

Task 1 : clustering

Task 2 : feature extraction

## IEMS 304 Lecture 8: Unsupervised Learning

---

Yiping Lu

yiping.lu@northwestern.edu

*Industrial Engineering & Management Sciences*  
*Northwestern University*



NORTHWESTERN  
UNIVERSITY

clustering

$\{(x_i)\}_{i=1}^n \rightarrow \text{Classification.}$

Supervised Classification.

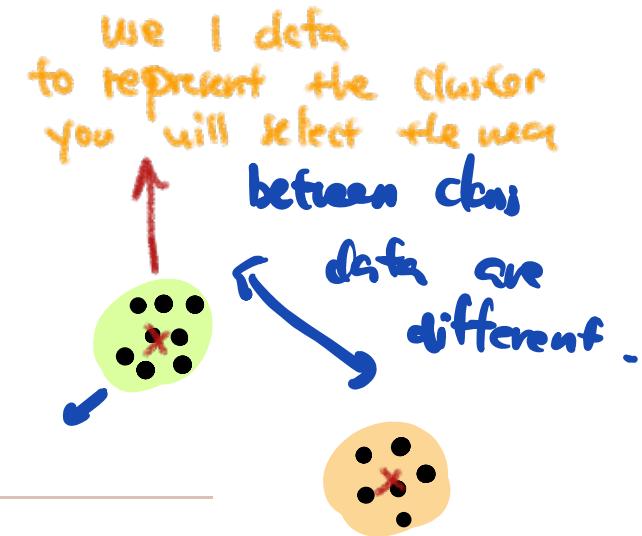
$\{(x_i, y_i)\}_{i=1}^n$

Classification Result.

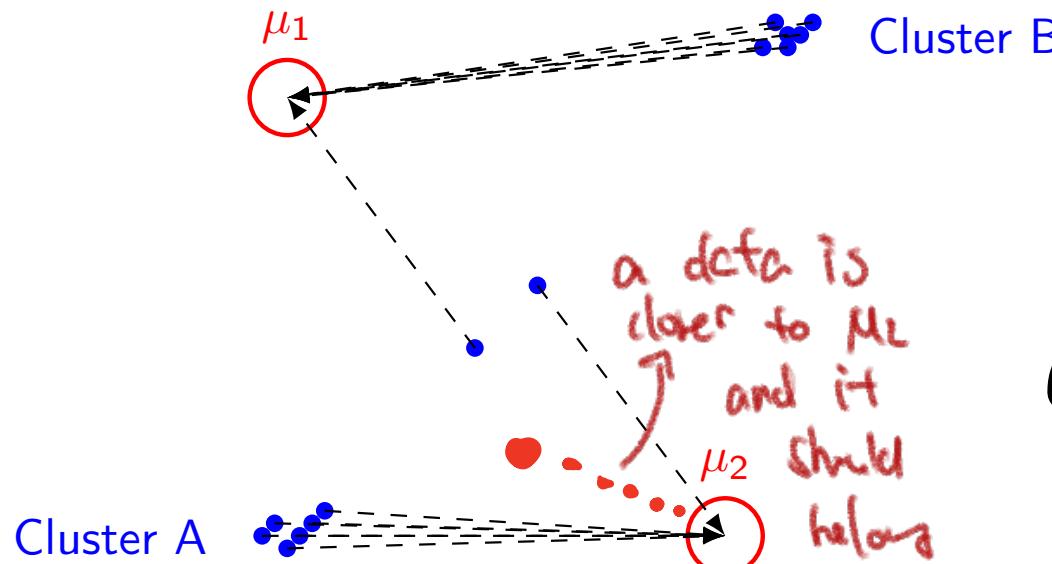
k-means  
J

K is the number of  
classes.

in class  
are similar



# Iteration 1: Initialization & Forced Assignment



## Assignment Summary (Iteration 1):

- $\mu_1 = (1, 4.5)$  gets: all Cluster B points (6 pts) + ambiguous point (2.5, 2.5) [total 7 pts].
- $\mu_2 = (4.5, 1)$  gets: all Cluster A points (6 pts) + ambiguous point (3, 3) [total 7 pts].

## Updated centroids (computed as the mean):

$$\mu'_1 = \left( \frac{30+2.5}{7}, \frac{30+2.5}{7} \right) \approx (4.643, 4.643)$$

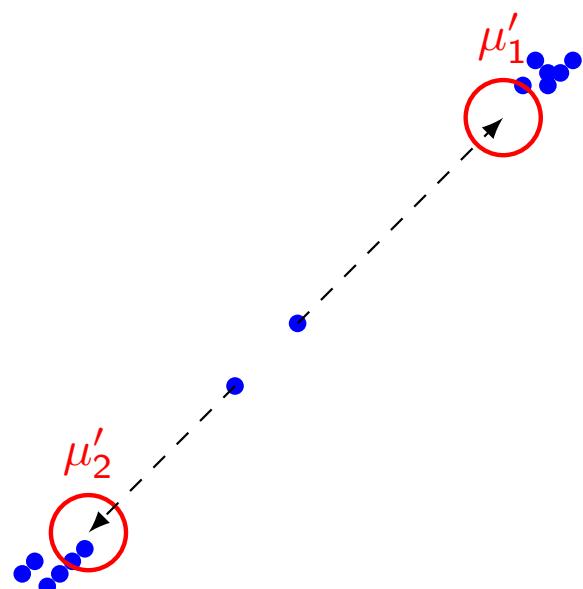
$$\mu'_2 = \left( \frac{6.3+3}{7}, \frac{6.3+3}{7} \right) \approx (1.329, 1.329)$$

① randomly initialize the k-means.

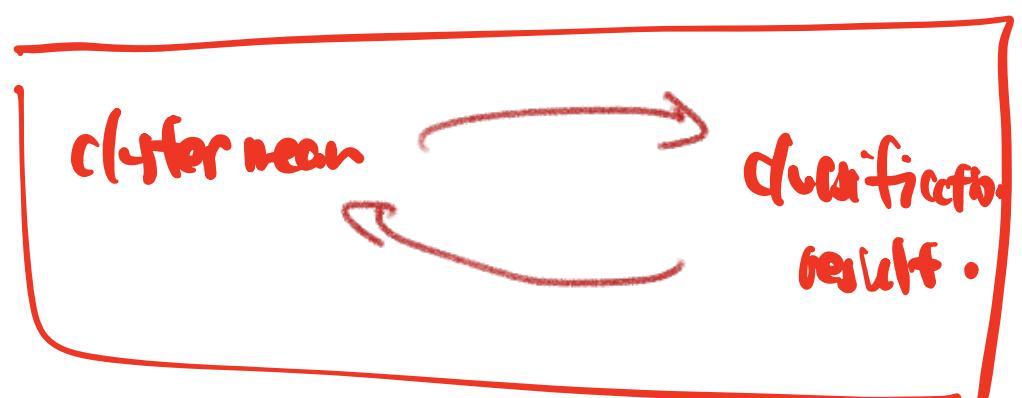
② once you know the k-means, you can do classification

iterative .

## Iteration 2: Reassignment



③ update the cluster mean .



