Task1 : Clustering   Task 2 : feature extraction

IEMS 304 Lecture 8: Unsupervised Learning

Yiping Lu
yiping.lu@northwestern.edu

Industrial Engineering & Management Sciences
Northwestern University

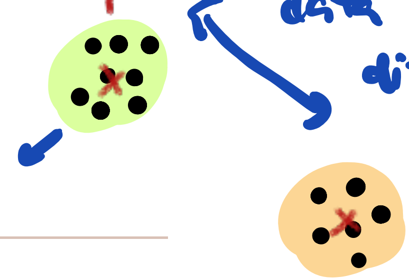
Clustering $\{(x_i)\}_{i=1}^{n}$ → Classification.

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

Classification Result.

use 1 data to represent the cluster you will select the mean

between class data are different.

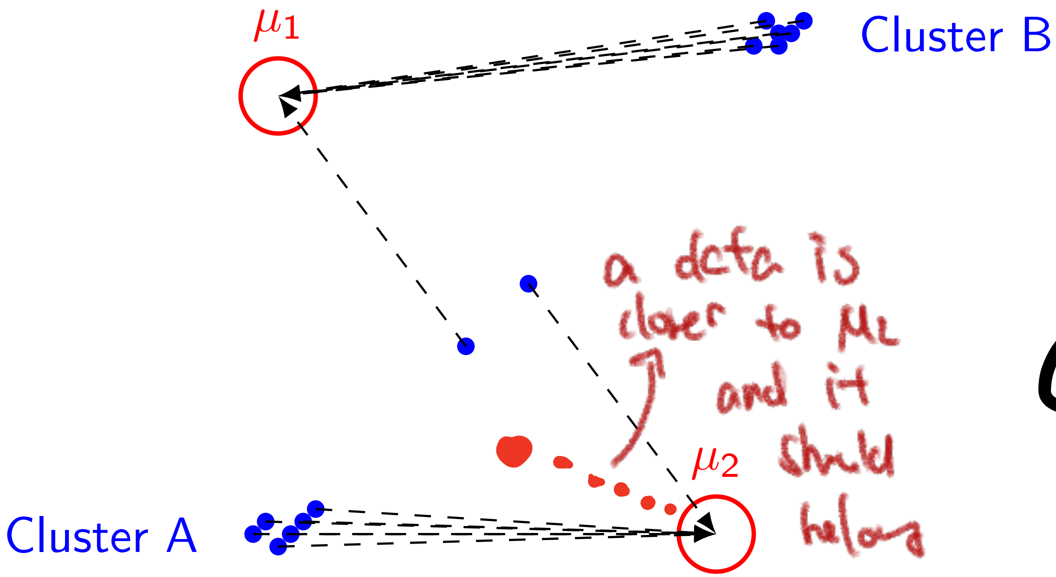


k-means

k is the number of classes.

in class are similar

Iteration 1: Initialization & Forced Assignment



a data is closer to μ_2 and it should belong to μ_2.

① randomly initialize the k-means.

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

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].

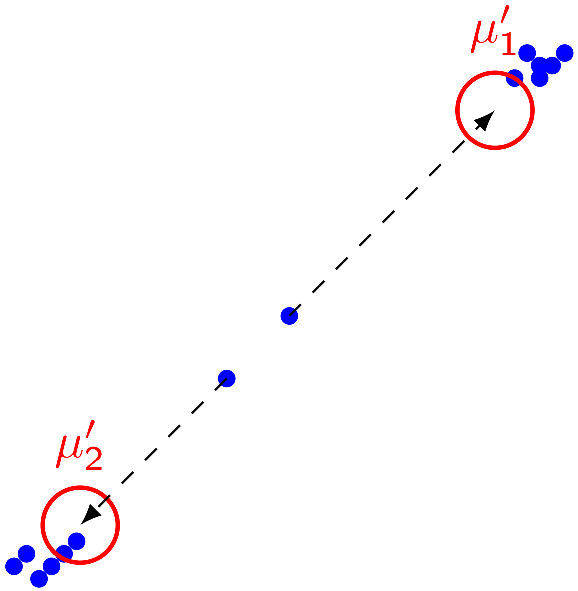
Updated centroids (computed as the mean):

$$\mu_1' = \left(\tfrac{30+2.5}{7}, \tfrac{30+2.5}{7}\right) \approx (4.643,\ 4.643)$$

$$\mu_2' = \left(\tfrac{6.3+3}{7}, \tfrac{6.3+3}{7}\right) \approx (1.329,\ 1.329)$$

iterative.

Iteration 2: Reassignment



③ update the cluster mean .

cluster mean

classification result .

Reassignment (Iteration 2):

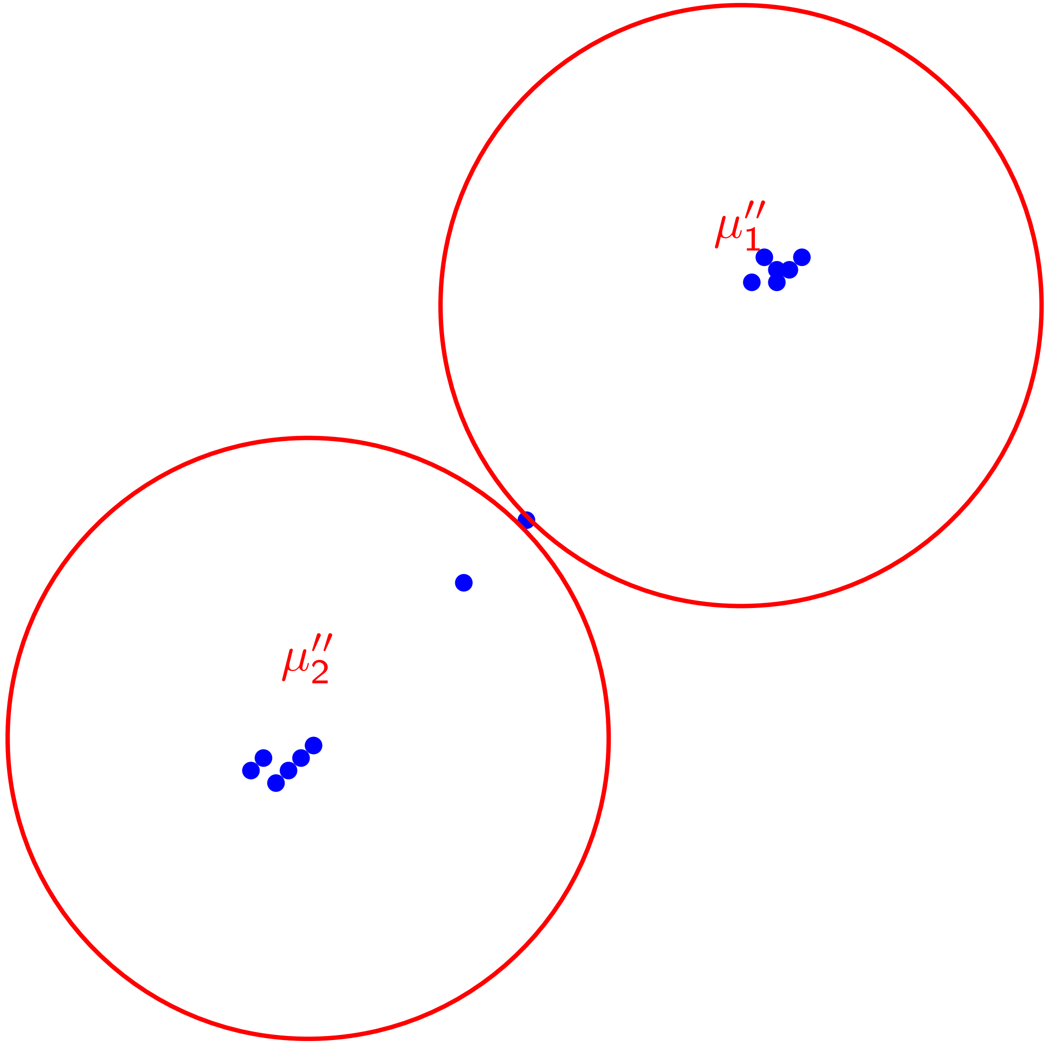
- $(2.5, 2.5)$ switches from μ_1 to μ'_2 (closer to $(1.329, 1.329)$).
- $(3, 3)$ switches from μ_2 to μ'_1 (closer to $(4.643, 4.643)$).

