(\hat{\theta}_{MLE} - \mathbb{E}[\hat{\theta}_{MLE}]\right)\left(\hat{\theta}_{MLE} - \mathbb{E}[\hat{\theta}_{MLE}]\right)^T\right]$$

$$= \mathbb{E}\left[(X^TX)^{-1} X^T \varepsilon \varepsilon^T X (X^TX)^{-1}\right]$$

$$= (X^TX)^{-1} X^T \underbrace{\mathbb{E}[\varepsilon \varepsilon^T]}_{I\sigma^2} X (X^TX)^{-1}$$

$$= \sigma^2 (X^TX)^{-1}$$

$$\hat{\theta}_{MLE} = \mathcal{N}\left(\theta_*, (X^TX)^{-1}\sigma^2\right)$$