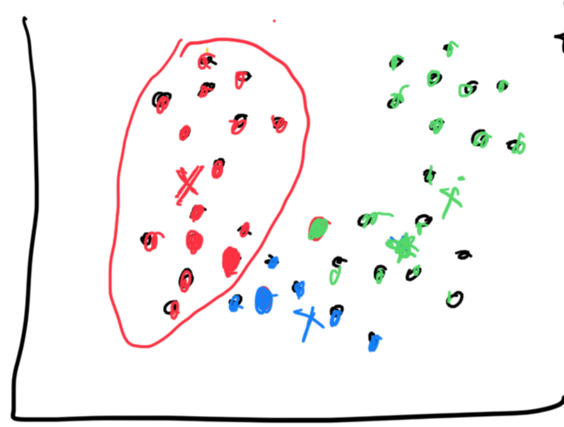
Unsupervised learning

clustering

latent = hidden

→ Finding hidden structure in the data.



$K = 3$

$\|x_i - x_j\|_2$

Euclidean Distance

→ 3 , 2

Cluster 1 →

7 , 4

3 , 1

1 , 8

Mean $\left(\frac{11}{3}, \frac{13}{3}\right)$

Applying k-means to non-vector data

$d(x_i, x_j)$

Medoid → A point which is closest to all other points.

k-medoid

Computational complexity of k-means.

$O(N K)$

$+ O(NK) = O(NK)$

$[N_1]$ $O(N_1^2)$

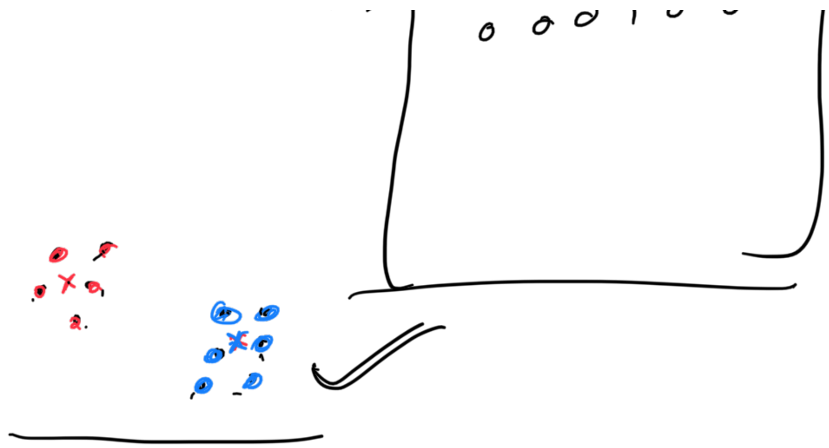
Stopping criteria:

Stop when cluster centers are not changing.