New centroids:

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

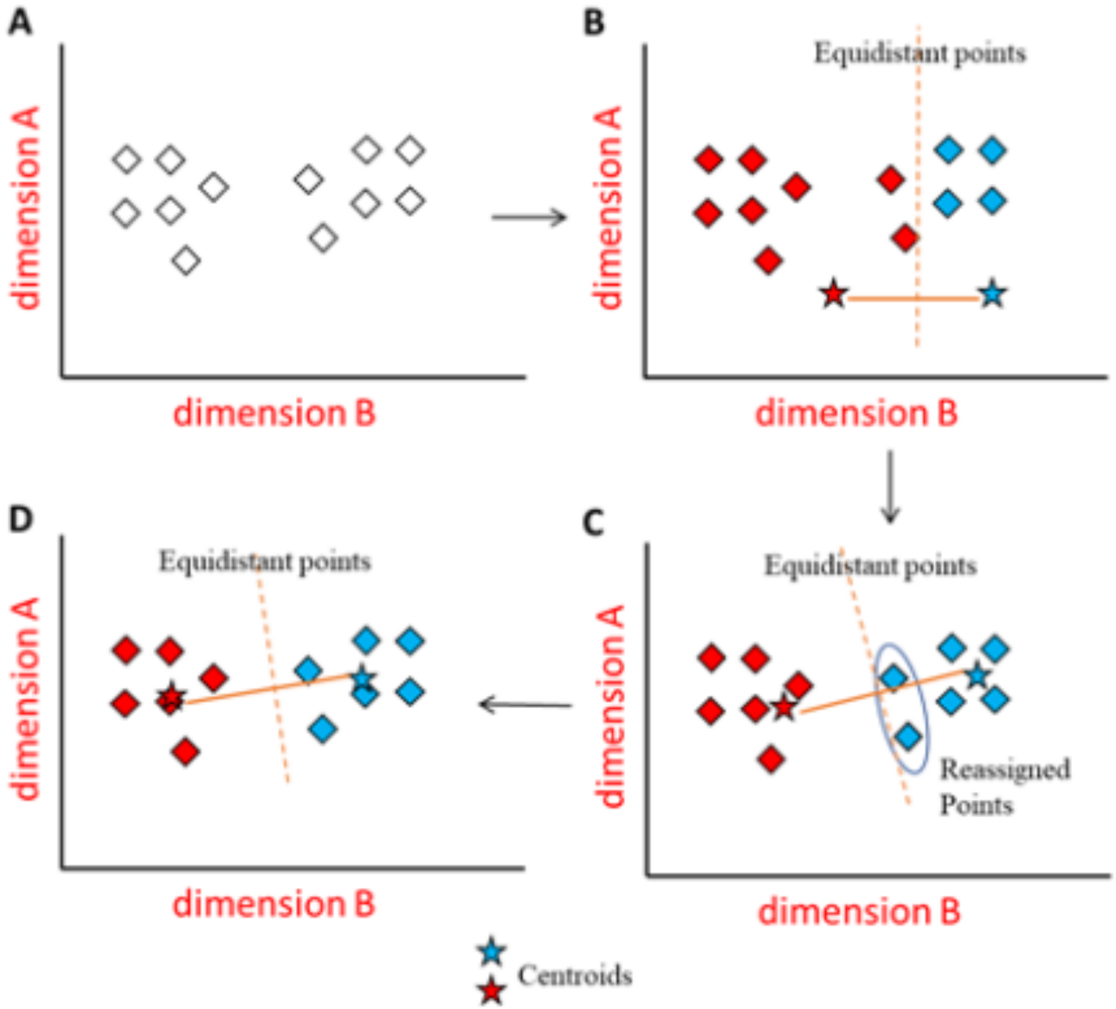
$$\mu''_2 = \left(\frac{6.3+2.5}{7}, \frac{6.3+2.5}{7}\right) = \left(\frac{8.8}{7}, \frac{8.8}{7}\right) \approx (1.257,\ 1.257)$$

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


A
dimension A
dimension B
B
Equidistant points
dimension A
dimension B
D
Equidistant points
dimension A
dimension B
C
Equidistant points
dimension A
Reassigned
Points
dimension B
Centroids

$k-$means as Optimization

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

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

sum over all the data in cluster C_j

How you cluster your data

the mean of the cluster

$k-$means as alternating direction optimization algorithm

fix μ_j and update C_j

- ❒ **Assignment:** Assign each x to its nearest μ_j (~~minimizes distance~~).

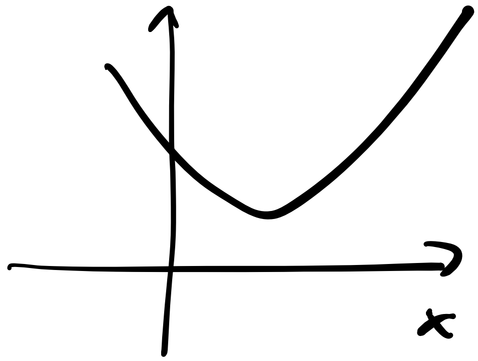
fix C_j and update μ_j.

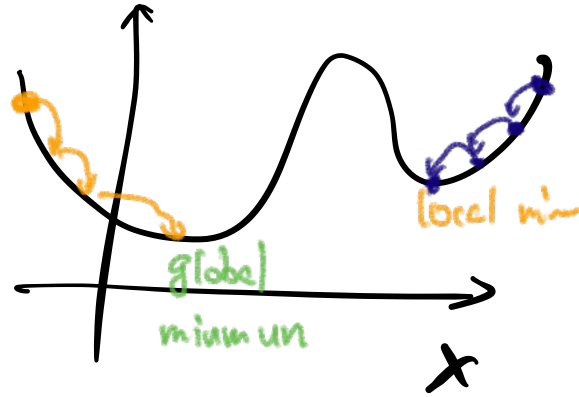
- ❒ **Update:** Recompute μ_j as the mean of C_j (~~minimizes variance~~).

