Graph-based Clustering Methods

Instructor: Ramin Javadi

Isfahan University of Technology

Sharif Optimization & Applications Lab, April 2022

Machine Learning
↳ clustering
↳ Graph-based Methods

Applications in :

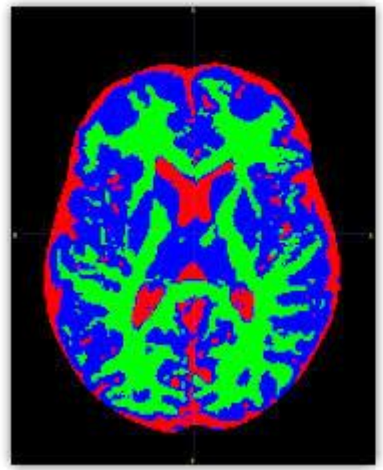
- Bialogical Networks (Neural networks)
- Telecomunication Networks
- Transportation Networks
- Social Networks
- Economic Networks
- Computer vision

Graph-based Clustering Methods

Thursday, April 14, 2022

Motivation

Image segmentation in computer vision



In Medicine

Community detection in social networks

Definitions

Input: A data set with their features
+ similarities or connections

EXP. Facebook: People, their features, Friendship relations

Cluster: A highly connected group of data elements

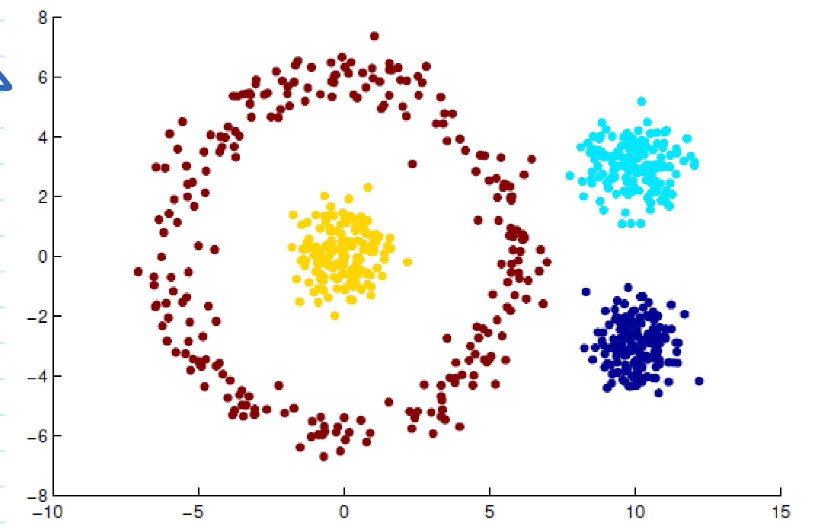
Job: (Clustering) Grouping data elements into clusters with the following properties:

- High intra-clusters similarity (connectivity)
- Low inter-clusters similarity (connectivity)

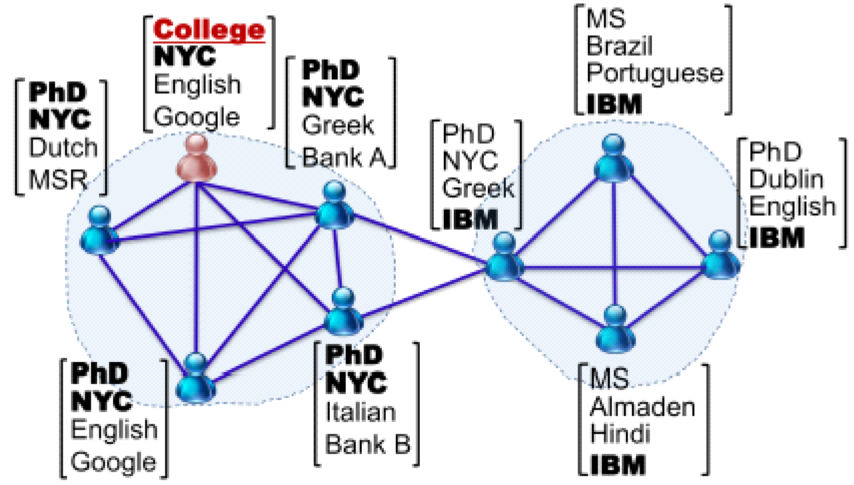
Model-Based clustring vs. dataset with no background information

Similarity based on

Euclidean distance



Attributes



- We want to make the computer learn how to do clustering.
- In high dimensional data, we don't have visual intuition.