### Reassignment (Iteration 2):

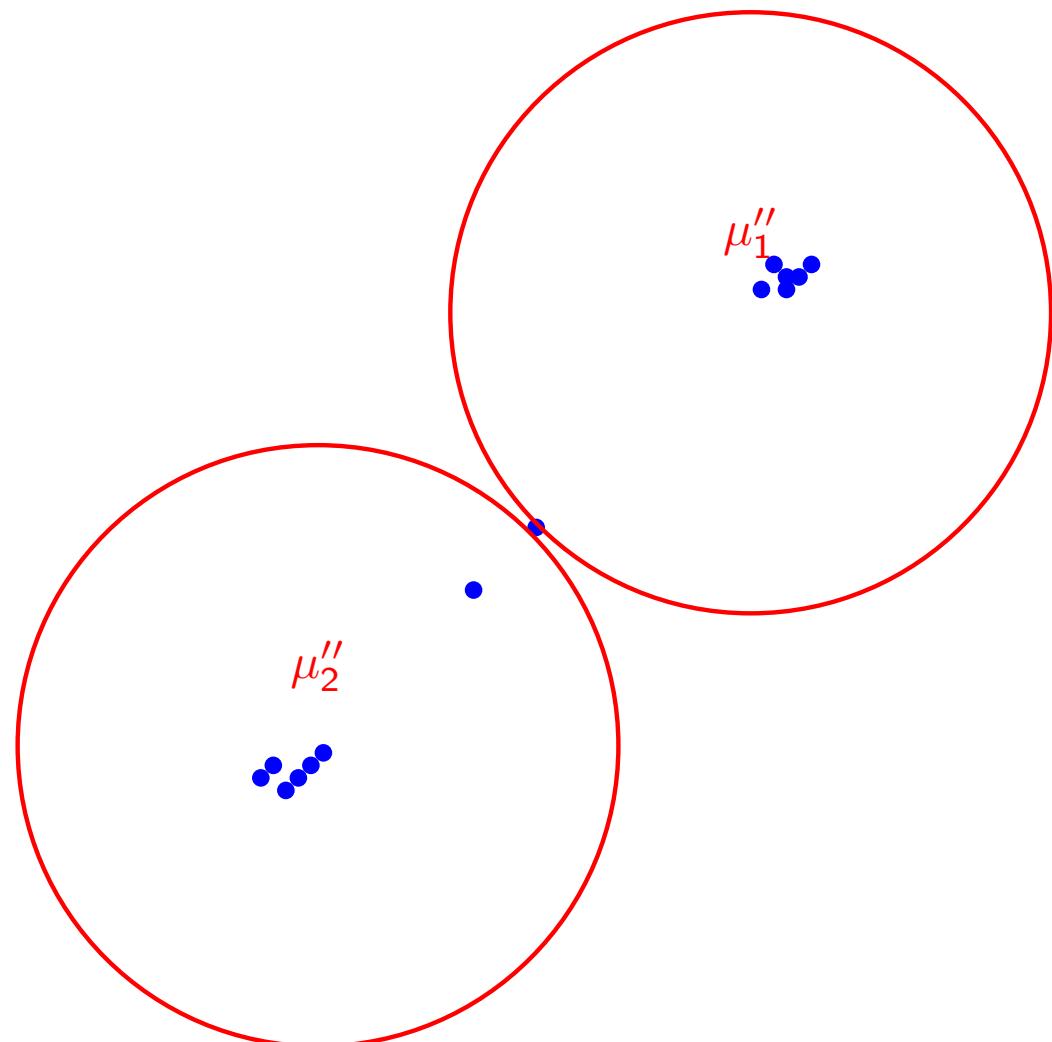
- (2.5, 2.5) switches from  $\mu_1$  to  $\mu'_2$  (closer to (1.329, 1.329)).
- (3, 3) switches from  $\mu_2$  to  $\mu'_1$  (closer to (4.643, 4.643)).

### New centroids:

$$\mu''_1 = \left( \frac{30+3}{7}, \frac{30+3}{7} \right) = \left( \frac{33}{7}, \frac{33}{7} \right) \approx (4.714, 4.714)$$

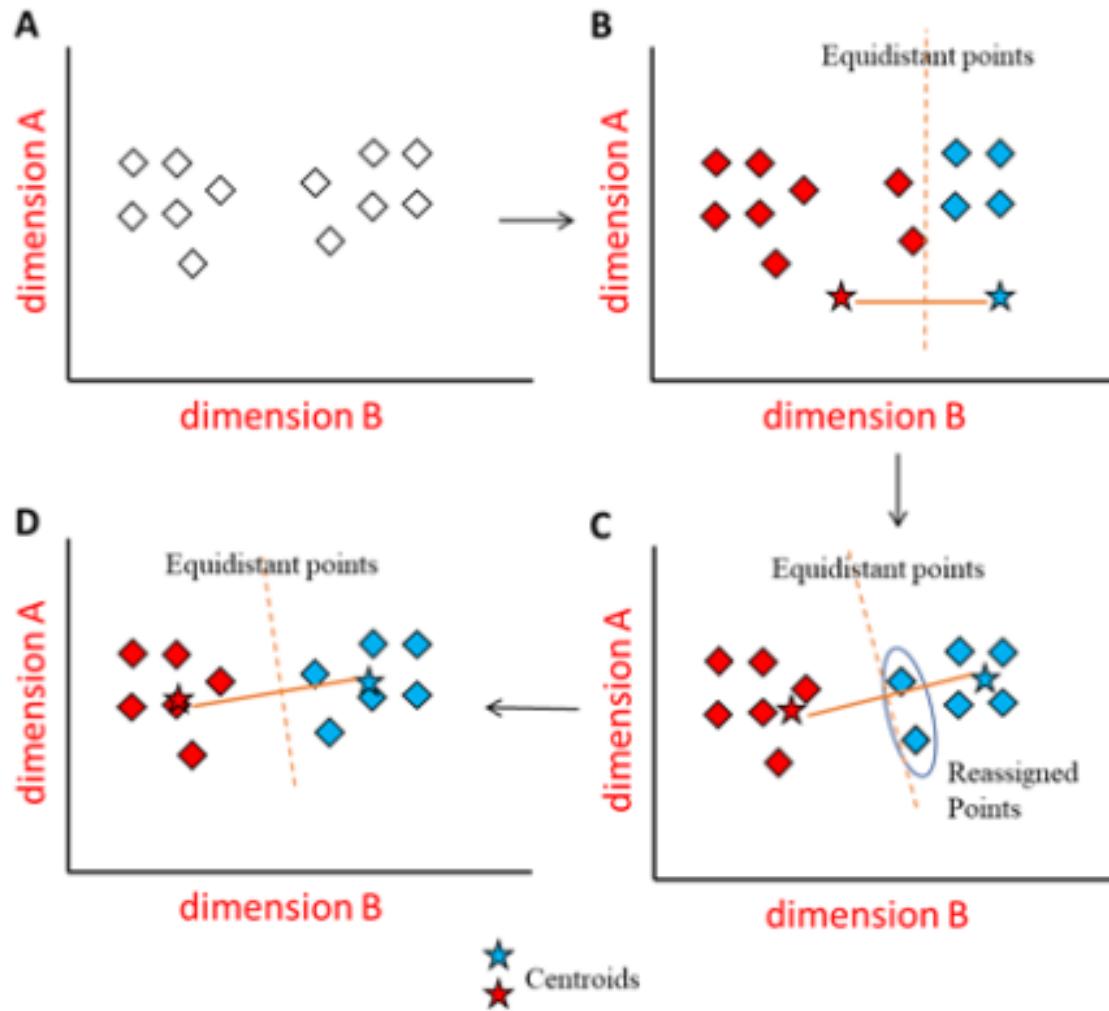
$$\mu''_2 = \left( \frac{6.3+2.5}{7}, \frac{6.3+2.5}{7} \right) = \left( \frac{8.8}{7}, \frac{8.8}{7} \right) \approx (1.257, 1.257)$$

## Iteration 3: Convergence



**Convergence:** With centroids  $\mu''_1 \approx (4.714, 4.714)$  and  $\mu''_2 \approx (1.257, 1.257)$ , all data points are now correctly grouped according to their true clusters.