---

① why we select the mean as the respresentation

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





Convex

linear Regression

Non-Convex

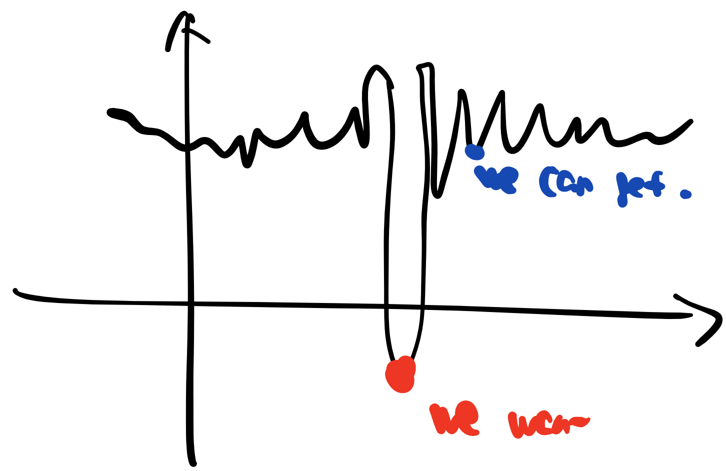
k-means

Neural Networks

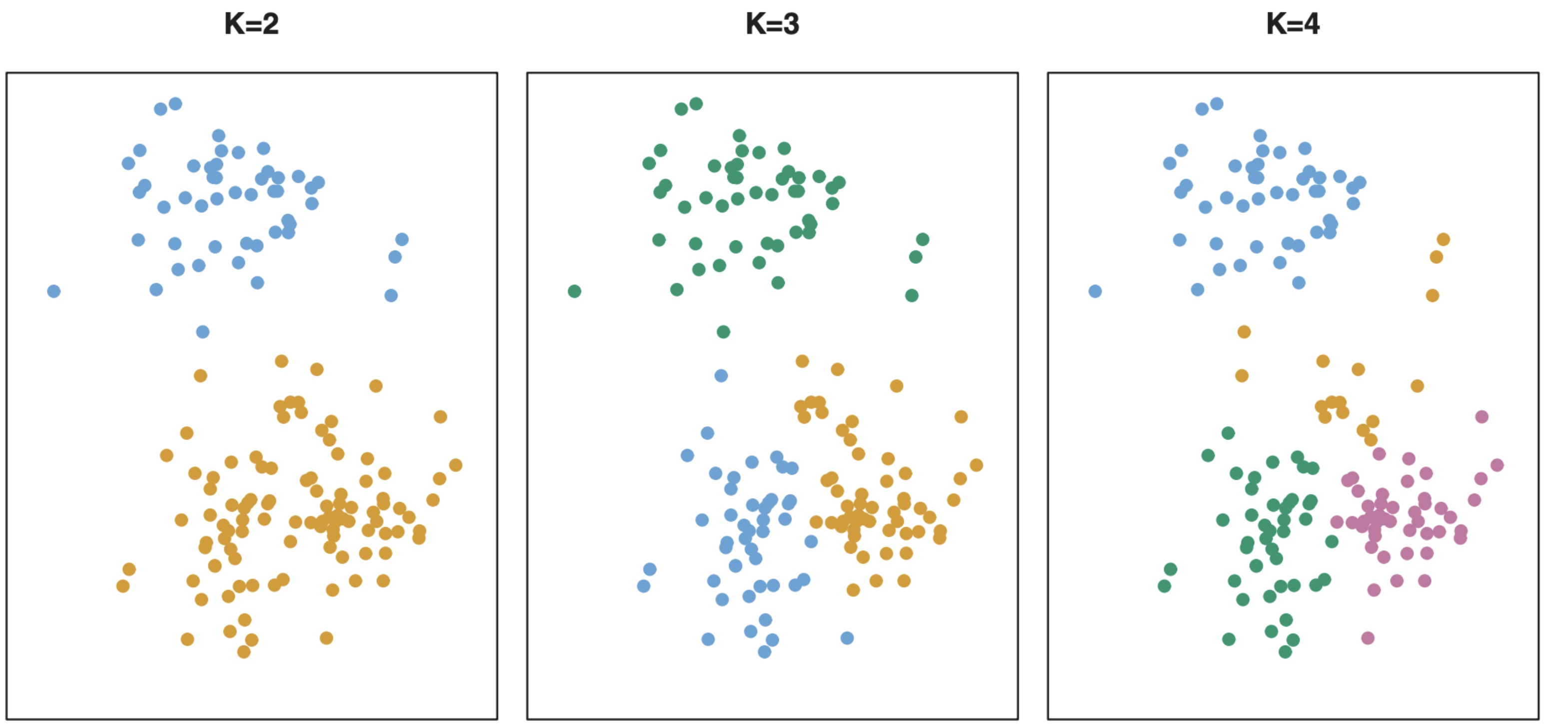
k-means ++.

with high prob.

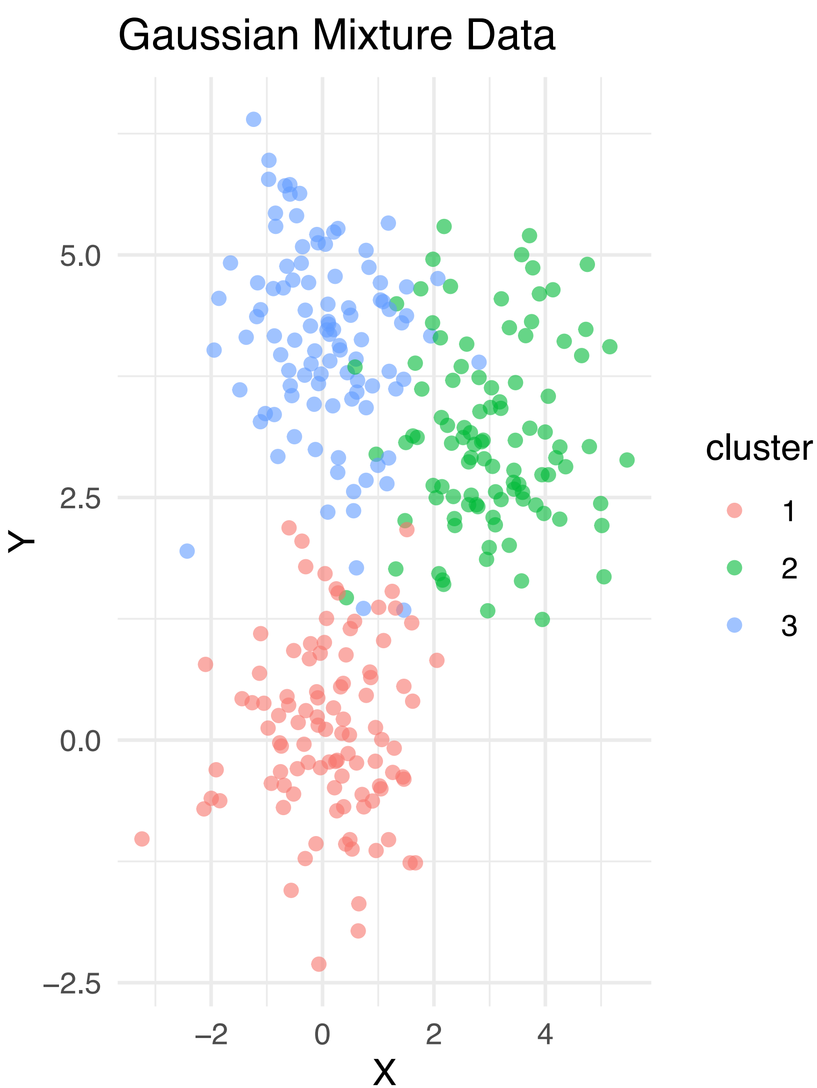
goes to global minimum.