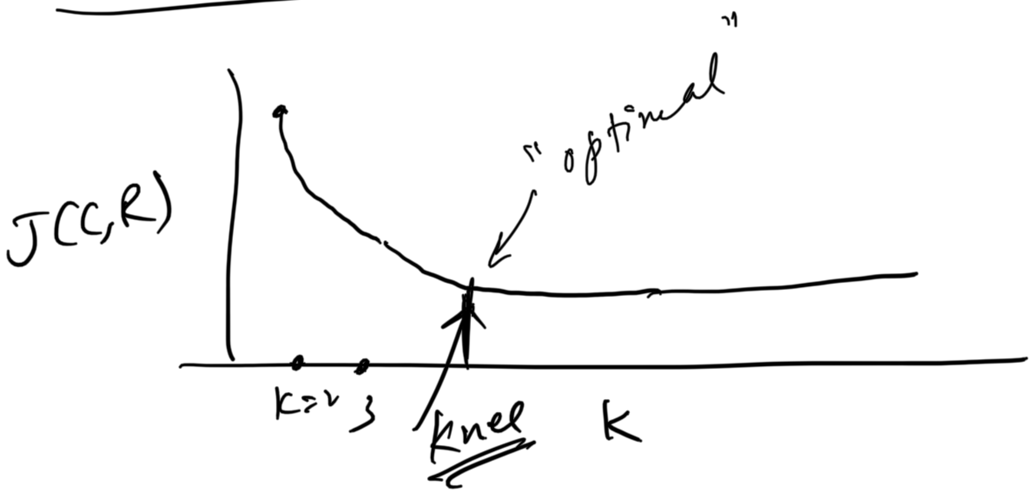
How to choose k

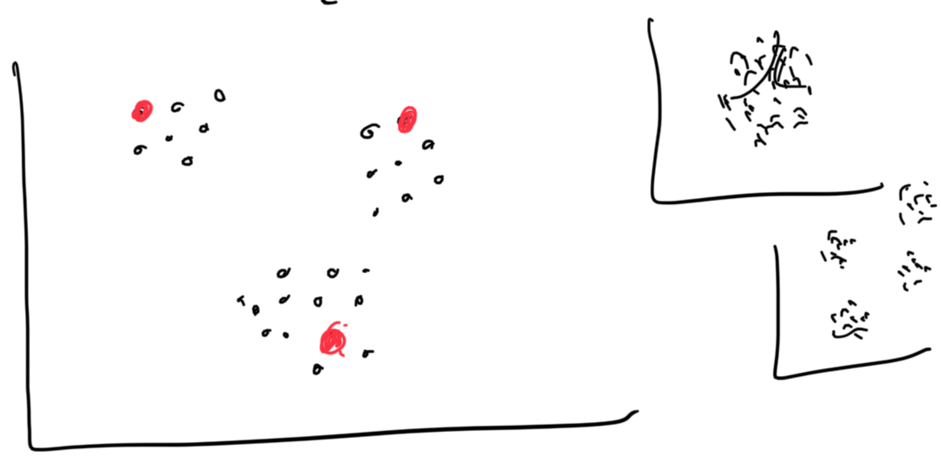
R - cluster assignment matrix

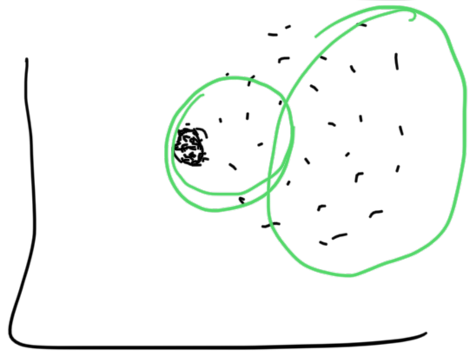
$N \times K$ → $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$



Can use $J(C,R)$ for stopping.



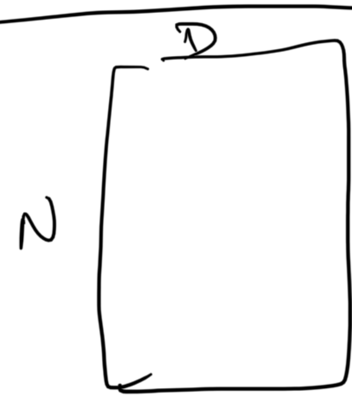




Hard clustering

Spectral Clustering

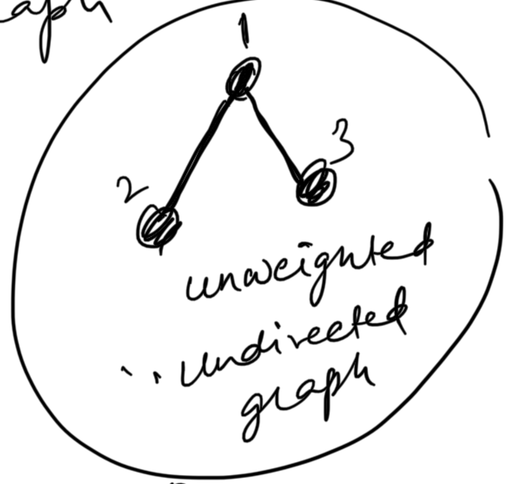
X



① $S_{N \times N}$

$$S_{ij} = \underline{sim}(\underline{x_i}, \underline{x_j})$$

③ Construct a graph



unweighted undirected graph

Adjacency matrix , $\mathbf{W}_{N \times N}$

If N – # of vertices (or nodes)

$$W = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 0 \\ 3 & 1 & 0 & 0 \end{array} \qquad W_{ij}$$

symmetric

unweighted directed graph

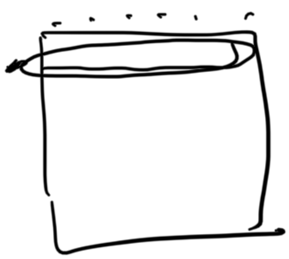
$$W = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \end{array}$$

need not be symmetric.