# $k$ -means



# $k$ -means as Optimization

$k$ -means aims to minimize the total within cluster (square) distance

$$\min_{\{C_j\}, \{\mu_j\}} \sum_{j=1}^k \sum_{x \in C_j} \|x - \mu_j\|^2$$

*How you cluster your data.*      *the mean of the cluster*

sum over all the data in cluster  $C_j$

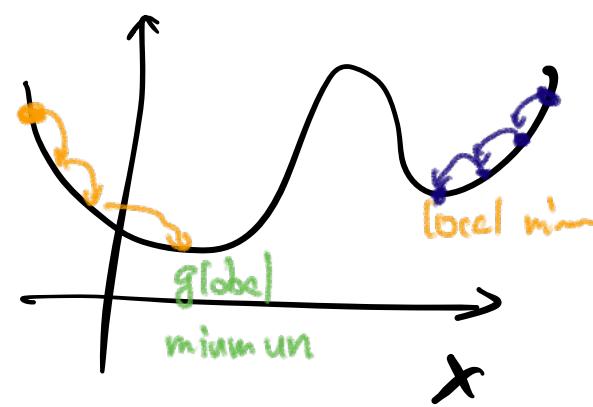
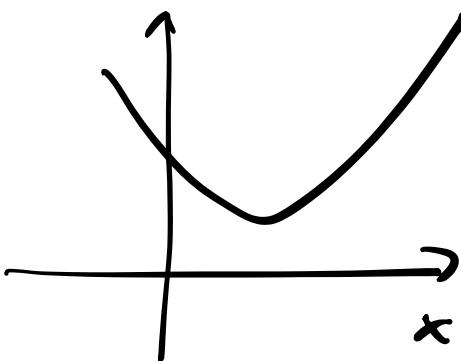
$k$ -means as alternating direction optimization algorithm

□ **Assignment:** Assign each  $x$  to its nearest  $\mu_j$  (minimizes distance).

□ **Update:** Recompute  $\mu_j$  as the mean of  $C_j$  (minimizes variance).

① why we select the mean as the representation

the mean  $\mu$  is minimize  $\sum_{X_i \text{ in the cluster}} \|x_i - \mu\|^2$



**Convex**

linear Regression

**Non-Convex**

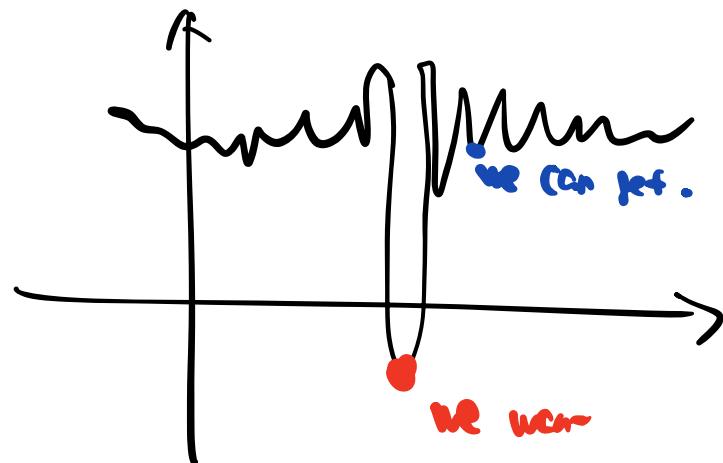
k-means

Neural Network

k-means  $\leftrightarrow$ .

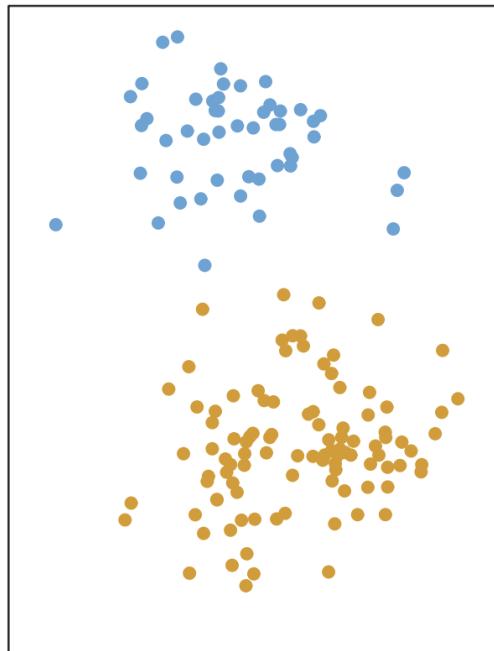
with high prob.