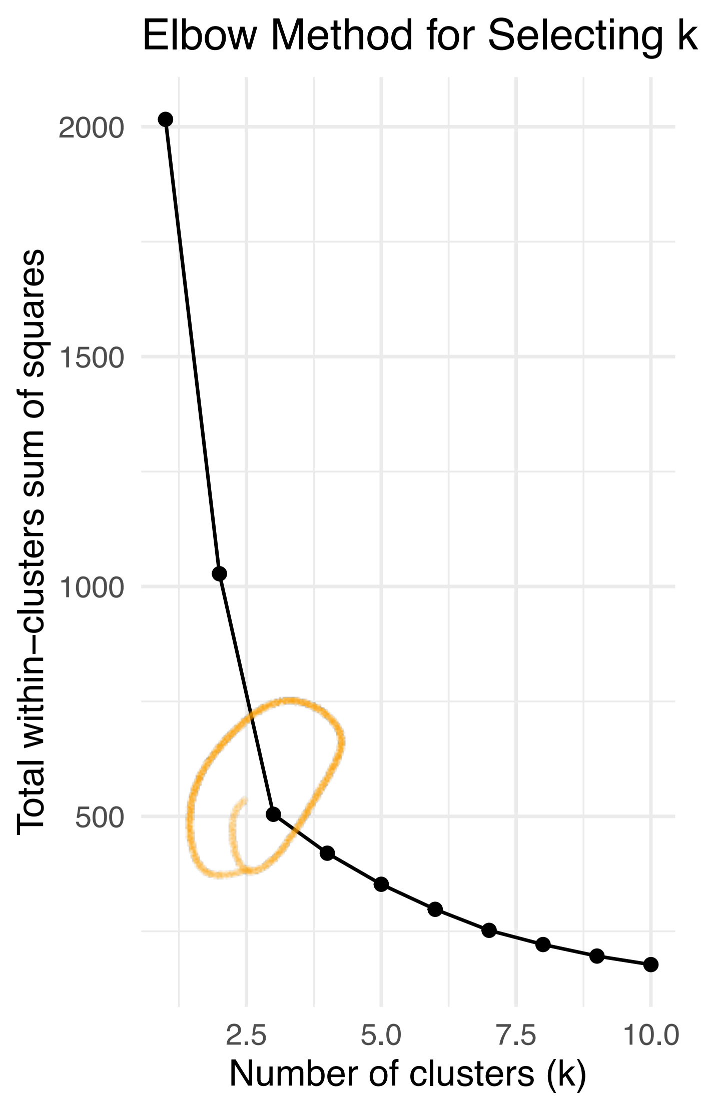


Wrong k can be Problematic



How to Select k: Elbow Effect





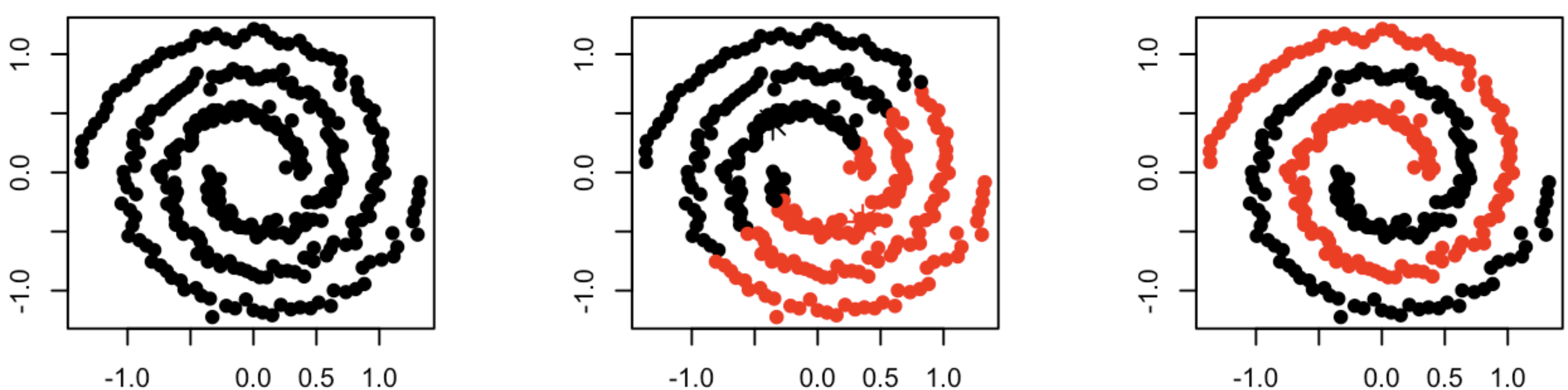
Spectral Clustering

Spectral Clustering

The failure here is because we just use mean to represent a cluster

⇒ we assume the clusters are convex

K-means | Spectral clustering



Spectral Clustering

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

Consider the Dirichlet form,

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

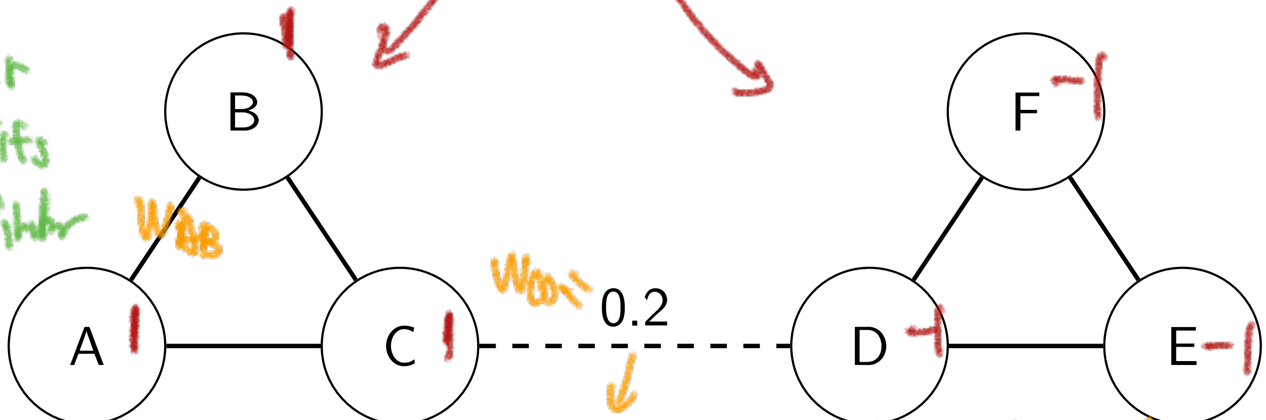
where L is the graph Laplacian defined as $L = D - W$ (where D is the degree matrix).

minimizing Dirichlet form.

e.g. Connect your data with its k-nearest neighbor

$w = h(-$ distance between data$)$

some function



assign a weight to the edge,

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

What would happen if we minimizing this form?