$$W = \begin{bmatrix} 0 & 0.7 & 4 \\ 0.7 & 0 & 0 \\ 4 & 0 & 0 \end{bmatrix}$$

$$S \longrightarrow W$$

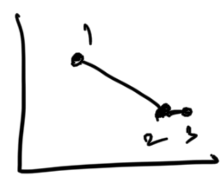
$$W_{ij} = \begin{cases} \underline{\text{Sim}(x_i, x_j)} & \text{if } x_i \text{ is nearest neighbor of } x_j \\ 0 & \text{otherwise} \end{cases}$$



$$W_{ij} = \begin{cases} 1 & \text{if } x_i \text{ is in the } \underline{k}\text{-nearest neighborhood of } x_j \\ 0 & \end{cases}$$

W will not be necessarily symmetric

$$\boxed{S + S^T}$$



Clustering $\longrightarrow$ Partition X into k clusters
(Hard)

<u>Graph</u> **W** $\rightarrow$ finding k cuts in this graph

$$\frac{1}{2}\sum_{k=1}^{K} \boxed{W(A_k, \bar{A}_k)}$$

all the nodes in cut k

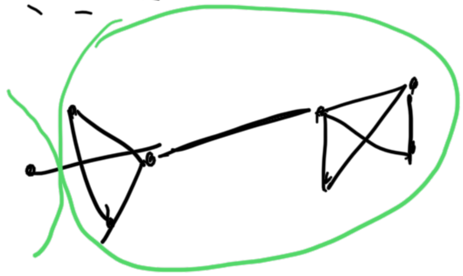
all the nodes ~~in~~ not in cut k

$$W(A_k, \bar{A}_k) = \text{sum of weight of edges between } A_k \text{ and } \bar{A}_k$$

min-cut: $\underset{A_1, A_2 \cdots A_k}{\arg\min} \ \frac{1}{2}\sum_{k=1}^{k} W(A_k, \bar{A}_k)$

Example:

$K = 2$



Might result in "degenerate" solution

$$\boxed{\text{normcut}(A_1, A_2, \cdots A_k) = \frac{1}{2}\sum_{k=1}^{k} \frac{W(A_k, \bar{A}_k)}{\text{vol}(A_k)}}$$

$$\boxed{Vol(A_k) = \sum_{i \in A} d_i}$$

NP-Hard Problem

0-1 Knapsack

Find N binary vectors

s.t $\mathbf{c}_{ik} = 1$ if i belongs to cluster k

$= 0$ otherwise

Eigen vector problem

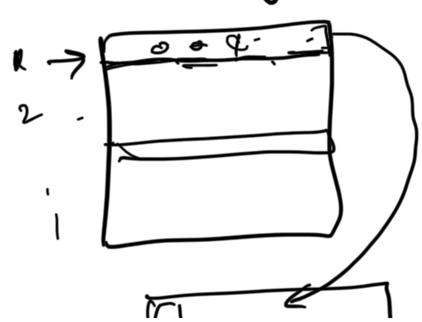
A

$$A \underset{N \times 1}{x} = \underset{\text{scalar}}{\lambda} x$$

Then x is an eigen vector of A

and λ is the eigen value.