goes to global minimum -

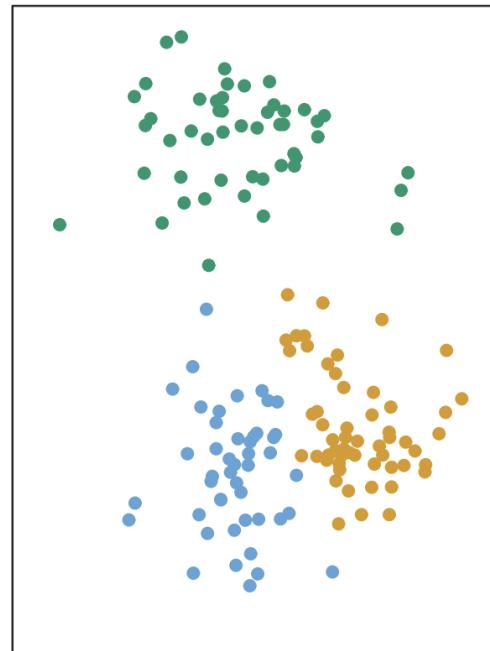


# Wrong $k$ can be Problematic

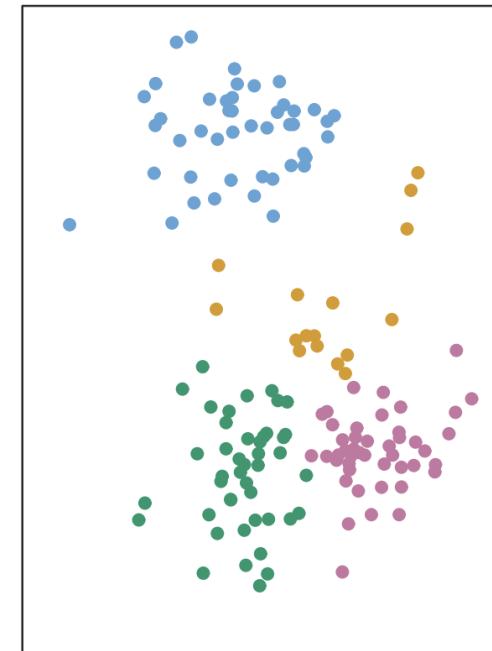
K=2



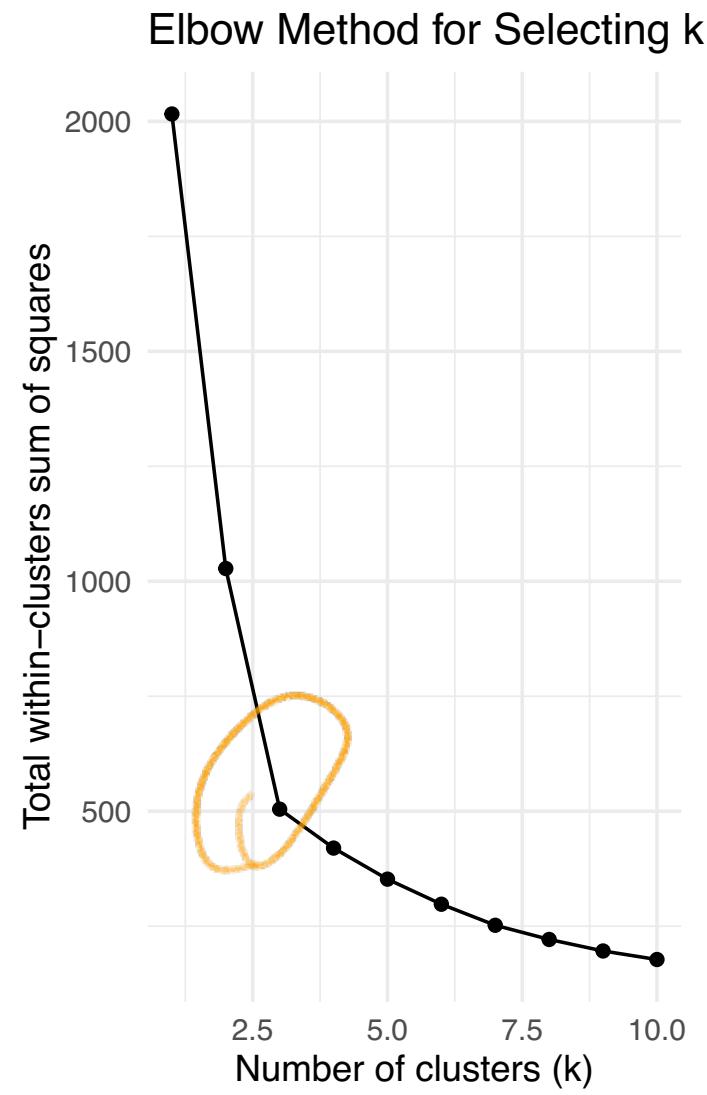
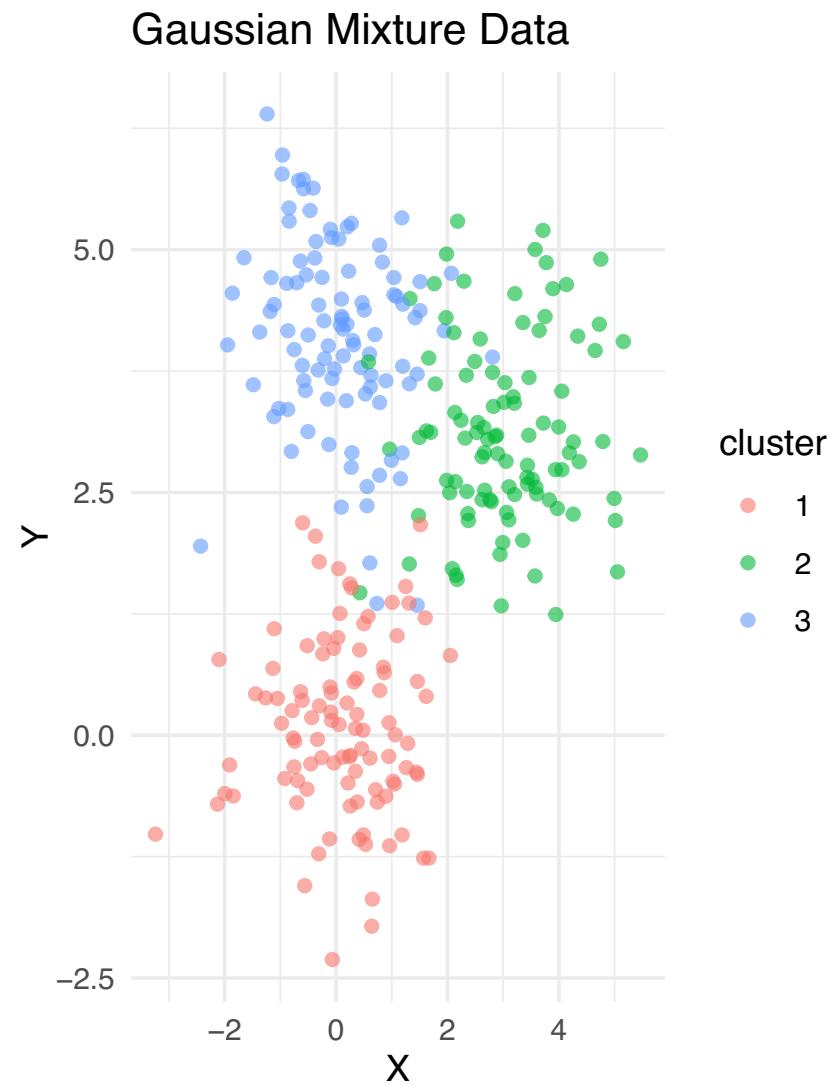
K=3



K=4



# How to Select $k$ : Elbow Effect

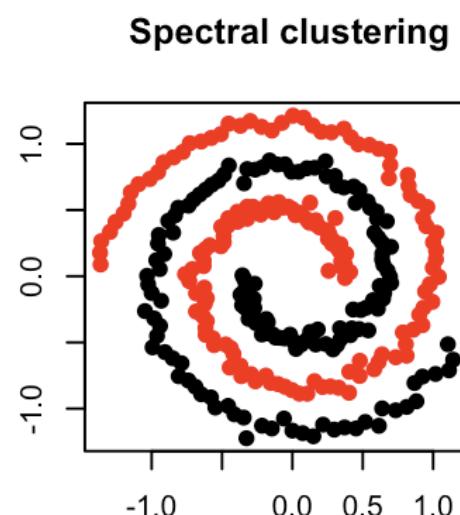
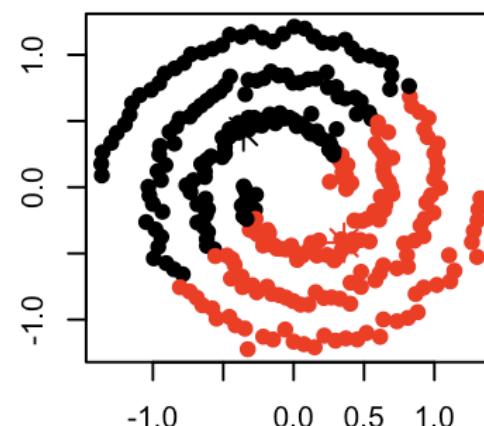
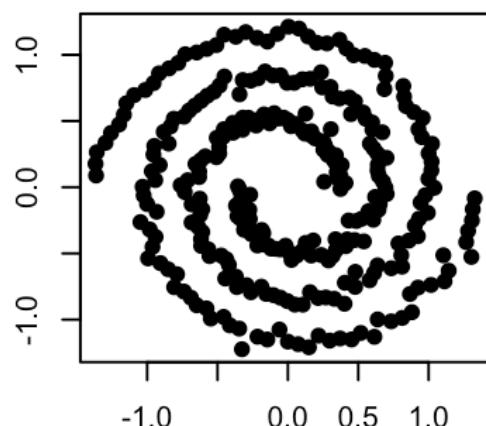


# Spectral Clustering

---

# Spectral Clustering

The failure here is because we just use mean to represent a cluster  
→ we assume the clusters are convex



# Spectral Clustering

We first represent data as a weighted graph  $G(V, E)$  with weights  $w_{ij}$ .

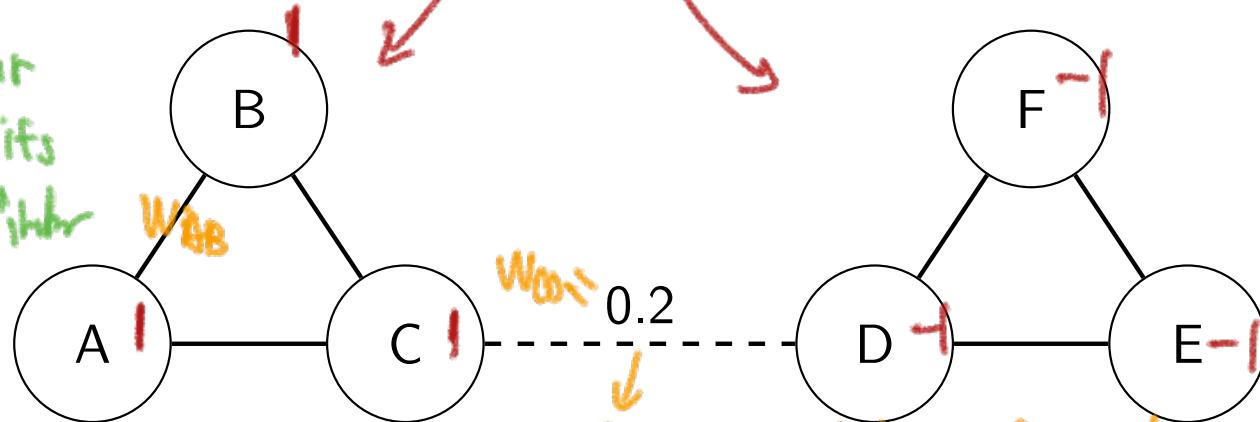
Consider the Dirichlet form,

$$\frac{1}{2} \sum_{i,j} w_{ij} (f(i) - f(j))^2 = f^T L f, \quad (\text{Why?})$$

where  $L$  is the graph Laplacian defined as  $L = D - W$  (where  $D$  is the degree matrix).

minimizing Dirichlet form.

e.g. Connect your data with its k-nearest neighbor  $W_{AB}$   
 $w = h(-\text{distance between data})$   
↓  
Some function



What would happen if we minimizing this form?