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 coefficient

$$\Downarrow$$

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

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

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

$$A^T = A.$$

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

Consider the Dirichlet form:

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

- terms involving $f(i)^2$:

$$\frac{1}{2}\Big(\sum_{i,j} w_{ij}f(i)^2 + \sum_{i,j} w_{ij}f(j)^2\Big)$$

$$= \sum_i f(i)^2 \sum_j w_{ij} = \sum_i d_i f(i)^2.$$

- The cross term simplifies to:

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

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

At the same time,

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

Understanding the Dirichlet Form

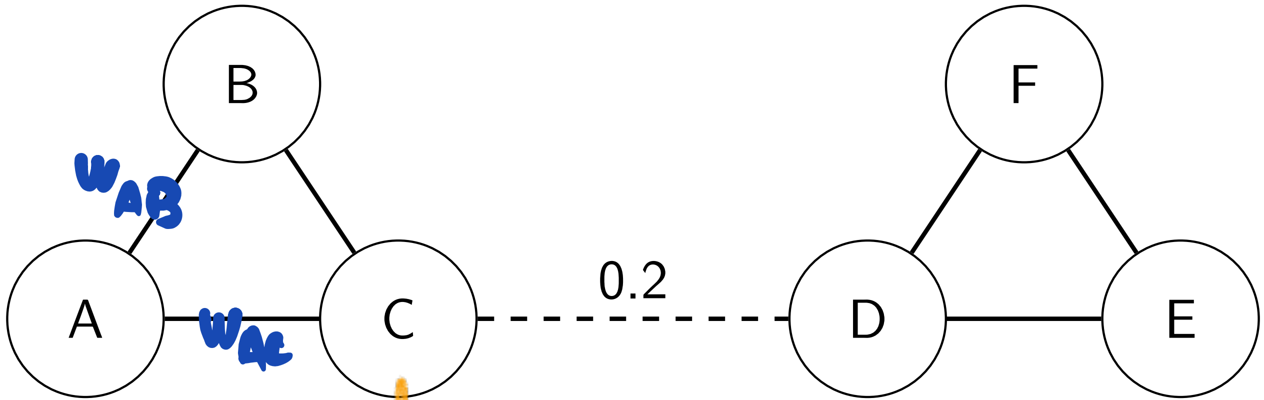
Definition

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



$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & & w_{AB} & w_{AC} & \\ B & w_{AB} & & w_{BC} & \\ C & w_{AC} & w_{BC} & & w_{CD} \\ D & & & w_{CD} & \\ E & & & & \\ F & & & & \end{array}$$

0.2

$d_C = w_{AC} + w_{BC} + w_{CD}$

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

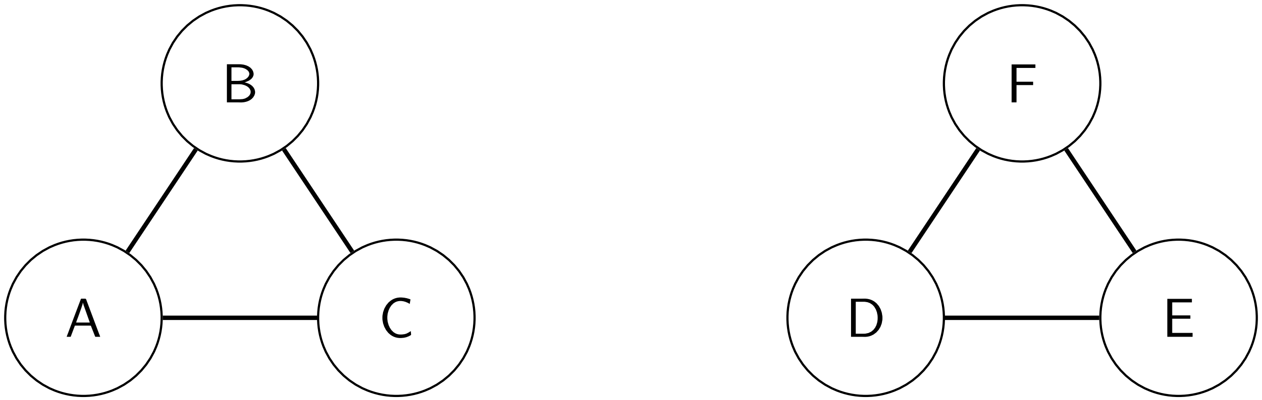
$$\begin{bmatrix} d_A & & & & & \\ & d_B & & & & \\ & & d_C & & & \\ & & & d_D & & \\ & & & & d_E & \\ & & & & & d_F \end{bmatrix}$$

Step 2: Compute the Graph Laplacian

$$L = D - W$$

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

Computing the Graph Laplacian



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

→ Closed form!!! is the smallest eigenvector!!

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

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

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

① Case1 Do No classification.

