CS 540 Introduction to Artificial Intelligence
Midterm Review

University of Wisconsin-Madison
Spring 2026 Sections 1 & 2

Midterm Information

- **Time: March 24th 5:45-7:15 PM**
- **Location (by section **):**
    - Section 001 (Tuesday/Thursday 11-12:15PM): 6210 Social Sciences Bldg
    - Section002 (Tuesday/Thursday 2:30-3:45PM): B10 Ingraham Hall
    - Students with McBurney accommodations should have received an email with additional information.
    - Students who cannot take the exam on the specified time should contact their instructor if they have not done it yet.
- **Topics: Topics covered up to and including Week 9**
- **Exclusion List (questions regarding the following topics will NOT appear on the midterm):**
    - **Logic (covered in sections 1 and 2)**
    - **SVM + Kernel Trick (covered in section 3)**
- **Format**: MCQ
- **Cheat sheet**: a handwritten single piece of paper, front and back
- **Calculator**: optional, if it doesn't have an Internet connection
- **Bring**: your WISC ID, pencil (No 2 or softer), your 1-sheet notes.
- **Past exam questions**: on Canvas →Files → Past Exams

Bayesian **Inference**

- Terminology:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}$$

- *H* is the hypothesis
- *E* is the evidence

Bayesian **Inference**

- Terminology:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)} \longleftarrow \textbf{Prior}$$

- Prior: estimate of the probability **without** evidence

Bayesian Inference

- Terminology:

Likelihood

$$P(H|E) = \frac{\underset{\uparrow}{P(E|H)}P(H)}{P(E)}$$

- Likelihood: probability of evidence **given a hypothesis**

Bayesian Inference

- Terminology:

$$\underset{\substack{\uparrow \\ \textbf{Posterior}}}{P(H|E)} = \frac{P(E|H)P(H)}{P(E)}$$

- Posterior: probability of hypothesis **given evidence**.

Naïve Bayes

- Conditional Probability & Bayes:

$$P(H|E_1, E_2, \ldots, E_n) = \frac{P(E_1, \ldots, E_n|H)P(H)}{P(E_1, E_2, \ldots, E_n)}$$

- If we further make the **conditional independence assumption (a.k.a. Naïve Bayes)**

$$P(H|E_1, E_2, \ldots, E_n) = \frac{P(E_1|H)P(E_2|H) \ \cdots \ P(E_n|H)P(H)}{P(E_1, E_2, \ldots, E_n)}$$

Break & Quiz

Q: You are using Bayesian inference to determine if a six-sided die is fair or biased. You know the die is either fair (each side has a probability of 1/6) or biased to favor the number 6 (the probability of rolling a 6 is 1/2, and all other numbers have a probability of 1/10). The prior probability for the die being fair is equal to the prior for it being biased, P (fair) = P (biased) = 0.5. You roll the die twice and observe the sequence "6,6". What is the posterior probability P (fair|"6, 6")?

A. 1/36
B. 1/9
C. 1/10
D. 1/4
E. 9/10

Break & Quiz

Q: You are using Bayesian inference to determine if a six-sided die is fair or biased. You know the die is either fair (each side has a probability of 1/6) or biased to favor the number 6 (the probability of rolling a 6 is 1/2, and all other numbers have a probability of 1/10). The prior probability for the die being fair is equal to the prior for it being biased, P (fair) = P (biased) = 0.5. You roll the die twice and observe the sequence "6,6". What is the posterior probability P (fair|"6, 6")?

A. 1/36
B. 1/9
C. 1/10
D. 1/4
E. 9/10

$$P(fair|"6,6") = \frac{P("6,6"|fair)P(fair)}{P("6,6")}$$

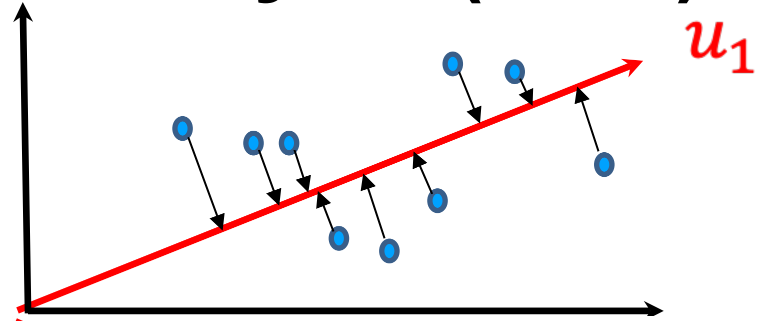
$$= \frac{P("6"|fair)P("6"|fair)P(fair)}{P("6,"|fair)P("6"|fair)P(fair) + P("6,"|biased)P("6,"|biased)P(biased)}$$

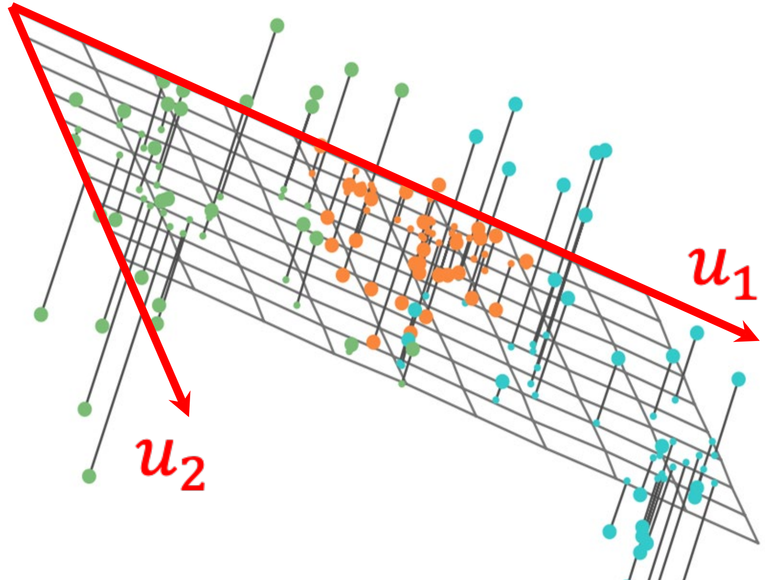
$$= \frac{\frac{1}{6}\times\frac{1}{6}\times\frac{1}{2}}{\frac{1}{6}\times\frac{1}{6}\times\frac{1}{2}+\frac{1}{2}\times\frac{1}{2}\times\frac{1}{2}} = \frac{1}{10}$$

PCA

Principal Components Analysis (PCA)

- Find axes $u_1, u_2, \ldots, u_n \in \mathbb{R}^d$ of a subspace
  - Will project to this subspace
- These vectors are the **principal components**
- Want to preserve data
  - Minimize projection error



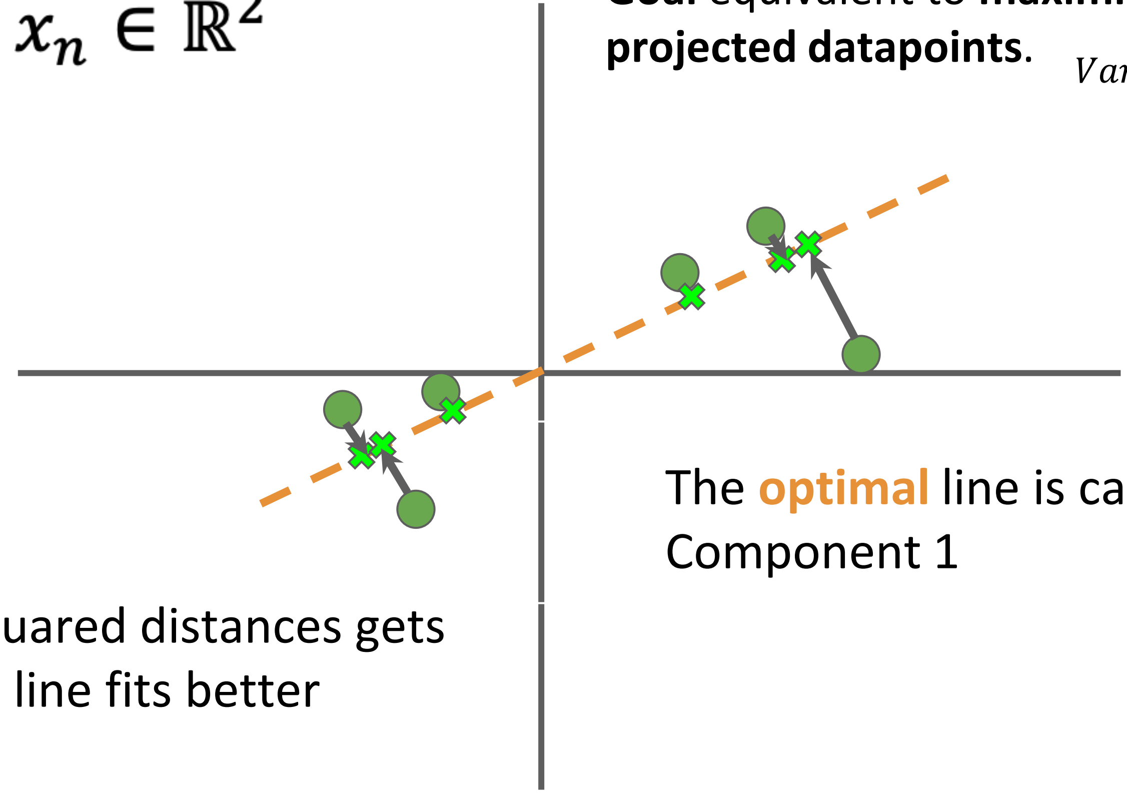


Projection: An Example

$x_1, x_2, \ldots, x_n \in \mathbb{R}^2$

Goal equivalent to **maximizing the variance of projected datapoints**.