W – adjacency matrix



$D_{ii} = \sum^{N} W_{ij}$

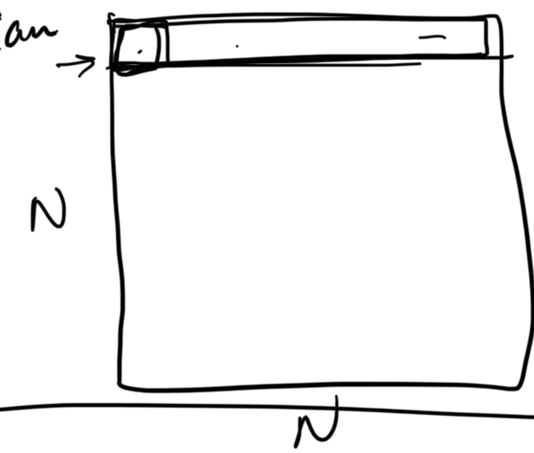
$D_{N\times N}$ $\begin{bmatrix} \square & & & 0 \\ & \square & & \\ & & \ddots & \\ 0 & & & \square \end{bmatrix}$

$D_{ii} = \sum_{j=1} w_{ij}$

$D_{ij} = 0 \quad i \neq j$

$$\boxed{L = D - W}$$

Laplacian



$\sum_{j=1}^{N} L_{ij} = 0$

~~$L_{ij} = D$~~

$L_{ij} = -w_{ij}$ if $i \neq j$

$= \sum w_{ij}$ if $i = j$

(1) $$\boxed{\sum_{j=1}^{N} L_{ij} = -\sum w_{ij} + \sum w_{ij} = 0}$$

(2) N

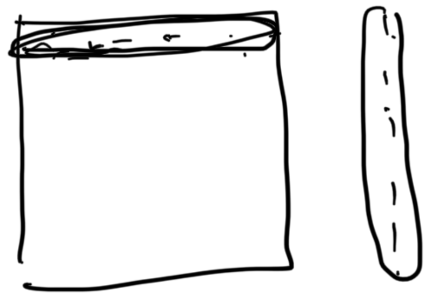
→ eigen vector of L

eigen value of 0

$\mathbb{1}$, 0 will be an eigen vector, eigen-value pair

$$Lx = \lambda x$$

$$\boxed{L\mathbf{1} = 0 \cdot \mathbf{1} = \mathbf{0}}$$



$$= \sum_{j=1}^{N} L_{ij} \neq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

③ L - Symmetric

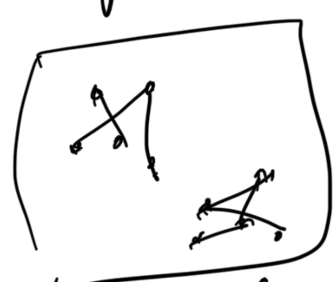
positive semi definite

$$\underset{N\times N}{A}\ \underset{N\times 1}{x} \geq 0 \quad \forall x \neq \mathbf{0}$$

④ will have N non-negative real-valued eigen values

⑤ If W has k connected components.

the L will have k eigen vectors whose eigen-values are 0 and the eigen vectors are