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

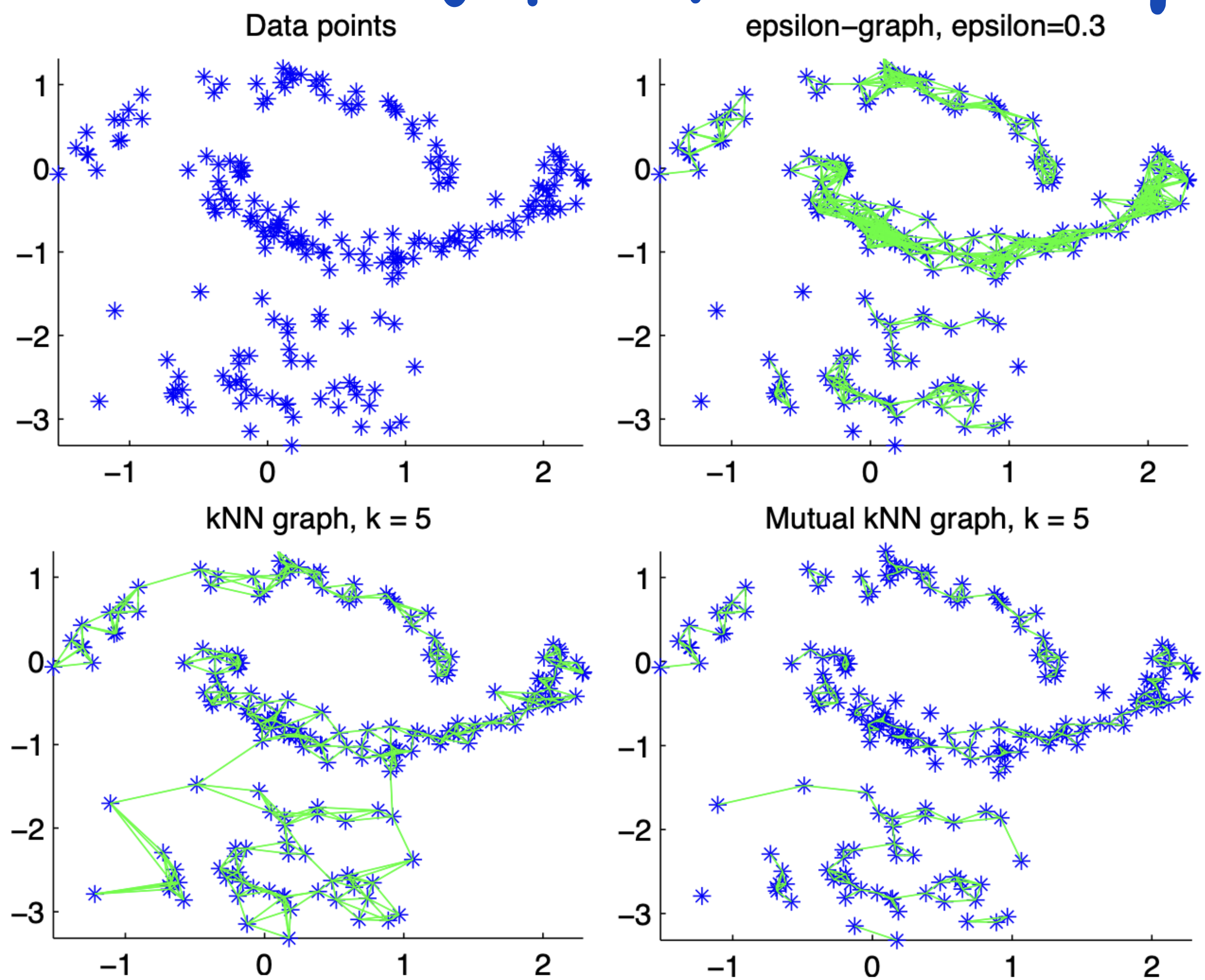
② Case2 If half the value of **f**, then the quadratic function will greater

(the solution can be f(i) = 0 to all data)

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

Graph

graph laplacian → several smallest eigen vectors



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 eigenvectors) of L

4. the cluster is clear now. you can run a k-means

	eigen 1	eigen 2
A	2	-1
B	2	-1
C	2	-1
D	-1	2
E	-1	2
F	-1	2
G	-1	-1
H	-1	-1
I	-1	-1

$\bullet (2, -1)$

$(-1, 2)$ $\bullet$

$(-1, -1)$ $\bullet$

Dimension Reduction

Framework of Dimension Reduction

Data: $X \in \mathbb{R}^n$ feature $\overset{\phi(x)}{\in} \mathbb{R}^m$

(We may consider m is much smaller than)

e.g. in the face dataset.

n: the number of pixel of face image

m: $m=2$: poses.

$\phi(x)$ should include most of the information in your data x. (compression).

$$\underset{\text{data}}{X} \xrightarrow[\text{compress the information}]{} \underset{\text{low dimension feature}}{\phi(x)} \xrightarrow{\text{reconstruction}} r(\phi(x))$$

we aim: $r(\phi(x)) \simeq x$ auto-encoder

encoder decoder.

objective function $x_1 \dots x_n$.

$$\min_{r, \phi} \sum_{i=1}^{n} \| r(\phi(x_i)) - x_i \|^2$$

If r, ϕ are Neural Networks.

$\Rightarrow$ auto-encoder

If r, ϕ are linear function

$\Rightarrow$ Principal Component Analysis

close form!!

(PCA)

Review: Projection.

X: linear projection.

$$x \xrightarrow{x^T u} \text{real number}$$

$x \in \mathbb{R}^n$

$u \in \mathbb{R}^n$

real number $\in \mathbb{R}^1$

$x^T u$: is the inner/dot product

Aim: X is the data,

$u_1 \cdots u_m$: (encoding)

$$X \to \underbrace{(\underbrace{X^T u_1}_{\text{scalar}}, X^T u_2, \cdots, X^T u_m)}_{\mathbb{R}^m}$$

Decoding $u_1' \cdots u_m' \in \mathbb{R}^n$

$$\underbrace{(\underbrace{a_1}_{\text{scalar}}, \cdots, a_m)}_{\mathbb{R}^m} \to a_1 u_1' + a_2 u_2' + \cdots + a_m u_m'$$

Close-form Solution!!!!!!

$u_i = u_i'$

$u_1 \cdots u_m$ are the top eign ctors of $X X^T$ (n×n matrix)

covariance

$$X = [x_1 \cdots\cdots x_k] \in \mathbb{R}^{n \times k}$$

k-data: x_i, $x_i \in \mathbb{R}^n$

XX^T is a matrix of size $\mathbb{R}^{n \times n}$

Fact 1 XX^T is a symmetric matrix

Fact 2 $u_1 \cdots u_m$ are orthogonal.

$$u_i^T u_j = 0$$

$U = [u_1 \cdots u_m]$

Question! What is the projection of data matrix X to the space spanned by U

$$\Rightarrow \underbrace{(U^T U)^{-1}}_{= I_m} U^T x = U^T x$$

Let's use the info that U is orthogonal

$$\begin{cases} u_i^T u_i = 1 \\ u_i^T u_j = 0 \end{cases}$$

$$U^T U = \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_m^T \end{bmatrix} \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \cdots & \cdots & 0 \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & \cdots & \cdots & 1 \end{bmatrix} = I_m$$