$$Variance = \frac{1}{n-1}\sum_{i=1}^{n}\langle u, x_i\rangle^2$$



The sum of squared distances gets smaller as the line fits better

The **optimal** line is called Principal Component 1

PCA Procedure

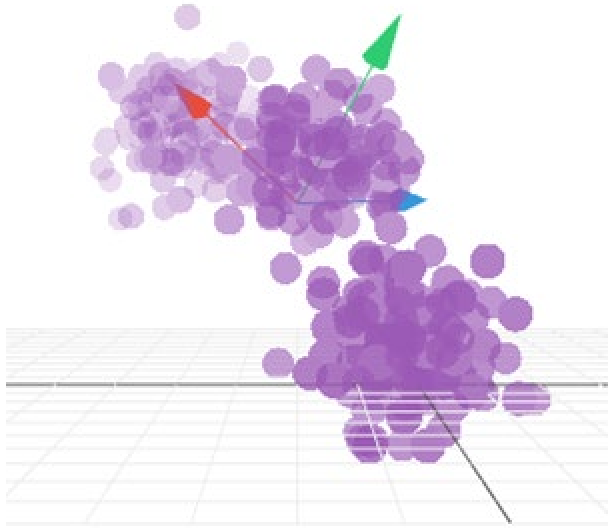
Input: data $x_1, x_2, \dots \dots x_n \in \mathbb{R}^d$

Step 1: Center data at the origin so that $\hat{\mu} = 0$

Step 1.1: Calculate the mean $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Step 1.2: Subtract the mean from the datapoints

$$x_1 = x_1 - \hat{\mu}$$
$$x_2 = x_2 - \hat{\mu}$$
....
$$x_n = x_n - \hat{\mu}$$



Victor Powell

PCA Procedure

Output: principal components $u_1, u_2, \ldots \ldots u_m \in \mathbb{R}^d$

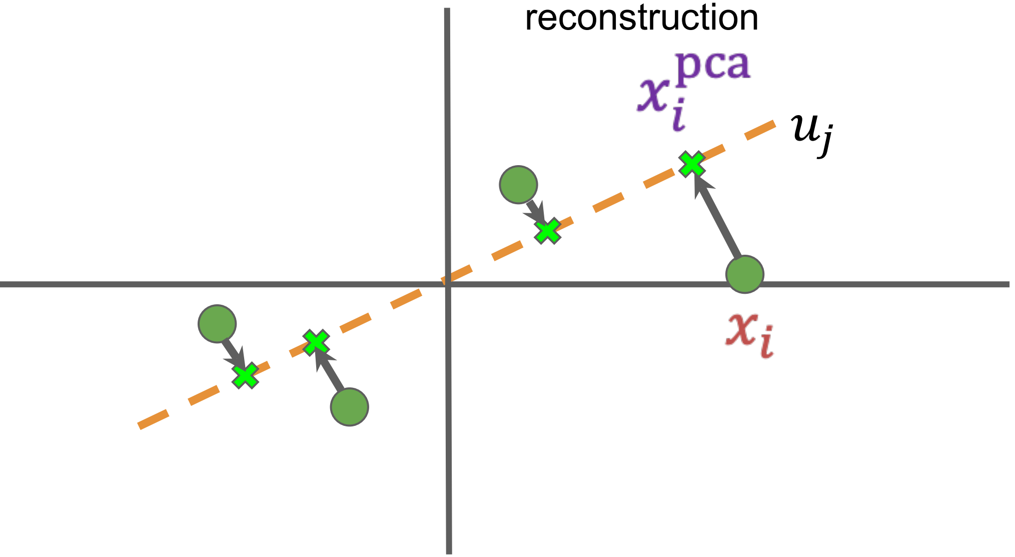
The $\boldsymbol{m}$ **principal components** are:

- orthogonal
- the **top-$\boldsymbol{m}$ eigenvectors** of the covariance matrix $(S = \frac{1}{n-1} \sum_{i=1}^{n} x_i x_i^T)$

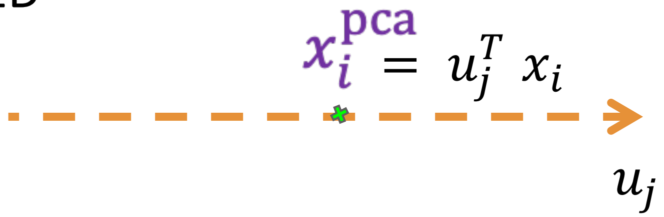
Step 2: Compute **covariance matrix**
Step 3: Compute **eigenvectors** and **eigenvalues** of covariance matrix
Step 4: **Sort eigenvalues** from high to low
Step 5: Select **top-$\boldsymbol{m}$ eigenvectors**

PCA Projection and Reconstruction

For each x_i: $x_i^{pca} = \underbrace{\sum_{j=1}^{m} \overbrace{(u_j^T x_i)}^{\text{projection}} u_j}_{\text{reconstruction}}$



2D → 1D



Break & Quiz

Suppose we perform PCA to compress a dataset of points in $\mathbb{R}^4$ into a dataset of points in $\mathbb{R}^2$. We find that the first principal component is $\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ and the second principal component is $\begin{pmatrix} \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \\ 0 \\ -\frac{1}{\sqrt{3}} \end{pmatrix}$. If $x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ is a point in the original dataset, what is the compressed version $\bar{x} \in \mathbb{R}^2$

A. $\begin{pmatrix} 1 \\ -\frac{1}{\sqrt{3}} \end{pmatrix}$ B. $\begin{pmatrix} 1 \\ \frac{1}{\sqrt{3}} \end{pmatrix}$ C. $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ D. $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Break & Quiz

Suppose we perform PCA to compress a dataset of points $\mathbb{R}^4$ into a dataset of points in $\mathbb{R}^2$. We find that the first principal component is $\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$ and the second principal component is $\begin{pmatrix} \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \\ 0 \\ -\frac{1}{\sqrt{3}} \end{pmatrix}$. If $x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ is a point in the original dataset, what is the compressed version $\bar{x} \in \mathbb{R}^2$

The compressed representation is $\begin{pmatrix} u_1^\top x \\ u_2^\top x \end{pmatrix}$

A. $\begin{pmatrix} 1 \\ -\frac{1}{\sqrt{3}} \end{pmatrix}$ B. $\begin{pmatrix} 1 \\ \frac{1}{\sqrt{3}} \end{pmatrix}$ C. $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ D. $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

NLP

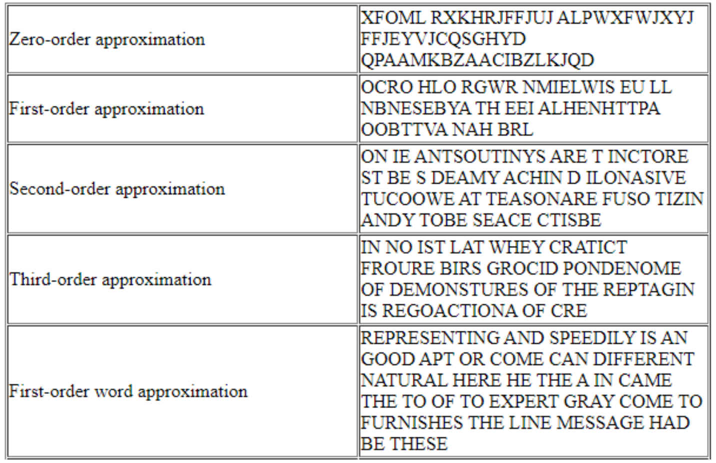
Language Models

Basic idea: use probabilistic models to **assign a probability to a sentence W**

$$P(W) = P(w_1, w_2, \ldots, w_n) \text{ or } P(w_{\text{next}} | w_1, w_2 \ldots)$$

Goes back to Shannon

- Information theory: letters



Zero-order approximation	XFOML RXKHRJFFJUJ ALPWXFWJXYJ FFJEYVJCQSGHYD QPAAMKBZAACIBZLKJQD
First-order approximation	OCRO HLO RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL
Second-order approximation	ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE
Third-order approximation	IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE
First-order word approximation	REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE

Training: Make Assumptions

- Markov assumption with shorter history:

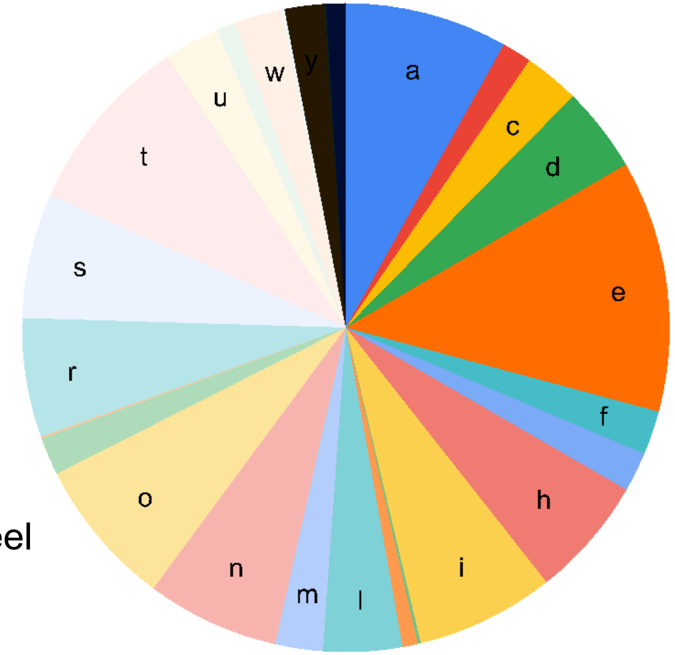
$$P(w_i|w_{i-1}w_{i-2}\ldots w_1) = P(w_i|w_{i-1}w_{i-2}\ldots w_{i-k})$$

- Present doesn't depend on whole past
  - Just recent past, i.e., *context*.
  - What's ***k=0?***

k=0: Unigram Model

- Full independence assumption:
  - (Present doesn't depend on the past)

$$P(w_1, w_2, \ldots, w_n) = P(w_1)P(w_2)\ldots P(w_n)$$

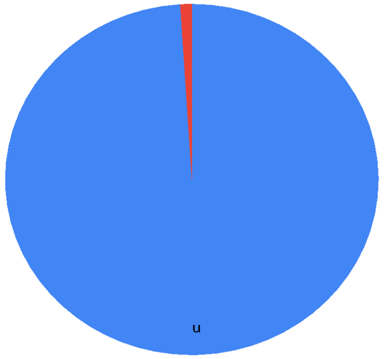


The English letter frequency wheel

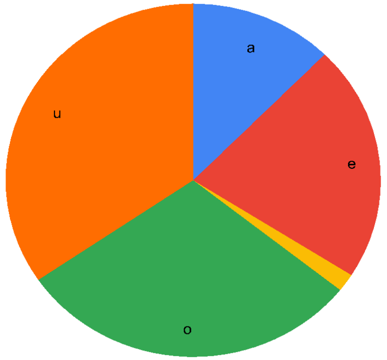
k=1: Bigram Model

- Markov Assumption:
  - (Present depends on immediate past)

$$P(w_1, w_2, \ldots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_2)\ldots P(w_n|w_{n-1})$$



$p(.|q)$: the “after q” wheel



$p(.|j)$: the “after j” wheel

```
texaco, rose, one, in, this, issue,
is, pursuing, growth, in, a, boiler,
house, said, mr., gurria, mexico, 's,
motion, control, proposal, without,
permission, from, five, hundred,
fifty, five, yen outside, new, car,
parking, lot, of, the, agreement,
reached this, would, be, a, record,
november
```

k=n-1: n-gram Model

Can do trigrams, 4-grams, and so on

- More expressive as *n* goes up
- Harder to estimate

Training: just count? I.e, for bigram:

$$P(w_i|w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$

n-gram Training

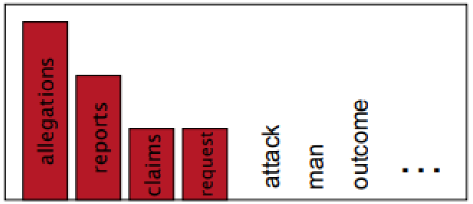
$$P(w_i|w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$

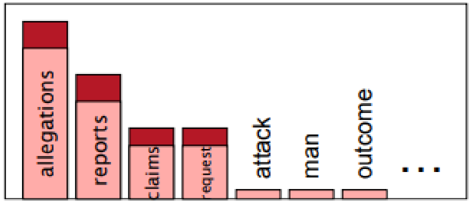
Issues:

- **1**. Multiply tiny numbers?
  - **Solution**: use logs; add instead of multiply
- **2.** n-grams with zero probability?
  - **Solution**: (Laplace) smoothing

$$P(w_i|w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i) + 1}{\text{count}(w_{i-1}) + V}$$

$P(w|\text{denied the})$





Dan Klein

Break & Quiz

Suppose we have the following training corpus in an NLP setup.

"I took my dog out for a walk but my dog ran away"

(i) What is the P(dog) using a unigram model without Laplace smoothing?
(ii) What is the P(my dog) using a unigram model without Laplace smoothing?
(iii) What is the P(my dog) using a bigram model without Laplace smoothing?
(iv) What is the P(dog | my) for a bigram model without Laplace smoothing

Break & Quiz

Suppose we have the following training corpus in an NLP setup.

"I took my dog out for a walk but my dog ran away"

(i) What is the P(dog) using a unigram model without Laplace smoothing?

(ii) What is the P(my dog) using a unigram model without Laplace smoothing?

(iii) What is the P(my dog) using a bigram model without Laplace smoothing?

(iv) What is the P(dog | my) for a bigram model without Laplace smoothing?

(i) P(dog) = 2 / 13 (note there are 13 words in the sentence; two are 'dog').

(ii) P(my dog) = P(my) $*$ P(dog) = (2/13) $*$ (2/13) = 4/169

(iii) P(my dog) = 2 / 12 = 1 / 6

(iv) P(dog | my) = count(my,dog) / count(my) = 2/2 = 1

Machine Learning

Supervised/Unsupervised Learning

Supervised learning:

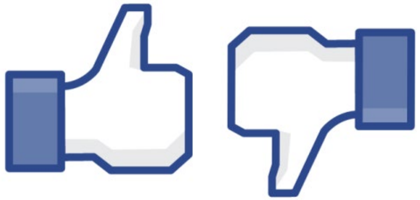
Make predictions, classify data, perform regression

Dataset: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

Features / Covariates / Input

Labels / Outputs

Goal: find function $f : X \rightarrow Y$ to predict label on **new** data



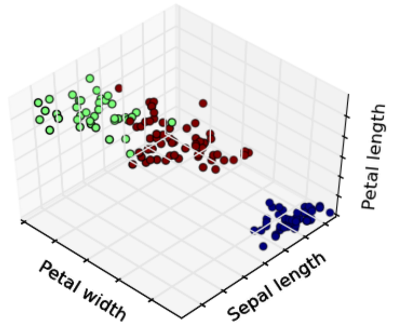
Supervised/Unsupervised Learning

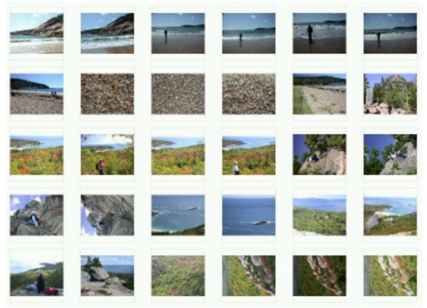
Unsupervised learning:

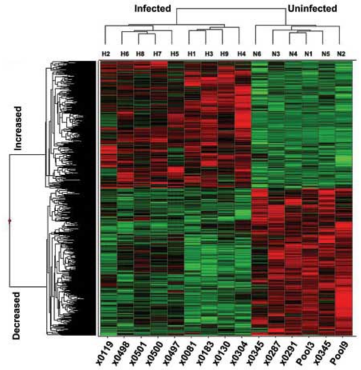
No labels; generally, won't be making predictions

Dataset: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$

Goal: find patterns & structures that help better understand data.





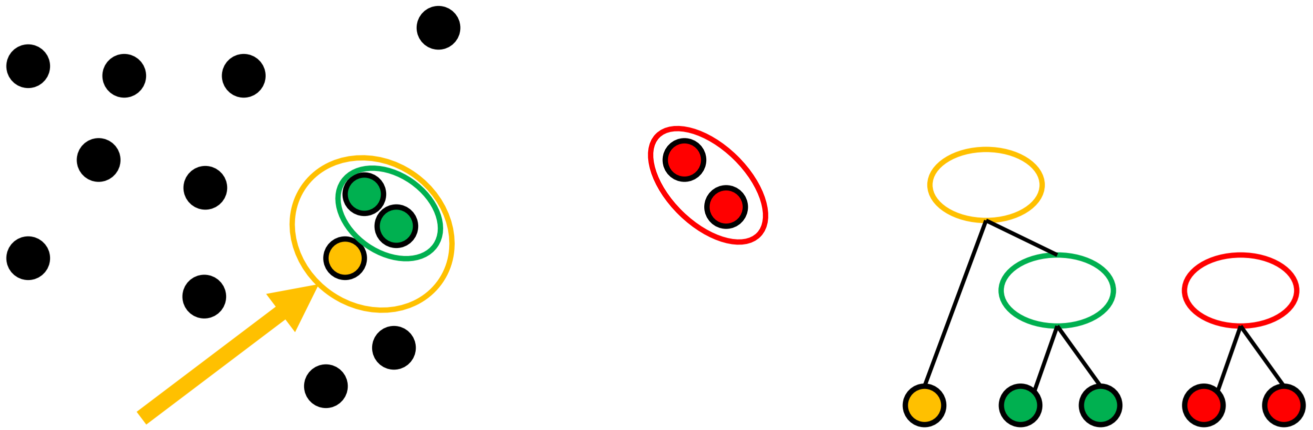


Mulvey and Gingold

Unsupervised Learning

Agglomerative Clustering Example

Repeat: Get pair of clusters that are closest and merge



Merging Criteria

Merge: use closest clusters. Define closest?