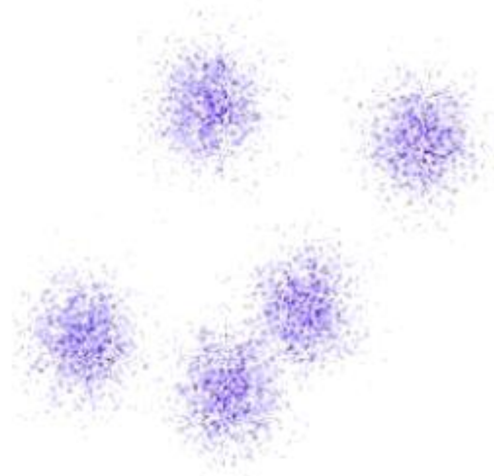
Clustering Criteria

→ Centrality

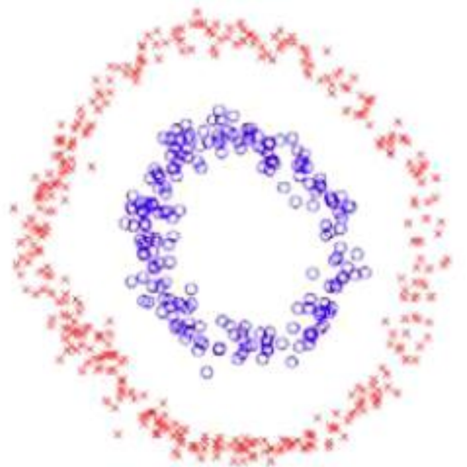
→ Compactness (Density)

→ Connectivity

→ Attributes



Compactness



Connectivity

Center-Based Methods { Distance-Based Methods } Graph-Based Methods
Kernel Methods ...

A Classic Method : k-means

Center-Based

Given a set of n points $X = \{x_1, x_2, \dots, x_n\} \subseteq \mathbb{R}^d$ and a number k.

Objective: Partition X into k sets $S_1, S_2, \dots, S_k$ minimizing cost function:

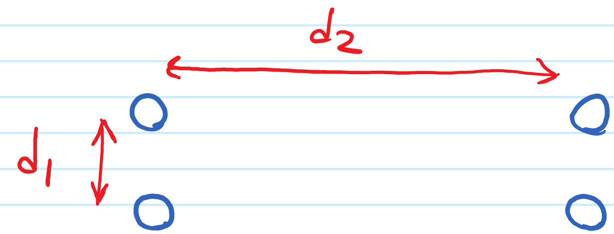
$$\min_{S_1, \dots, S_k} \sum_{i=1}^{k} \sum_{x \in S_i} \| x - c_i \|^2$$

where c_i = centroid of S_i, i.e. $c_i = \frac{1}{|S_i|} \sum_{x \in S_i} x$

Graph-based Clustering Methods

Thursday, April 14, 2022

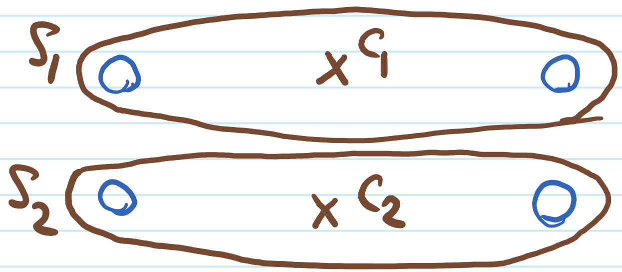
Exp.



$d_2 \gg d_1$

Clustering 1

Bad :(



$\text{Cost} = d_2^2$

Clustering 2

Good :)

$\text{Cost} = d_1^2$

Bad News: The Problem is NP-hard even for $k=2$! :(

Good News: Engineers always have heuristic methods in their pockets! :)

K-means Algorithm: (Lloyd's Algorithm)

1- Start from an initial random centroids.

2- Move each point to the cluster with nearest centroid.