If two data are less similar then the weight becomes smaller.

# Quadratic Function as a Quadratic Form

put all variable as a vector .

$$v^T A v = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} 3 & 2 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 3x^2 + 2xy + 2xy + 2y^2 = 3x^2 + 4xy + 2y^2.$$

all the coeff of  $x$

$$(x, y) \begin{pmatrix} 3x+2y \\ 2x+2y \end{pmatrix} = x(3x+2y) + y(2x+2y)$$

$$(x, y) \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = a_{11}x^2 + (a_{12} + a_{21})xy + a_{22}y^2$$

fix  $a_{12} = a_{21}$ . which means  $A$  is symmetric

$$A^T = A.$$

# Why is the Dirichlet Form Equal to $f^T L f$ ?

Consider the Dirichlet form:

$$\frac{1}{2} \sum_{i,j} w_{ij} (f(i) - f(j))^2 = \frac{1}{2} \sum_{i,j} w_{ij} [f(i)^2 - 2f(i)f(j) + f(j)^2].$$

□ terms involving  $f(i)^2$ :

$$\begin{aligned} & \frac{1}{2} \left( \sum_{i,j} w_{ij} f(i)^2 + \sum_{i,j} w_{ij} f(j)^2 \right) \\ &= \sum_i f(i)^2 \sum_j w_{ij} = \sum_i d_i f(i)^2. \end{aligned}$$

□ The cross term simplifies to:

$$- \sum_{i,j} w_{ij} f(i)f(j).$$

$$\frac{1}{2} \sum_{i,j} w_{ij} (f(i) - f(j))^2 = \sum_i d_i f(i)^2 - \sum_{i,j} w_{ij} f(i)f(j).$$

At the same time,

$$f^T L f = \sum_i d_i f(i)^2 - \sum_{i,j} w_{ij} f(i)f(j), \text{ where } L = D - W,$$

# Understanding the Dirichlet Form

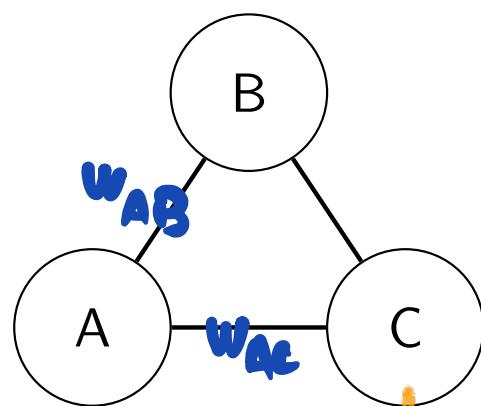
## Definition

The Dirichlet form on a graph is defined as:

$$\frac{1}{2} \sum_{i,j} w_{ij} (f(i) - f(j))^2 = f^T L f.$$

- It sums the squared differences of the function values  $f(i)$  over every edge, weighted by  $w_{ij}$ .
- A small value of  $f^T L f$  indicates that neighboring nodes (with high similarity  $w_{ij}$ ) have similar function values.
- Minimizing the Dirichlet form under constraints leads to smooth functions on the graph, thus revealing inherent cluster structure.

# Computing the Graph Laplacian



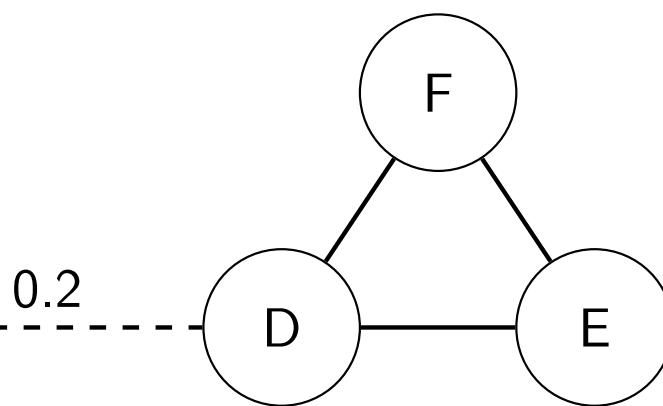
## Step 1: Define the Matrices

- Weighted Adjacency Matrix  $W$ :

For each edge  $(i, j)$ ,  $w(i, j) = 1$   
except for the edge between  $C$  and  
 $D$  where  $w(C, D) = 0.2$ .

- Degree Matrix  $D$ : Diagonal with

$d_A = 2, d_B = 2, d_C = 2.2, d_D = 2.2, d_E = 2, d_F = 2$

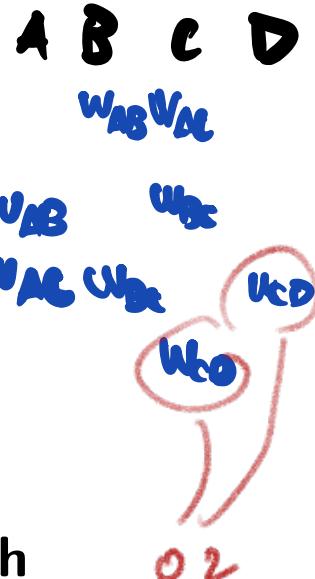


## Step 2: Compute the Graph Laplacian

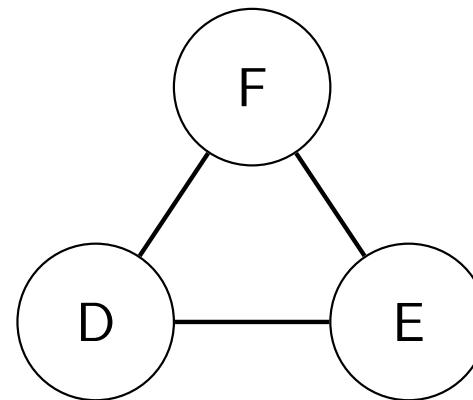
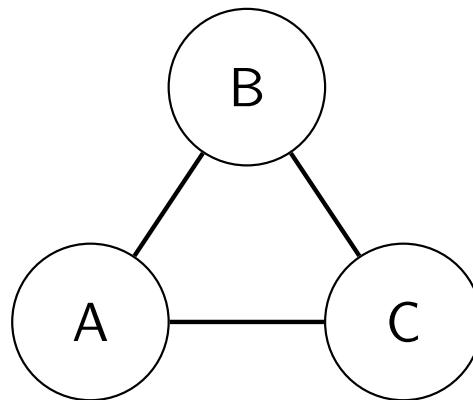
$$L = D - W$$

$$= \begin{pmatrix} 2 & -1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 2.2 & -0.2 & 0 & 0 \\ 0 & 0 & -0.2 & 2.2 & -1 & -1 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{pmatrix}.$$

$[d_A \ d_B \ d_C \ d_D \ d_E \ d_F]$



# Computing the Graph Laplacian



What is the smallest eigenvalue/eigenvectors of the graph laplacian?  
What would happen if we have  $l$ -connected component

# Spectral Clustering

$$\min f^\top L f \quad \text{s.t. } f^\top \mathbb{1} = 0, \|f\|_2 = 1$$

→ Closed form!!! is the smallest eigen vector!

Then run a  $k$ -means on the spectral clustering representation  $f$ . (homework)

① Case1: Do No classification.