Single-linkage

$$d(A,B) = \min_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$

Complete-linkage

$$d(A,B) = \max_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$

Average-linkage

$$d(A,B) = \frac{1}{|A||B|} \sum_{x_1 \in A, x_2 \in B} d(x_1, x_2)$$

Break & Quiz

Q: We want to perform hierarchical clustering with two clusters for the following points: 0,1, 10, 18, 20, 25. What are the clusters if we use single-linkage, complete-linkage, and average-linkage?

A. Single linkage: (0, 1, 10), (18, 20, 25)
Complete linkage: (0, 1), (10, 18, 20, 25)
Average linkage: (0, 1, 10), (18, 20, 25)

B. Single linkage: (0, 1, 10, 18, 20), (25)
Complete linkage: (0), (1, 10, 18, 20, 25)
Average linkage: (0), (1, 10, 18, 20, 25)

C. Single linkage: (0, 1), (10, 18, 20, 25)
Complete linkage: (0, 1), (10, 18, 20, 25)
Average linkage: (0, 1), (10, 18, 20, 25)

D. Single linkage: (0, 1), (10, 18, 20, 25)
Complete linkage: (0, 1), (10, 18, 20, 25)
Average linkage: (0, 1, 10), (18, 20, 25)

E. None of the above

Break & Quiz

Q: We want to perform hierarchical clustering with two clusters for the following points: 0,1, 10, 18, 20, 25. What are the clusters if we use single-linkage, complete-linkage, and average-linkage?

A. Single linkage: (0, 1, 10), (18, 20, 25)
   Complete linkage: (0, 1), (10, 18, 20, 25)
   Average linkage: (0, 1, 10), (18, 20, 25)

B. Single linkage: (0, 1, 10, 18, 20), (25)
   Complete linkage: (0), (1, 10, 18, 20, 25)
   Average linkage: (0), (1, 10, 18, 20, 25)

C. Single linkage: (0, 1), (10, 18, 20, 25)
   Complete linkage: (0, 1), (10, 18, 20, 25)
   Average linkage: (0, 1), (10, 18, 20, 25)

D. Single linkage: (0, 1), (10, 18, 20, 25)
   Complete linkage: (0, 1), (10, 18, 20, 25)
   Average linkage: (0, 1, 10), (18, 20, 25)

E. None of the above

We can find the two clusters by iterating through clustering with each linkage method:

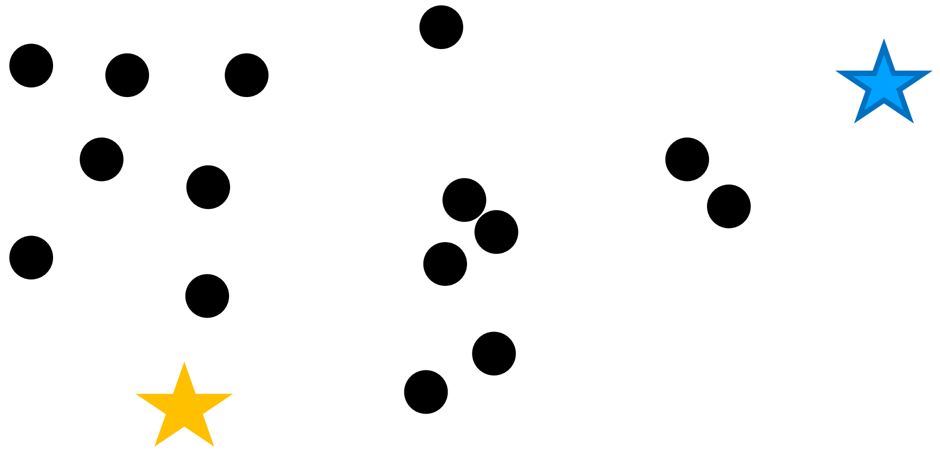
Single Linkage:
Iteration 1: (0, 1), 10, 18, 20, 25
Iteration 2: (0, 1), 10, (18, 20), 25
Iteration 3: (0, 1), 10, (18, 20, 25)
Iteration 4: (0, 1), (10, 18, 20, 25)

Complete Linkage:
Iteration 1: (0, 1), 10, 18, 20, 25
Iteration 2: (0, 1), 10, (18, 20), 25
Iteration 3: (0, 1), 10, (18, 20, 25)
Iteration 4: (0, 1, 10), (18, 20, 25)

Average Linkage:
Iteration 1: (0, 1), 10, 18, 20, 25
Iteration 2: (0, 1), 10, (18, 20), 25
Iteration 3: (0, 1), 10, (18, 20, 25)
Iteration 4: (0, 1, 10), (18, 20, 25)

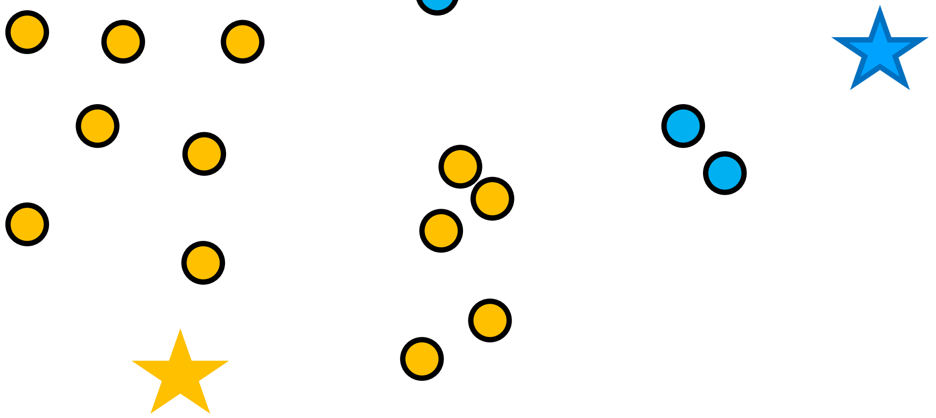
K-Means Clustering

Steps: **1**. Randomly pick k cluster centers



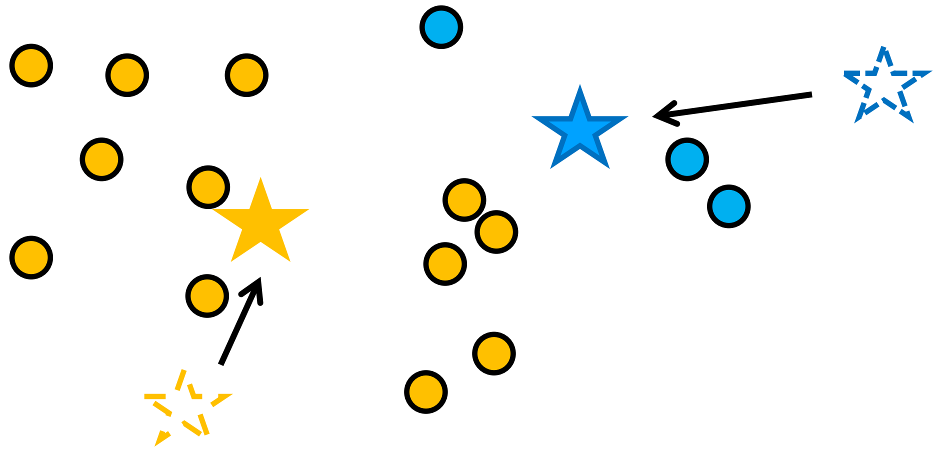
K-Means Clustering

2. Find closest center for each point



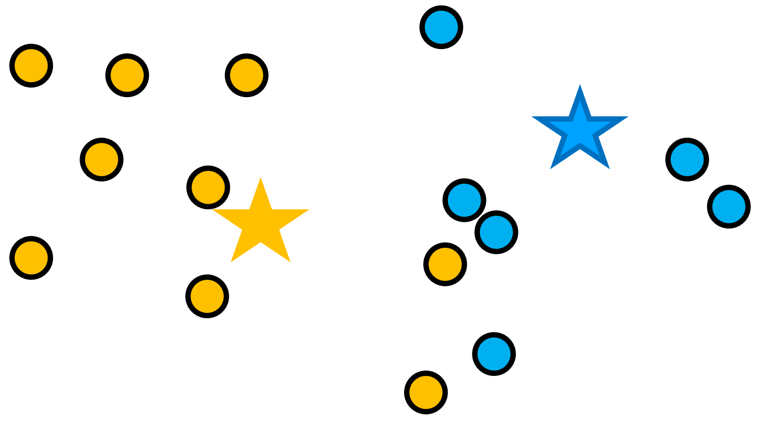
K-Means Clustering

3. Update cluster centers by computing centroids



K-Means Clustering

Repeat Steps 2 & 3 until convergence



K-means algorithm

Input: $x_1, x_2, \ldots, x_n, k$

Step 1: select k cluster centers $c_1, c_2, \ldots, c_k$

Step 2: for each point x_i, assign it to the closest center in Euclidean distance:

$$y(x_i) = \text{argmin}_j \, ||x_i - c_j||$$

Step 3: update all cluster centers as the centroids:

$$c_j = \frac{\sum_{x:y(x)=j} x}{\sum_{x:y(x)=j} 1}$$

Repeat Step 2 and 3 until cluster centers no longer change

Questions on k-means

- What is k-means trying to optimize?
- Will k-means stop (converge)?
- Will it find a global or local optimum?
- How to pick starting cluster centers?
- How many clusters should we use?