PCA

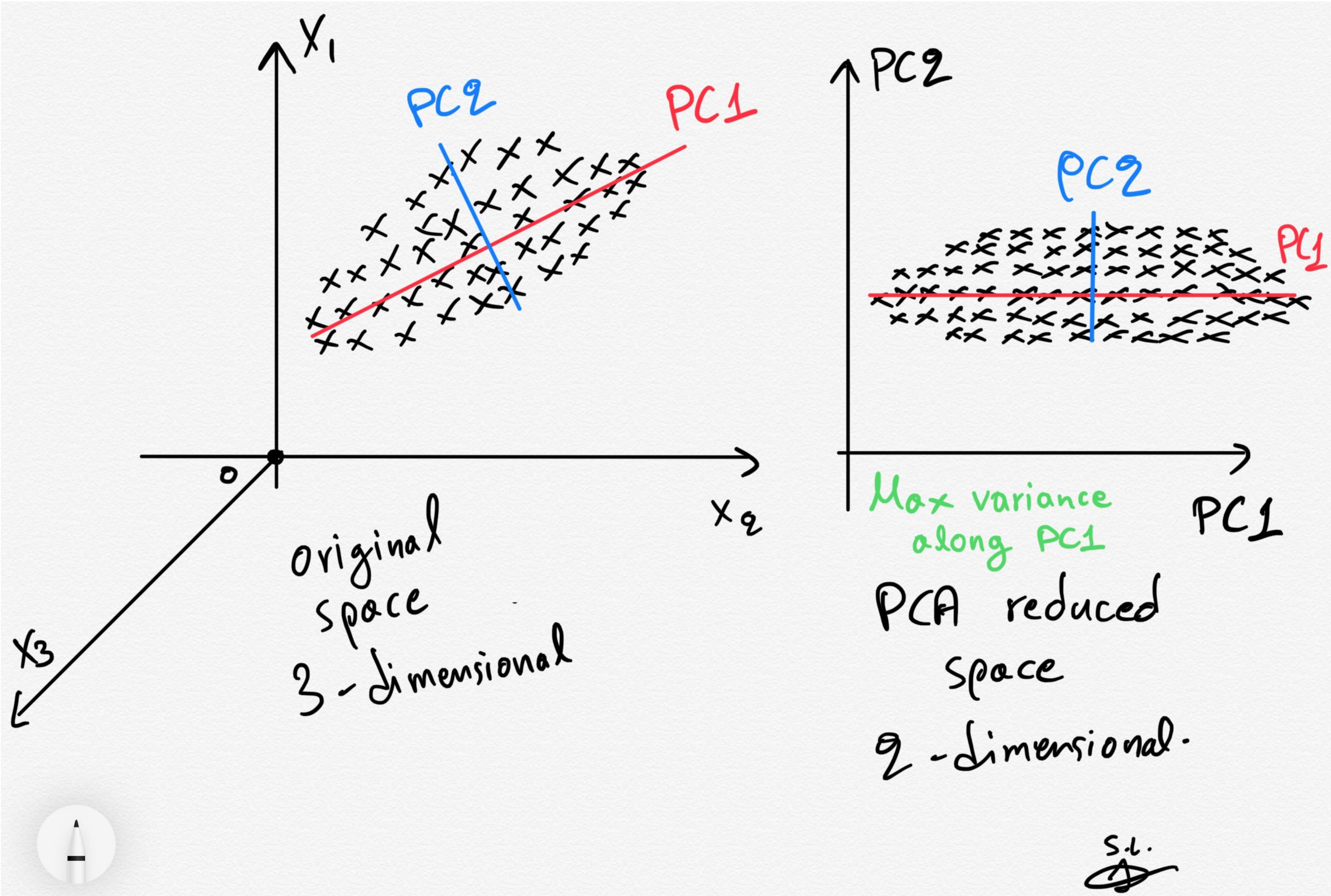
;

the projection that preserves most of the information of your data is the projection to the top eigenspaces of the covariance matrix,

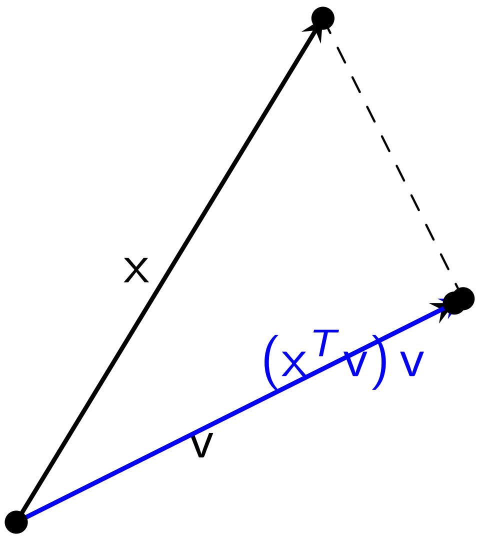
largest!

number of basis → bias- variance .

Principal Component Analysis (PCA)



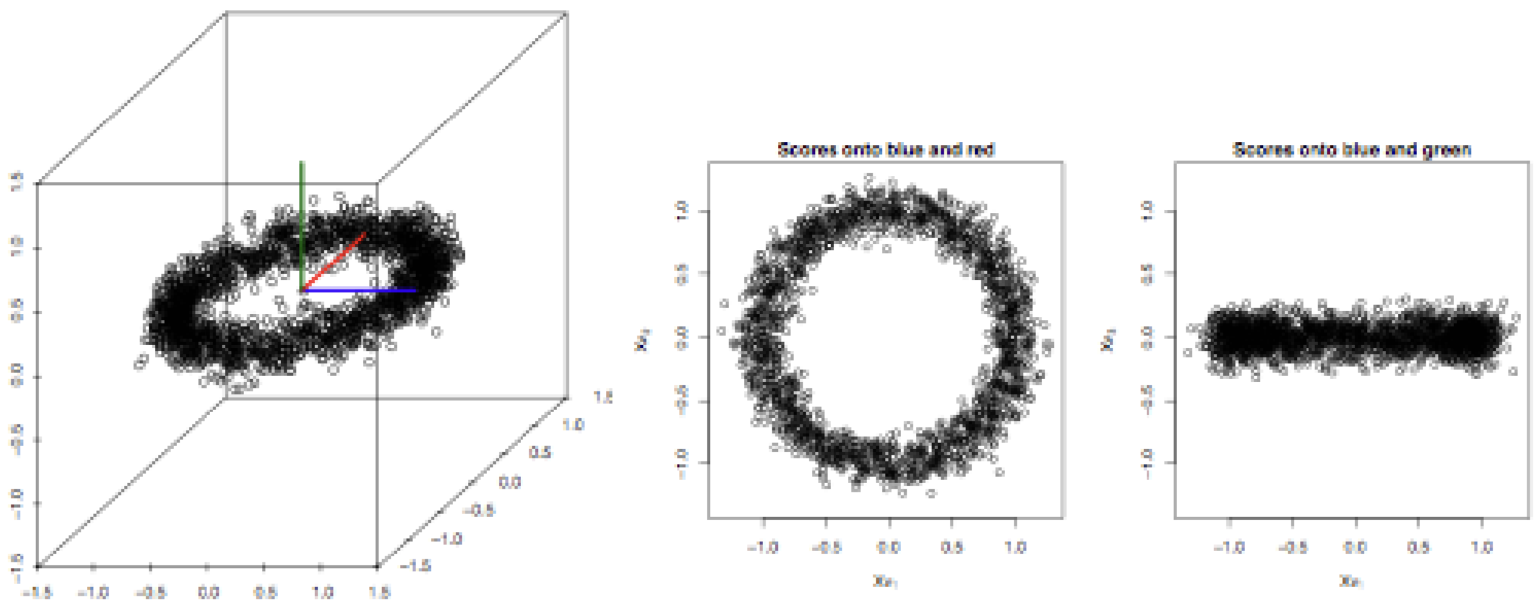
Projection



- $x^T v \;\in\; \mathbb{R}$: 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 $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n \in \mathbb{R}^d$ and a parameter $k \in \{1, 2, \ldots, d\}$. We assume that the data is centered, meaning that $\sum_{i=1}^{n} \mathrm{x}_i = 0$. (How to do that?)

$\boxed{v^T \text{ matrix } v}$ ⇒ quadratic function of v

AIM. Find directions that maximize the information preserved

The output of the method is defined as k orthonormal vectors $\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_k$ — the "top k principal components" — that maximize the objective function :

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} (\mathrm{x}_i \cdot \mathrm{v}_j)^2.$$

$$\underbrace{(X^T v)^T (X^T v)}_{\text{sum of square}} = v^T X X^T v$$

max the variance of the data after the projection

Question: Why we want the principal components orthonormal?