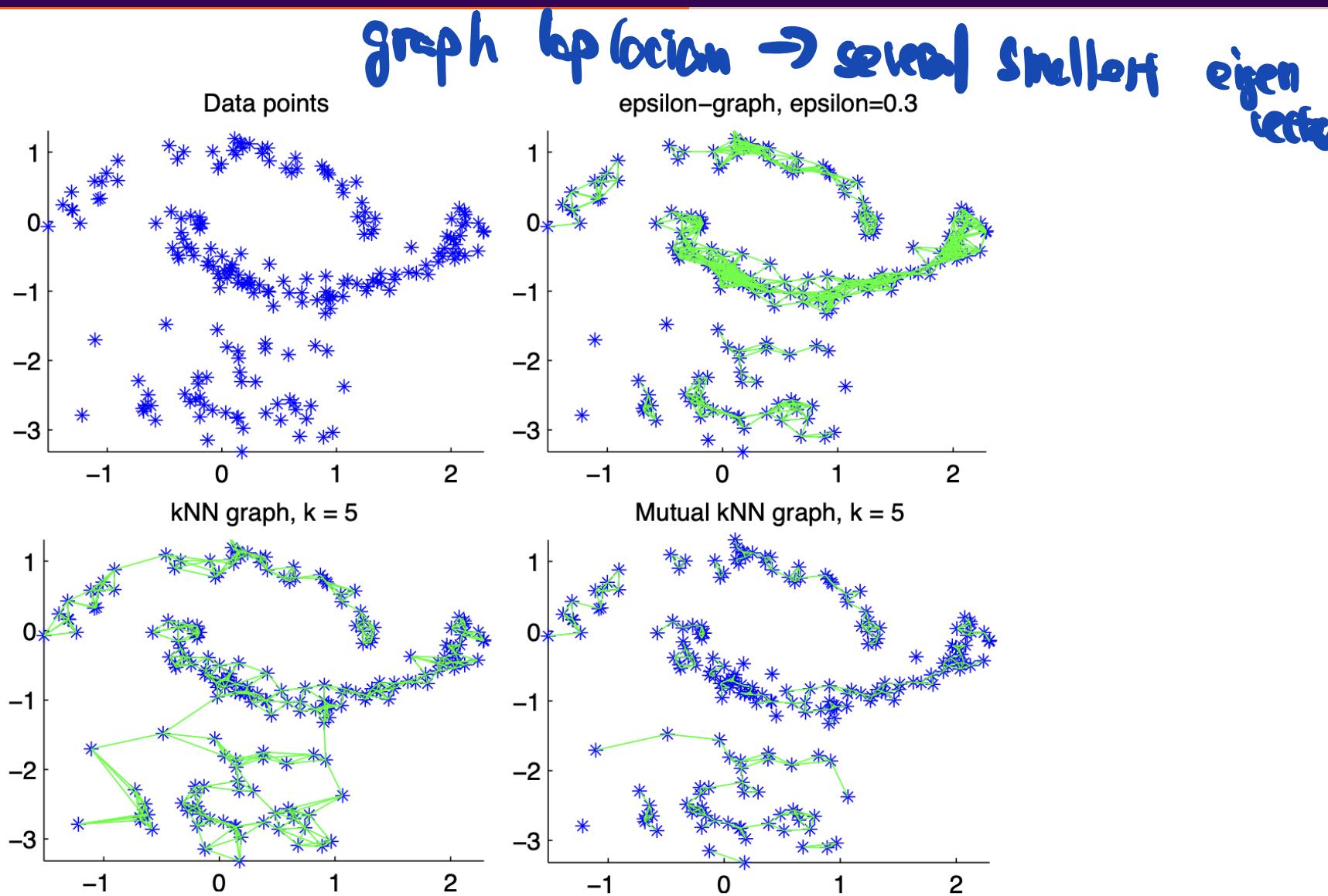
$$f_1 + f_2 + \dots + f_n = 0$$

( $f(i) = 1$  to all the data,  $f^\top L f = 0$ )

② Case2: If half the value of  $f_i$ , then the quadratic function will factor  
(the solution can be  $f(i) = 0$  to all data)