The optimization problem of k-means

- What is k-means trying to optimize?

$$\min_{c,y} \sum_{i=1}^{n} ||x_i - c_{y(x_i)}||^2$$

Questions on k-means

- Will k-means stop (converge)? Yes

Given a fixed dataset and a fixed number of clusters, there are only a **finite number of ways** to assign data points to clusters.

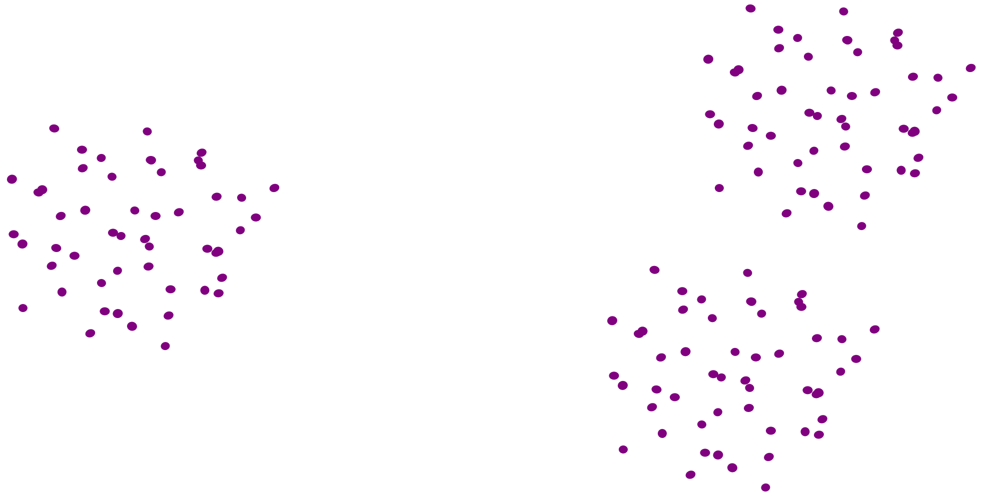
Each iteration consists of:

- Assignment Step: Assign each data point to the closest centroid.
- Update Step: Recompute centroids as the mean of assigned points.

These steps **always reduce or keep the same** the objective function (sum of squared distance), ensuring termination.

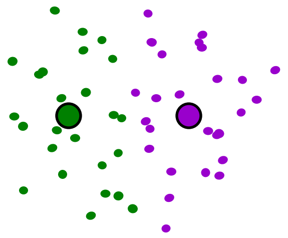
Questions on k-means

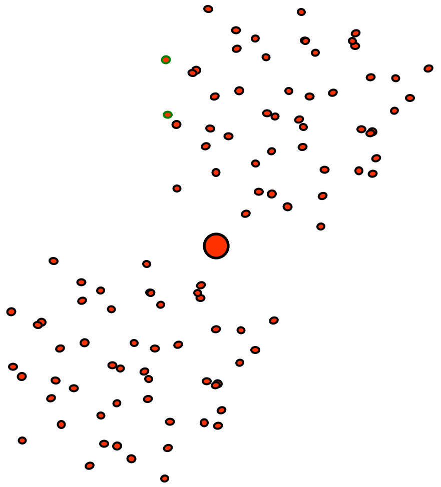
- Will it find a global or local optimum? (sadly no guarantee)



Questions on k-means

- Will it find a global or local optimum? (sadly no guarantee)





Questions on k-means

How many clusters should we use?

- Difficult problem
- Domain knowledge
- Elbow Method
  - Compute the within-cluster sum of squares (WCSS) for different values of k.
  - Plot WCSS vs. k and look for the **elbow point** where the reduction in WCSS slows down. The optimal k is typically at this elbow.

Break & Quiz

Q : In a k-means algorithm (with k = 3), the centroids are currently at the following 2D coordinates:

$$C_1 = (2, 1), \qquad C_2 = (6, 3), \qquad C_3 = (2, 6)$$

A data point x = (2, 3) needs to be assigned to a cluster. Using Euclidean distance, to which cluster centroid will x be assigned?

- A. C_1
- B. C_2
- C. C_3
- D. C_1 or C_3 (equidistant)
- E. None of the above

Break & Quiz

Q : In a k-means algorithm (with k = 3), the centroids are currently at the following 2D coordinates:

$$C_1 = (2, 1), \quad C_2 = (6, 3), \quad C_3 = (2, 6)$$

A data point x = (2, 3) needs to be assigned to a cluster. Using Euclidean distance, to which cluster centroid will x be assigned?

- **A. C_1**
- B. C_2
- C. C_3
- D. C_1 or C_3 (equidistant)
- E. None of the above

$$\boldsymbol{d(x, C_1) = \sqrt{(2-2)^2 + (3-1)^2} = 2}$$

$$\boldsymbol{d(x, C_2) = \sqrt{(2-6)^2 + (3-3)^2} = 4}$$

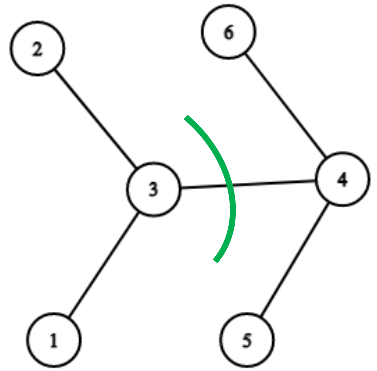
$$\boldsymbol{d(x, C_3) = \sqrt{(2-2)^2 + (3-6)^2} = 3}$$

The minimum distance is 2 which corresponds to centroid C_1. Therefore, the point x is assigned to cluster 1.

Graph-Based Clustering

Want: partition V into V_1 and V_2

- Implies a graph "cut"
- One idea: minimize the **weight** of the cut



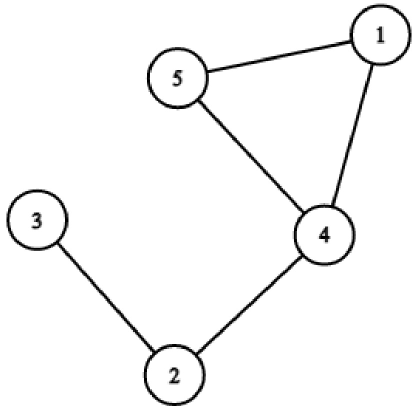
$$W(A,B) = \sum_{i \in A, j \in B} w_{ij}$$

$$\mathrm{cut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A}_i).$$

Partition-Based Clustering

How do we compute these?

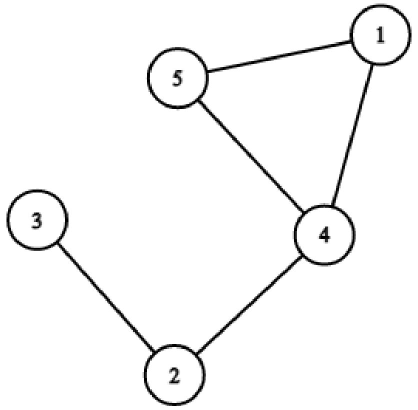
- Hard problem → heuristics
  - Greedy algorithm
  - "Spectral" approaches
- Spectral clustering approach:
  - **Adjacency** matrix



$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Partition-Based Clustering

- Spectral clustering approach:
  - **Adjacency** matrix
  - **Degree** matrix

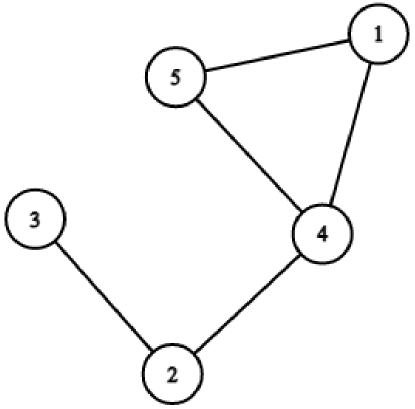


$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Spectral Clustering

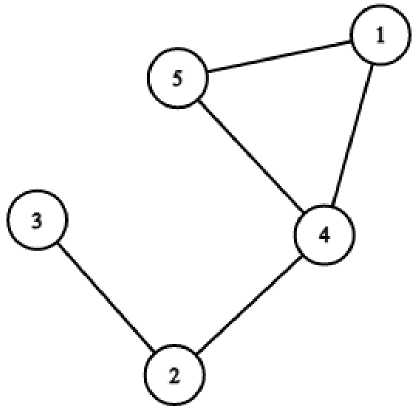
- Spectral clustering approach:
  - 1. Compute **Laplacian L** = **D** – **A**

(Important tool in graph theory)



$$L = \underbrace{\begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}}_{\text{Degree Matrix}} - \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Adjacency Matrix}} = \underbrace{\begin{bmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 2 & -1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}}_{\text{Laplacian}}$$

Spectral Clustering



- Spectral clustering approach:
  - 1. Compute **Laplacian** **L** = **D** – **A**
  - 1a (optional): compute normalized Laplacian:

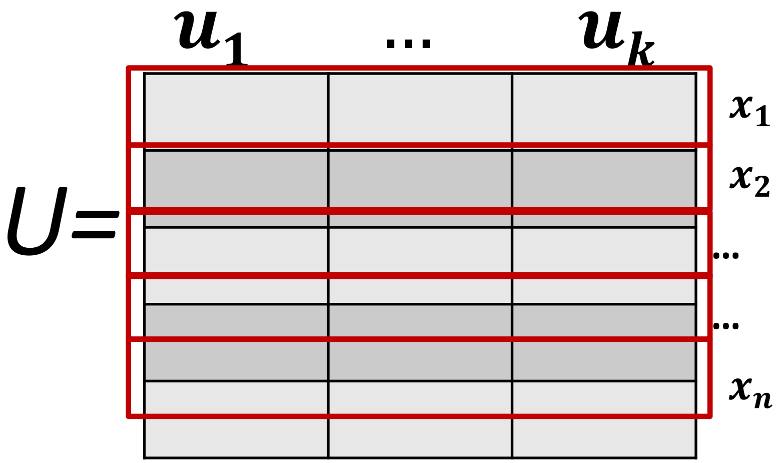
    L = I – D$^{-1/2}$AD$^{-1/2}$, or L = I – D^{-1}A

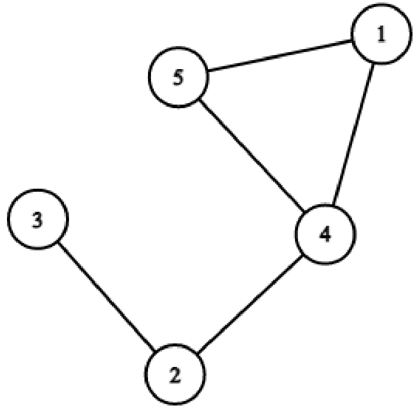
  - 2. Sort eigenvalues in desceding order and compute the ***bottom - k*** eigenvectors of **L**:

    $\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_k \in \mathbb{R}^n$

Spectral Clustering

- 3. Set *U* to be the *n* x *k* matrix with $\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_k$ as columns.
- 4. Take the *n* rows formed as points





- 5. Run k-means on the representations $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n$

Break & Quiz

Q 1.1: We have two datasets: a social network dataset S_1 which shows which individuals are friends with each other along with image dataset $S_{2.}$

What kind of clustering can we do? Assume we do not make additional data transformations.

- A. k-means on both S_1 and S_2
- B. graph-based on S_1 and k-means on S_2
- C. k-means on S_1 and graph-based on S_2
- D. hierarchical on S_1 and graph-based on S_2

Break & Quiz

Q 1.1: We have two datasets: a social network dataset S_1 which shows which individuals are friends with each other along with image dataset $S_{2.}$

What kind of clustering can we do? Assume we do not make additional data transformations.

- A. k-means on both S_1 and S_2
- **B. graph-based on S_1 and k-means on S_2**
- C. k-means on S_1 and graph-based on S_2
- D. hierarchical on S_1 and graph-based on S_2

Break & Quiz

Q 1.1: We have two datasets: a social network dataset S_1 which shows which individuals are friends with each other along with image dataset S_2.
What kind of clustering can we do? Assume we do not make additional data transformations.

- A. k-means on both S_1 and S_2 **(No: can't do k-means on graph)**
- **B. graph-based on S_1 and k-means on S_2**
- C. k-means on S_1 and graph-based on S_2 **(Same as A)**
- D. hierarchical on S_1 and graph-based on S_2 **(No: S_2 is not a graph)**

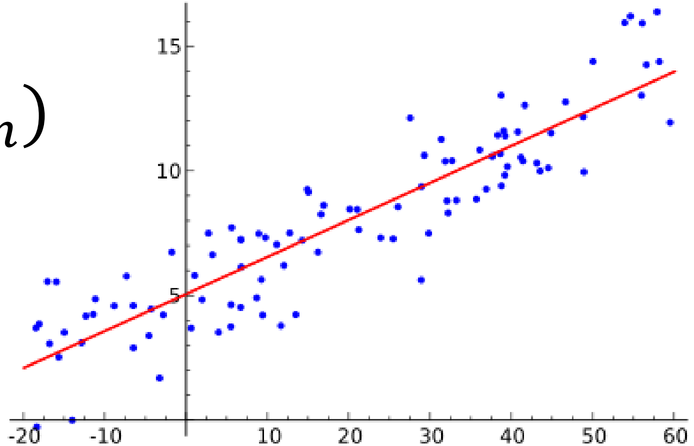
Linear Regression

Linear Regression

- Simplest form of regression
- Find a line that fits the data

- Example:
  - Training data $(x_1, y_1), \dots, (x_n, y_n)$
  - Find $a, b \in \mathbb{R}$ such that

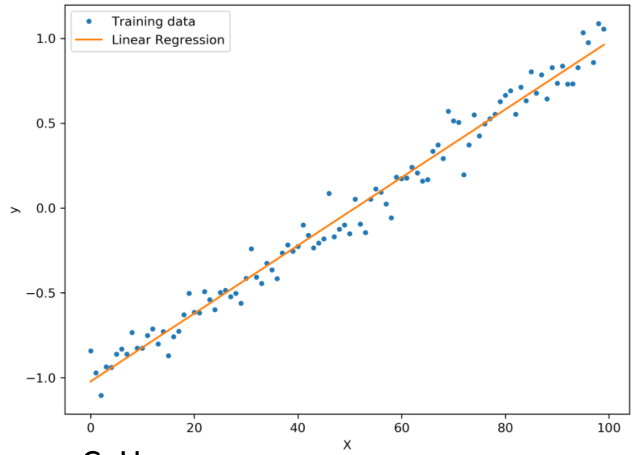
$$\forall i,\ y_i \approx a x_i + b$$



Linear Regression

- **Inputs**: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$
  - *x*'s are vectors, *y*'s are scalars.
  - "**Linear**": predict a linear combination

  of x components + intercept



C. Hansen

$$f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_d x_d = \theta_0 + x^T\theta$$

- **Want**: parameters θ

Linear Regression Setup

Problem Setup

- Goal: figure out how to minimize square loss
- Let’s organize it. Train set

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$$

Notational Trick

- When x is a scalar: $f_{(a,b)}(x) = ax + b$
- When x is a vector:

$$f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d = \theta_0 + x^T\theta$$

- Give x a "dummy dimension" to simplify notation

Old

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

New

$$x = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \quad f(x) = \begin{bmatrix} 1 & x_1 & x_2 & \cdots & x_d \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ x_d \end{bmatrix} = \langle x, \theta \rangle = x^T\theta$$

Linear Regression Setup

Problem Setup

- Train set $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$
- Take train features and make it a n*(d+1) matrix, and y a vector:

$$X = \begin{bmatrix} x_1^T \\ \ldots \\ x_n^T \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ \ldots \\ y_n \end{bmatrix}$$

- Then, the empirical risk is $\frac{1}{n}\|X\theta - y\|^2$

Finding The Estimated Parameters

Have our loss: $\frac{1}{n}\|X\theta - y\|^2$

- Could optimize it with SGD, etc…
- But the minimum also has a closed-form solution (vector calculus):

Hat: indicates an estimate

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

Not always invertible…

“Normal Equations”

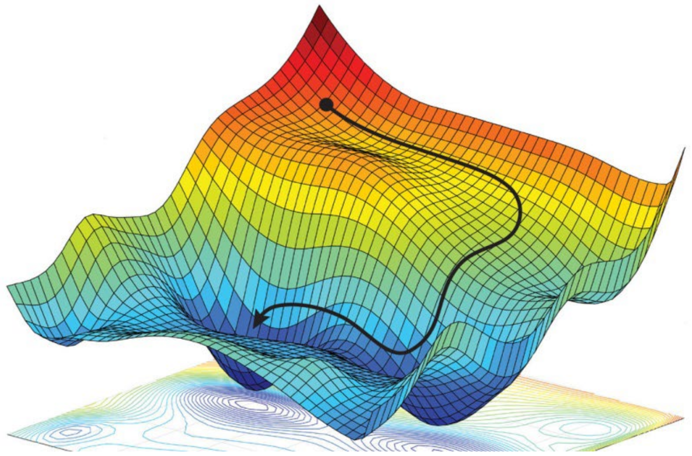
Gradients & Gradient Descent

- Gradient descent takes iterative steps to make loss function smaller

Gradient Descent

Input: dataset (X, y), loss function L, number of steps T, step size η

```
1.  Initialize $\theta_0$
2.  For $t = 1,2, \ldots, T$
3.      Calculate $g_t = \nabla L(\theta_{t-1}; X, y)$
4.      Update $\theta_t \leftarrow \theta_{t-1} - \eta g_t$
5.  Return $\theta_T$
```


M. Hutson

Linear Regression → Classification?

What if we want the same idea, but *y* is 0 or 1?

Need to convert the $\theta^T x$ to a probability in [0,1]

$$p(y=1|x) = \frac{1}{1+\exp(-\theta^T x)}$$

Logistic function

Why does this work?

If $\theta^T x$ is really big, $\exp(-\theta^T x)$ is really small → *p* close to 1

If really negative exp is huge → *p* close to 0

“Logistic Regression”

Break & Quiz

Q 2.3: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and $(x_2, y_2) = ([2,3,1], 4)$.