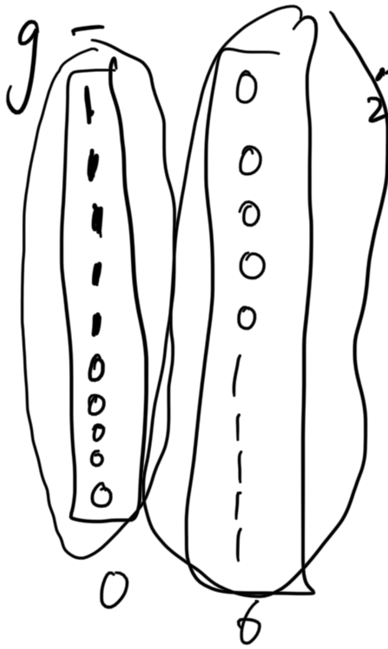


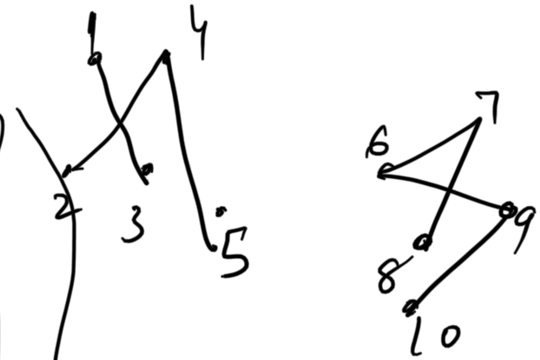
2 connected component

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix}$$

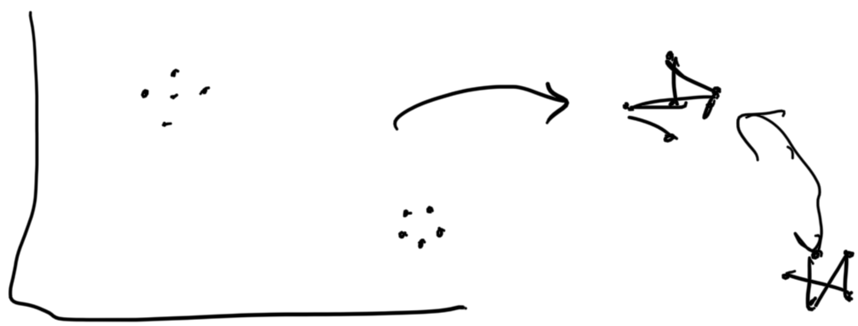
$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

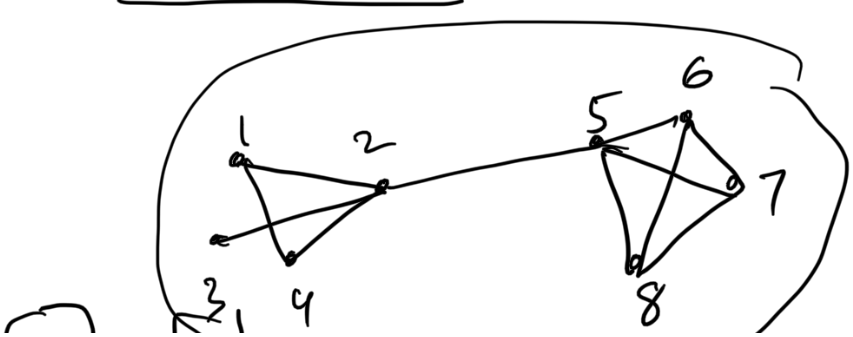
e.g.

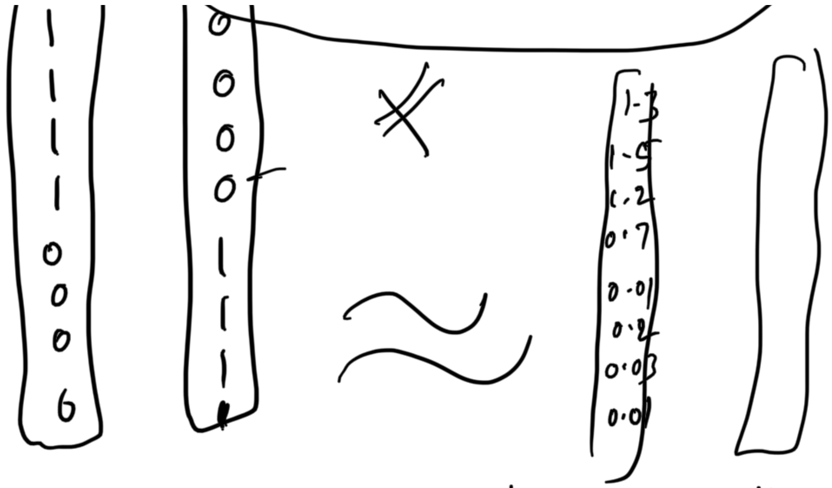




$Ax = \lambda x$







Perturbation theory

$U = N \times k$

kmeans

eigen vector corresponding to smallest eigen-value

---

① $X \longrightarrow S$ $\quad S_{ij} = \underline{sim(x_i, x_j)}$

② $S \longrightarrow W$ $\quad$ k-nearest neighbor

③ $W \longrightarrow L$ $\quad L = D - W$

④ eig (L)

(4) e, v →

eigen values

(5) Choose k-smallest eigen-~~val~~ vectors

↳ U

(6) Perform ~~clus~~ k-means clustering on U.