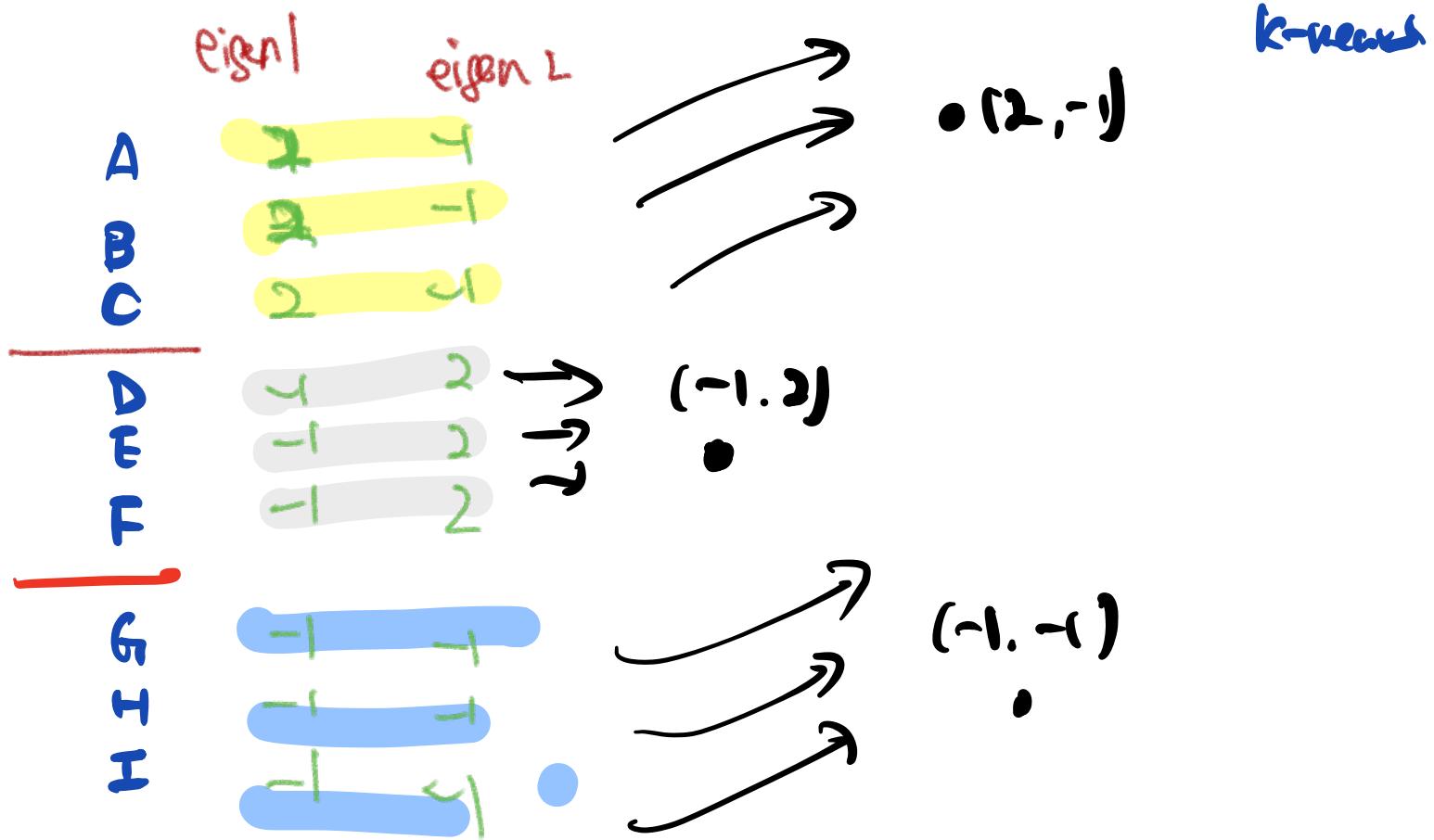
$$f_1^2 + f_2^2 + \dots + f_n^2 = 1$$

# Graph



# The procedure of spectral clustering.

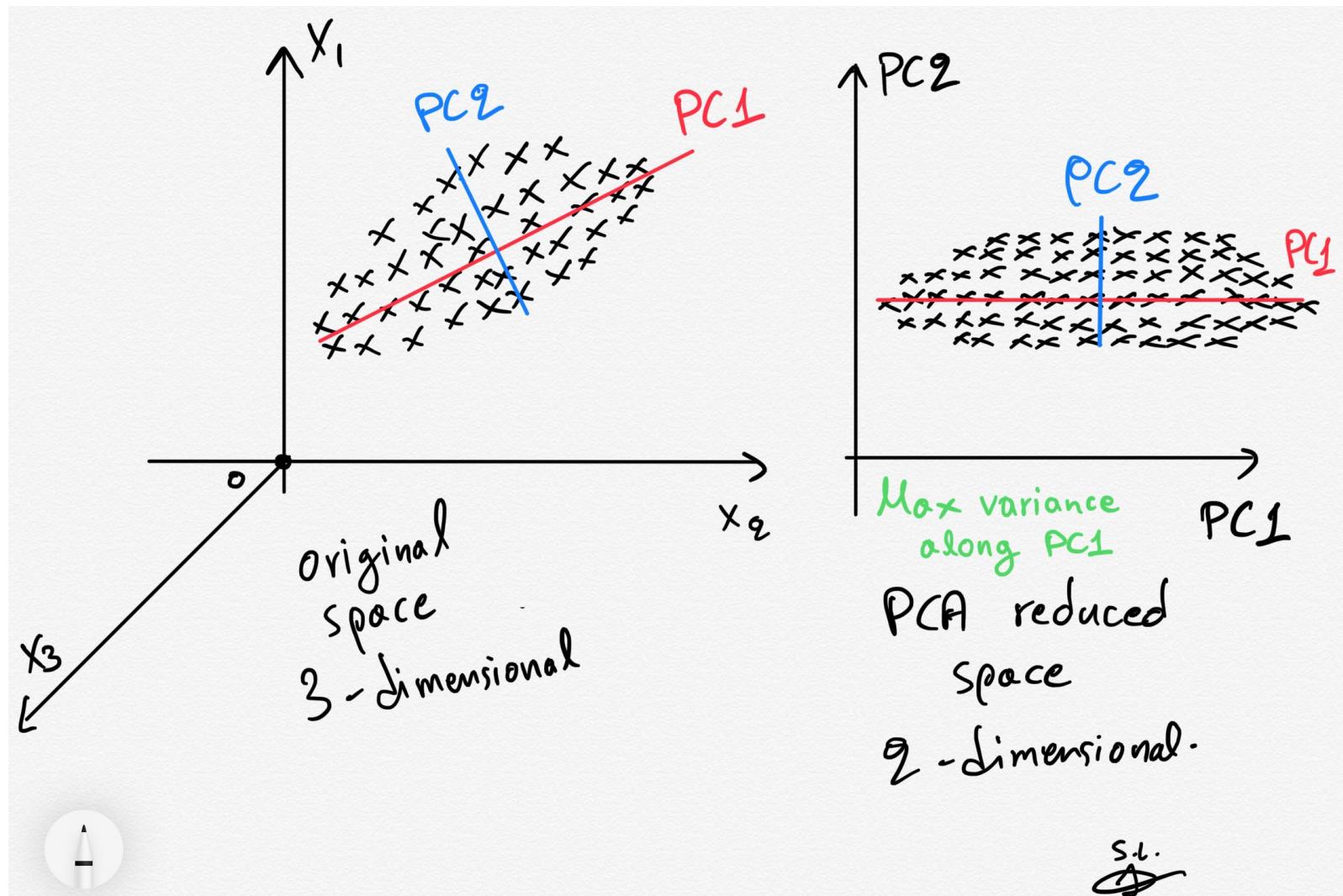
1. Construct a graph based on if two data are similar
2. Compute the matrix  $L$ .
3. Compute the eigenvectors (the smallest few eigenvalues) of  $L$
4. the cluster is based now. you can run a k-means



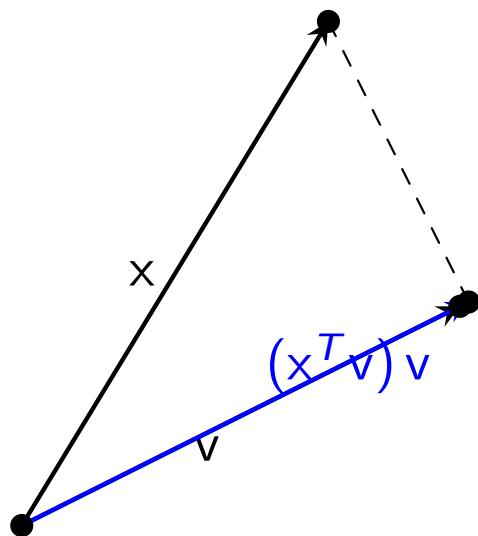
# Dimension Reduction

---

# Principal Component Analysis (PCA)



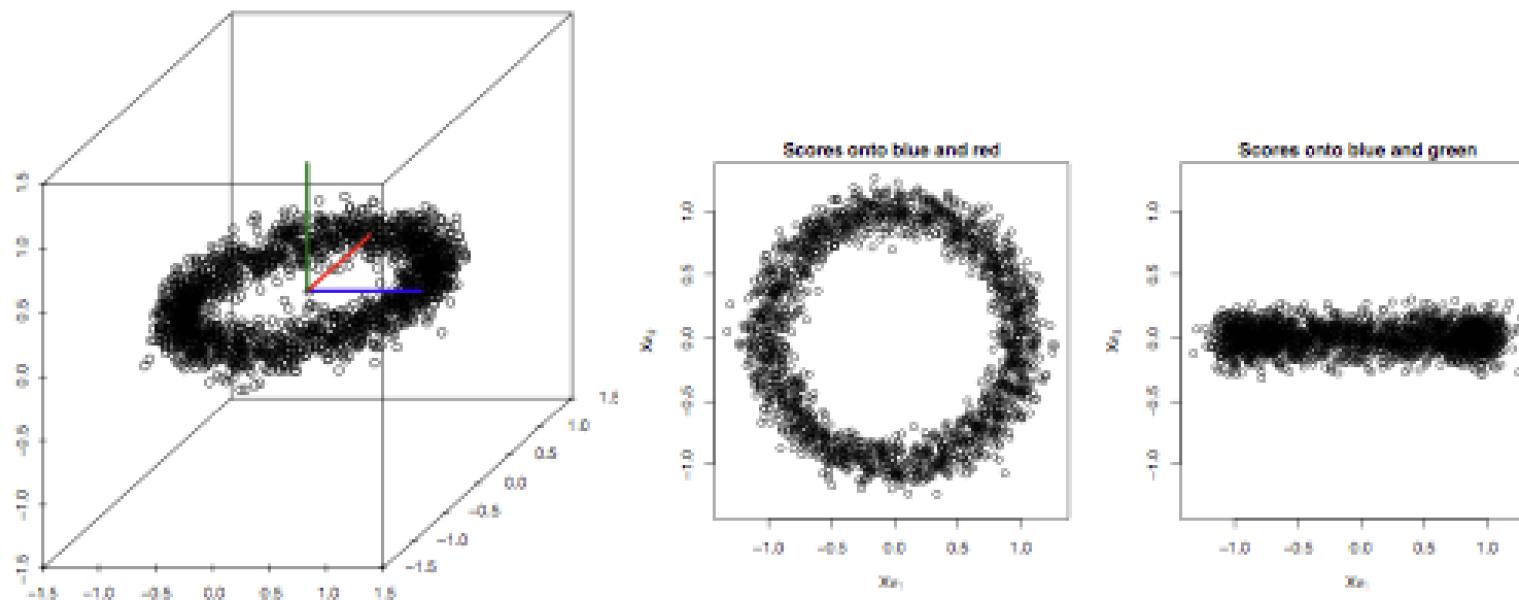
# Projection



- $x^T v \in \mathbb{R}$  : score
- $(x^T v)v \in \mathbb{R}^p$  : projection

# Not All Projection are the Same

Example:  $X \in \mathbb{R}^{2000 \times 3}$ , and  $v_1, v_2, v_3 \in \mathbb{R}^3$  are the unit vectors parallel to the coordinate axes



Not all linear projections are equal! What makes a good one?

# PCA: Preserve Most Information

We have  $n$   $d$ -dimensional data points  $x_1, x_2, \dots, x_n \in \mathbb{R}^d$  and a parameter  $k \in \{1, 2, \dots, d\}$ . We assume that the data is centered, meaning that  $\sum_{i=1}^n x_i = 0$ . (How to do that?)