We have the weights $\beta_0 = 0, \beta_1 = 2, \beta_2 = 1, \beta_3 = 1$. What is the mean squared error (MSE) on the training set?

- A. 9
- B. 13/2
- C. 25/2
- D. 25

Break & Quiz

Q 2.3: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and$(x_2, y_2) = ([2,3,1], 4)$.

We have the weights $\beta_0 = 0, \beta_1 = 2, \beta_2 = 1, \beta_3 = 1$. What is the mean squared error (MSE) on the training set?

- A. 9
- B. 13/2
- C. **25/2**
- D. 25

Break & Quiz

Q 2.3: You have a dataset for regression given by $(x_1, y_1) = ([-1,0,1], 2)$ and $(x_2, y_2) = ([2,3,1], 4)$.

We have the weights $\beta_0 = 0, \beta_1 = 2, \beta_2 = 1, \beta_3 = 1$. What is the mean squared error (MSE) on the training set?

- A. 9
- B. 13/2
- C. **25/2**
- D. 25

Compute the predicted label for each data point, then compute the squared error for each data point, then take the mean error of the two points:

$$f(x_1) = \beta_0 - 1 * \beta_1 + 0 * \beta_2 + 1 * \beta_3 = -1$$
$$\ell(f(x_1), y_1) = (-1-2)^2 = 9$$

$$f(x_2) = \beta_0 + 2 * \beta_1 + 3 * \beta_2 + 1 * \beta_3 = 8$$
$$\ell(f(x_2), y_2) = (8-4)^2 = 16$$
$$MSE = (16+9)/2 = 2$$

Break & Quiz

Let $\binom{0}{0}, \binom{-2}{2}, \binom{-1}{3}, \binom{2}{2} \in \mathbb{R}^2$ be a dataset with labels 1,−1, 2, 4, respectively. We want to perform linear regression to find the weights $\theta_0, \theta_1, \theta_2$ that give us the line of best fit $f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ according to the mean squared error (MSE) loss. What is the resulting weights after one step of gradient descent with initialization $\theta_0 = \theta_1 = \theta_2 = 0$ and learning rate 0.1? Recall that the gradient of the squared error loss on a single data point (x, y) is $\nabla \ell(\theta; x, y) = 2(f(x) - y)x$.

A. $\theta_0 = -0.1, \theta_1 = 0.4, \theta_2 = -0.6$

B. $\theta_0 = 0.1, \theta_1 = -0.4, \theta_2 = 0.6$

C. $\theta_0 = -0.3, \theta_1 = -0.4, \theta_2 = -0.2$

D. $\theta_0 = 0.3, \theta_1 = 0.4, \theta_2 = 0.2$

E. None of the above.

Break & Quiz

Let $\binom{0}{0}, \binom{-2}{2}, \binom{-1}{3}, \binom{2}{2} \in \mathbb{R}^2$ be a dataset with labels 1,−1, 2, 4, respectively. We want to perform linear regression to find the weights $\theta_0, \theta_1, \theta_2$ that give us the line of best fit $f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ according to the mean squared error (MSE) loss. What is the resulting weights after one step of gradient descent with initialization $\theta_0 = \theta_1 = \theta_2 = 0$ and learning rate 0.1? Recall that the gradient of the squared error loss on a single data point (x, y) is $\nabla\ell(\theta;\, x, y) = 2(f(x) - y)x$.

A. $\theta_0 = -0.1, \theta_1 = 0.4, \theta_2 = -0.6$

B. $\theta_0 = 0.1, \theta_1 = -0.4, \theta_2 = 0.6$

C. $\theta_0 = -0.3, \theta_1 = -0.4, \theta_2 = -0.2$

D. $\boldsymbol{\theta_0 = 0.3, \theta_1 = 0.4, \theta_2 = 0.2}$

E. *None of the above.*

Break & Quiz

Let $\binom{0}{0}, \binom{-2}{2}, \binom{-1}{3}, \binom{2}{0} \in \mathbb{R}^2$ be a dataset with labels 1,−1, 2, 4, respectively. We want to perform linear regression to find the weights $\theta_0, \theta_1, \theta_2$ that give us the line of best fit $f(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ according to the mean squared error (MSE) loss. What is the resulting weights after one step of gradient descent with initialization $\theta_0 = \theta_1 = \theta_2 = 0$ and learning rate 0.1? Recall that the gradient of the squared error loss on a single data point (x, y) is $\nabla \ell(\theta; x, y) = 2(f(x) - y)x$.

$\hat{y} = 0$ for all i, we compute that

$$\nabla L(\theta) = \frac{2}{4} \times ((-1) \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + 1 \times \begin{pmatrix} 1 \\ -2 \\ 2 \end{pmatrix} + (-2) \times \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix} + (-4) \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}) = \begin{pmatrix} -3 \\ -4 \\ -2 \end{pmatrix}$$

Thus, the updated value of θ is $\begin{pmatrix} 0.3 \\ 0.4 \\ 0.2 \end{pmatrix}$ and the correct answer is D.

K-Nearest Neighbors

K-nearest neighbors for classification

- **Input**: **Training data** $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

  Distance function $d(\mathbf{x}_i, \mathbf{x}_j)$; **number of neighbors** k; **test data** $\mathbf{x}^*$

1. Find the k training instances $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}$ closest to $\mathbf{x}^*$ under $d(\mathbf{x}_i, \mathbf{x}_j)$
2. Output y^* , the majority class of $y_{i_1}, \ldots, y_{i_k}$. Break ties randomly.

k-Nearest Neighbors: Distances

Discrete features: Hamming distance

$$d_H(x^{(i)}, x^{(j)}) = \sum_{a=1}^{d} 1\{x_a^{(i)} \neq x_a^{(j)}\}$$

Continuous features:

Euclidean distance:

$$d(x^{(i)}, x^{(j)}) = \left(\sum_{a=1}^{d} (x_a^{(i)} - x_a^{(j)})^2 \right)^{\frac{1}{2}}$$

L1 (Manhattan) dist.:

$$d(x^{(i)}, x^{(j)}) = \sum_{a=1}^{d} |x_a^{(i)} - x_a^{(j)}|$$

k-Nearest Neighbors: Regression

Training/learning: given

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$$

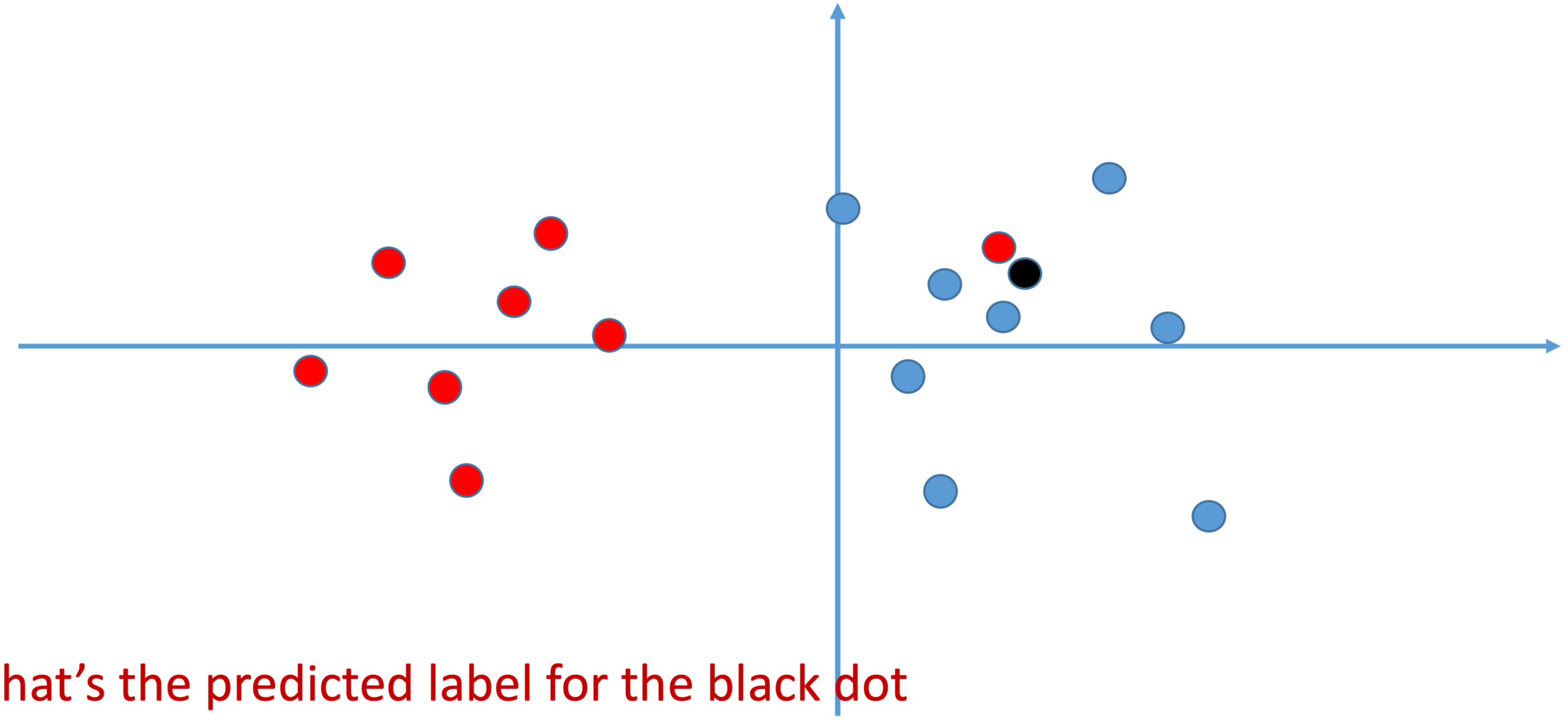
Prediction: for x , find ***k*** most similar training points

Return

$$\hat{y} = \frac{1}{k} \sum_{i=1}^{k} y^{(i)}$$

- I.e., among the ***k*** points, output mean label.