(Or I preserve the most varying direction of my data)

Review: Projection Under Orthonormal Basis

Let $A = [v_1, \cdots, v_k]$ where $v_1, \cdots, v_k$ 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 $\mathrm{span}\{v_1, \cdots, v_k\}$ means

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

Matrix Formulation

Matrix Formulation: Define $V \in \mathbb{R}^{d\times k}$ with columns $\mathrm{v}_1, \dots, \mathrm{v}_k$, representing the k principal components.

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

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

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

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

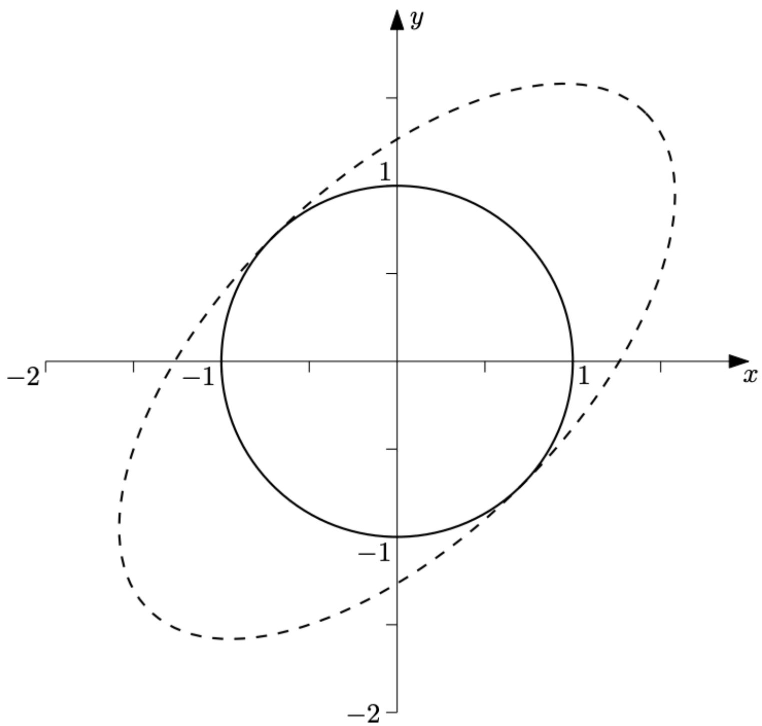
$$\|XV\|_F^2 = \mathrm{tr}((XV)^T(XV)) = \mathrm{tr}(V^TX^\top XV). \qquad (\text{for } \mathrm{tr}(AB) = \mathrm{tr}(BA))$$

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

Matrix Formulation

Covariance Matrix: Rotation on Principal Component

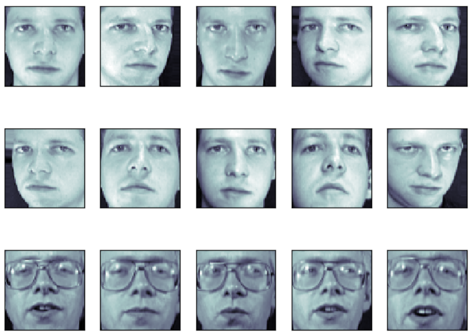
$$\begin{pmatrix} \frac{3}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{3}{2} \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}}_{\text{rotate back } 45^\circ} \cdot \underbrace{\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}}_{\text{stretch}}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

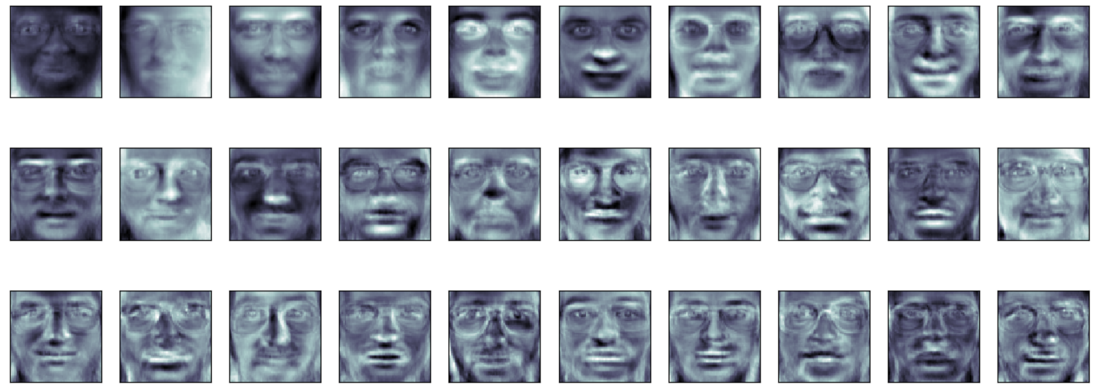


PCA as Top Eigenvectors

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

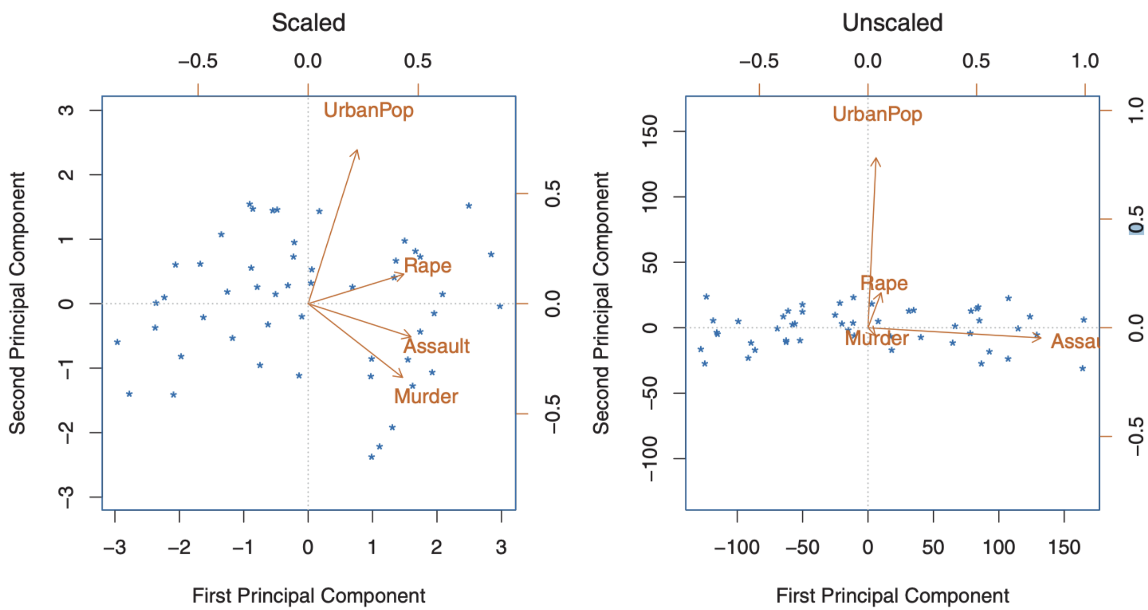
Eigen-Face





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