**AIM.** Find directions that maximize the information preserved

The output of the method is defined as  $k$  orthonormal vectors  $v_1, v_2, \dots, v_k$  — the “top  $k$  principal components” — that maximize the objective function :

$$\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k (x_i \cdot v_j)^2.$$

---

**Question:** Why we want the principal components orthonormal?

## Review: Projection Under Orthonormal Basis

Let  $A = [v_1, \dots, v_k]$  where  $v_1, \dots, v_k$  are orthonormal. Remind. Least square solution:  $A\beta \approx b$ , then  $\beta = (A^\top A)^{-1} A^\top b$ . Then  $A\beta = A(A^\top A)^{-1} A^\top b$

Review. Orthonormal means  $A^\top A = I$

Check. Project  $b$  to  $\text{span}\{v_1, \dots, v_k\}$  means

$$\langle v_1, b \rangle v_1 + \langle v_2, b \rangle v_2 + \dots + \langle v_k, b \rangle v_k$$

# Matrix Formulation

**Matrix Formulation:** Define  $V \in \mathbb{R}^{d \times k}$  with columns  $v_1, \dots, v_k$ , representing the  $k$  principal components.

The total variance captured when projecting the data onto the subspace spanned by  $V$  is

$$\frac{1}{n} \|XV\|_F^2 = \text{tr}\left(V^T \left(\frac{1}{n} X^T X\right) V\right) = \text{tr}(V^T S V),$$

where  $S = \frac{1}{n} X^T X$  is the covariance matrix.

Note that  $\|A\|_F^2 = \text{tr}(A^T A)$ . For  $A = XV$ , we have:

$$\|XV\|_F^2 = \text{tr}((XV)^T (XV)) = \text{tr}(V^T X^T X V). \quad (\text{for } \text{tr}(AB) = \text{tr}(BA))$$

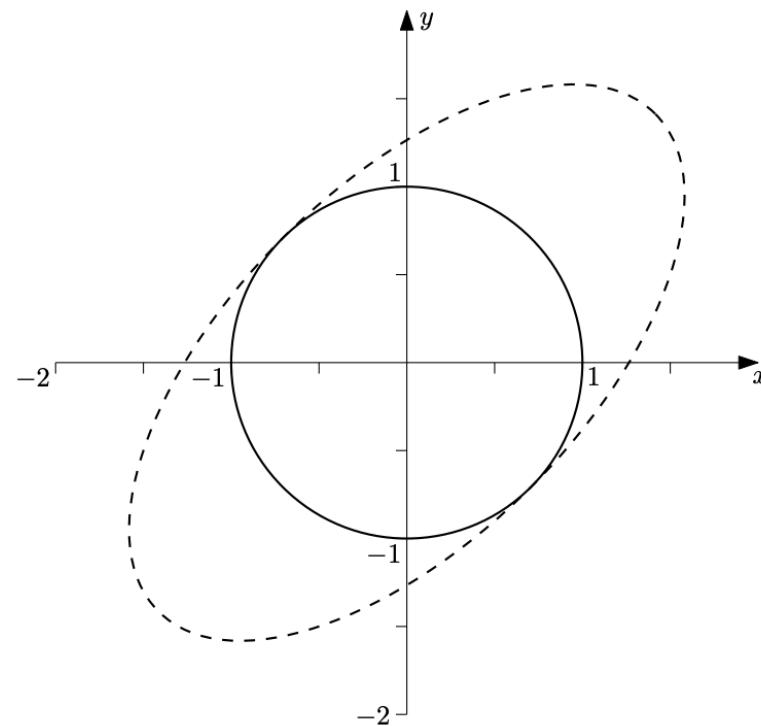
$$\max_{V \in \mathbb{R}^{d \times k}} \text{tr}(V^T S V) \quad \text{subject to} \quad V^T V = I_k.$$

# Matrix Formulation

---

# Covariance Matrix: Rotation on Principal Component

$$\begin{pmatrix} \frac{3}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{3}{2} \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}}_{\text{rotate back } 45^\circ} \cdot \underbrace{\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}}_{\text{stretch}} \cdot \underbrace{\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}}_{\text{rotate clockwise } 45^\circ}$$

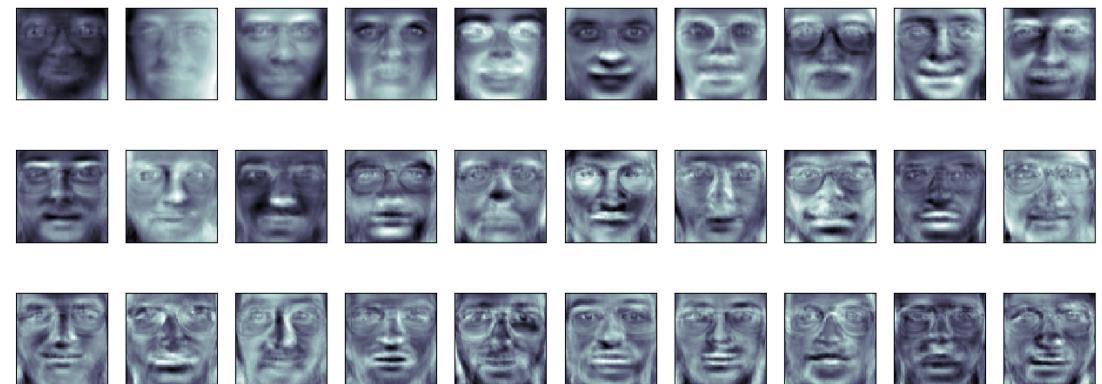


## PCA as Top Eigenvectors

PCA boils down to computing the  $k$  eigenvectors of the covariance matrix  $X^T X$  that have the largest eigenvalues.

---

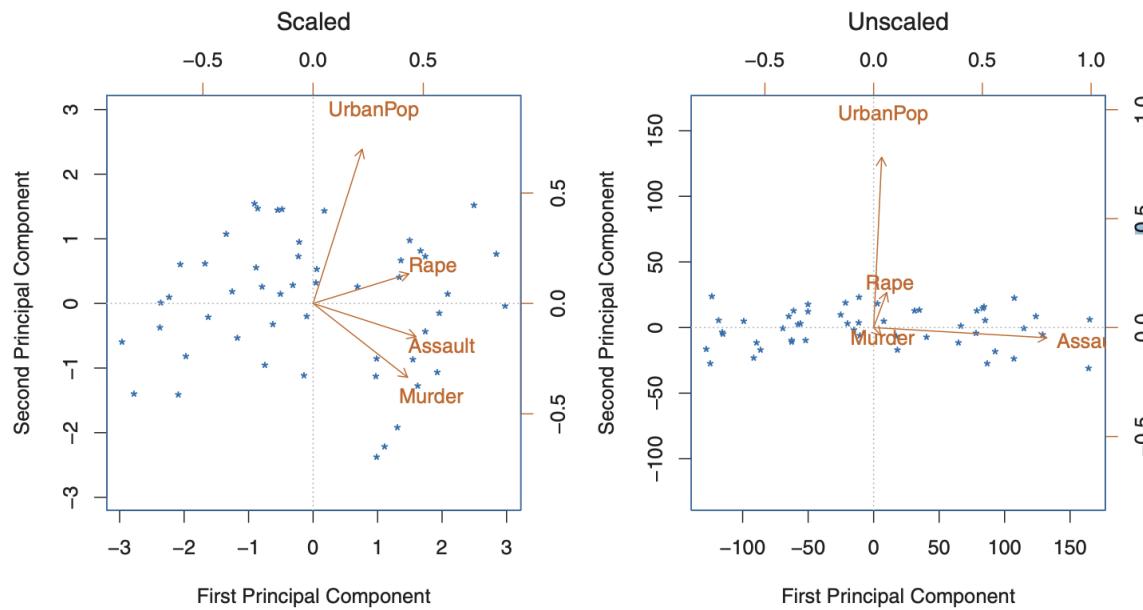
# Eigen-Face



---

The components (“eigenfaces”) are ordered by their importance from top-left to bottom-right. We see that the first few components seem to primarily take care of lighting conditions; the remaining components pull out certain identifying features: the nose, eyes, eyebrows, etc.

# Normalize Your Data



**Murder**, **Rape**, and **Assault** are reported as the number of occurrences per 100, 000 people, and **UrbanPop** is the percentage of the state's population that lives in an urban area. These four variables have variance 18.97, 87.73, 6945.16, and 209.5