Effect of k



What's the predicted label for the black dot using 1 neighbor? 3 neighbors?

Break & Quiz

We are given four training points in 2D: $x_1 = (1,0), y_1 = Class\ A$; $x_2 = (0,2), y_2 = Class\ B$; $x_3 = (3,1), y_3 = Class\ A$; $and\ x_4 = (2,3), y_4 = Class\ B$. Suppose we are running kNN for the input $x = (0,0)$ using Manhattan distance. What is the set of distances to the k = 3 nearest neighbors, and what is the final predicted class?

- A. Distances {1, 2, 4}, Class A.
- B. Distances {1, 2, 4}, Class B.
- C. Distances {1, 2, $\sqrt{10}$}, Class A.
- D. Distances {1, 2, 5}, Class B.
- E. None of the above.

Break & Quiz

We are given four training points in 2D: $x_1 = (1,0), y_1 = Class\ A$; $x_2 = (0,2), y_2 = Class\ B$; $x_3 = (3,1), y_3 = Class\ A$; $and\ x_4 = (2,3), y_4 = Class\ B$. Suppose we are running kNN for the input $x = (0,0)$ using Manhattan distance. What is the set of distances to the k = 3 nearest neighbors, and what is the final predicted class?

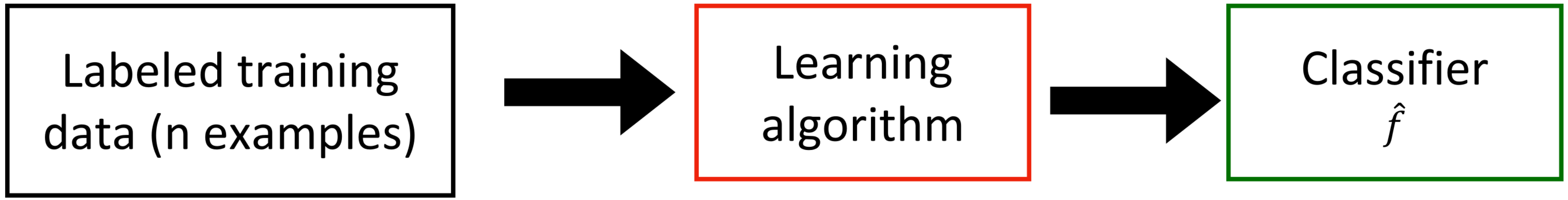
- A. Distances {1, 2, 4}, Class A.
- B. Distances {1, 2, 4}, Class B.
- C. Distances {1, 2, $\sqrt{10}$}, Class A.
- D. Distances {1, 2, 5}, Class B.
- E. None of the above.

$d(x, x_1) = |0-1| + |0-0| = 1$ Label : Class A
$d(x, x_2) = |0-0| + |0-2| = 2$ Label : Class B
$d(x, x_3) = |0-3| + |0-1| = 4$ Label : Class A
$d(x, x_4) = |0-2| + |0-3| = 5$ Label : Class B

The sorted distances are 1, 2, 4, 5. The k = 3 nearest neighbors are x_1, x_2, x_3. The set of distances to these neighbors is {1, 2, 4}. The labels of these neighbors are {Class A, Class B, Class A}. By majority vote (2 for A, 1 for B), the predicted class is Class A.

Supervised Machine Learning

Statistical modeling approach



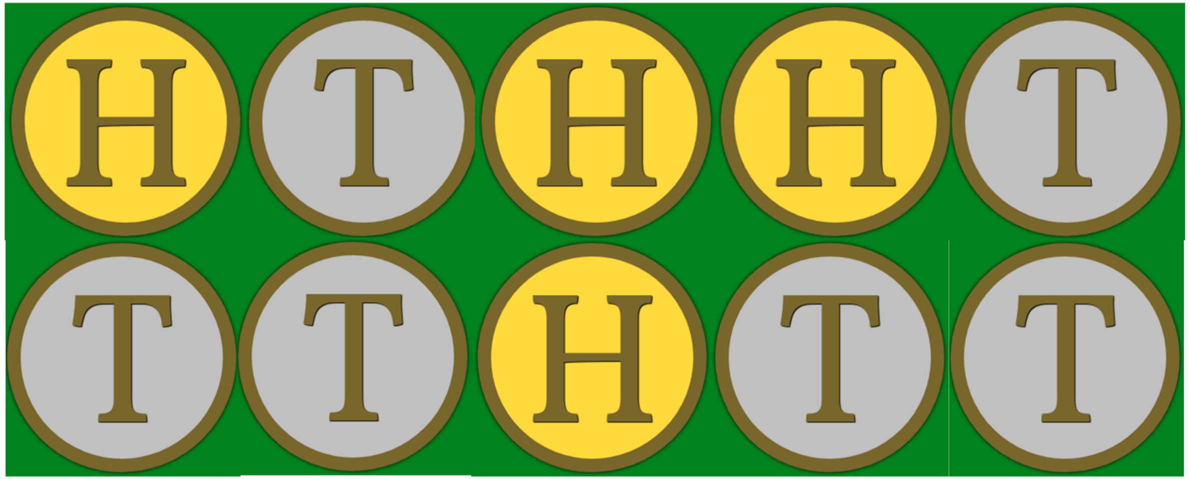
$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$

drawn **independently** from a fixed underlying distribution, also called the i.i.d. (independent and identically distributed) assumption

select $\hat{f}(\theta)$ from a pool of models $\mathcal{F}$ that **best describe the data observed**

Maximum Likelihood Estimation: An Example

Flip a coin 10 times, how can you estimate $\theta = p(Head)$?



Intuitively, $\theta = 4/10 = 0.4$

How good is θ?

It depends on how likely it is to generate the observed data

$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ (Let’s forget about label for a second)

Likelihood function

$$L(\theta) \underset{\substack{\Big\uparrow \\ \text{Under i.i.d assumption}}}{=} \Pi_i p(\mathbf{x}_i | \theta)$$

Interpretation: How **probable** (or how likely) is the data given the probabilistic model p_θ?

How good is θ?

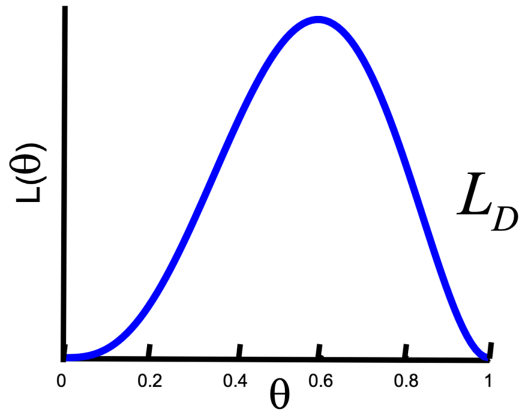
It depends on how likely it is to generate the observed data

$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ (Let's forget about label for a second)

Likelihood function $$L(\theta) = \Pi_i p(\mathbf{x}_i|\theta)$$

H,T, T, H, H

$$L_D(\theta) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta$$



Bernoulli distribution

Log-likelihood function

$$L_D(\theta) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta$$

$$= \theta^{N_H} \cdot (1-\theta)^{N_T}$$

Log-likelihood function

$$\ell(\theta) = \log L(\theta)$$

$$= N_H \log\theta + N_T \log(1-\theta)$$

Maximum Likelihood Estimation (MLE)

Find optimal θ^* to maximize the likelihood function (and log-likelihood)

$$\theta^* = \operatorname{argmax} \; N_H \log\theta + N_T \log(1-\theta)$$

$$\frac{\partial l(\theta)}{\partial \theta} = \frac{N_H}{\theta} - \frac{N_T}{1-\theta} = 0 \;\Rightarrow\; \theta^* = \frac{N_H}{N_T + N_H}$$

which confirms your intuition!

Break & Quiz

A coin is flipped 20 times, and it lands on heads 12 times. What is the Maximum Likelihood Estimate (MLE) for the probability of the coin landing on heads, P(heads)?

- A. 0.5
- B. 0.4
- C. 0.6
- D. 12
- E. This cannot be determined without a prior.

Break & Quiz

A coin is flipped 20 times, and it lands on heads 12 times. What is the Maximum Likelihood Estimate (MLE) for the probability of the coin landing on heads, P(heads)?

- A. 0.5
- B. 0.4
- **C. 0.6**
- D. 12
- E. This cannot be determined without a prior.

The correct answer is C. The Maximum Likelihood Estimate (MLE) finds the parameter value that makes the observed data most probable. For a sequence of coin flips (a Bernoulli process), the MLE for the probability of heads is simply the proportion of heads observed in the data. The calculation is the number of heads divided by the total number of flips: 12/20 = 0.6