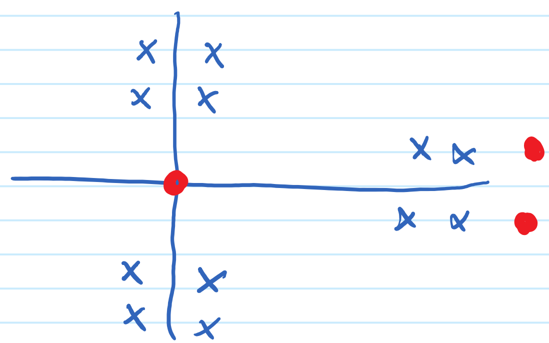
3- Compute the centroids of new clusters.

4- Repeat 2, 3 until clustering does not change.

Initialization: $\begin{cases} \text{1- Random partition method} \\ \text{2- Forgy method (choose } k \text{ observations as initial centroids)} \end{cases}$

Let's see a GIF!

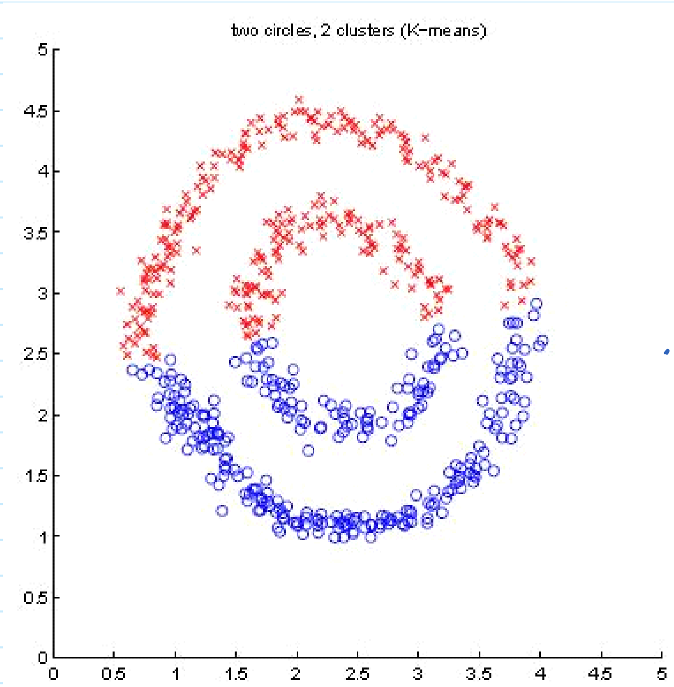
Lloyd's Algorithm is sensitive to initialization and can grind to a halt in local extrema.

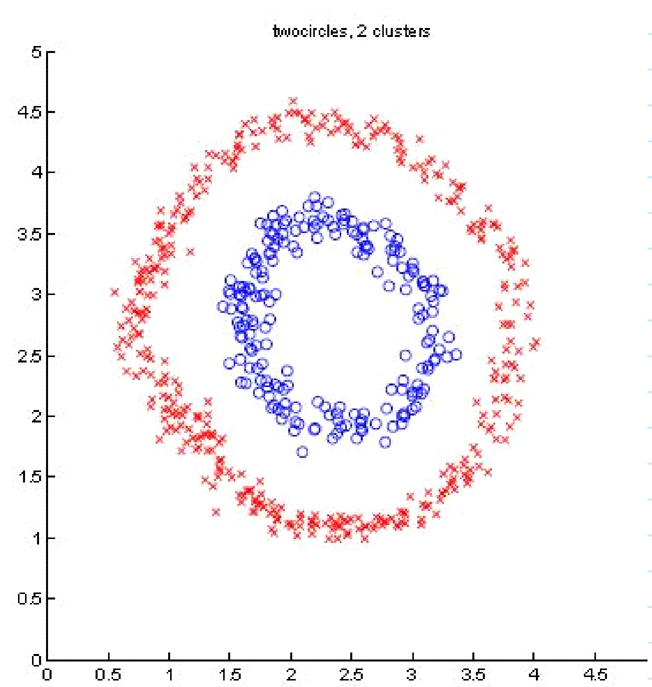


← Bad initialization

* there are some heuristics for initialization such as farthest Traversal and kmeans++
* there are some exact algorithm which run in $O(n^{kd+1})$.

Weakness of K-means and other center-based methods





K-means algorithm

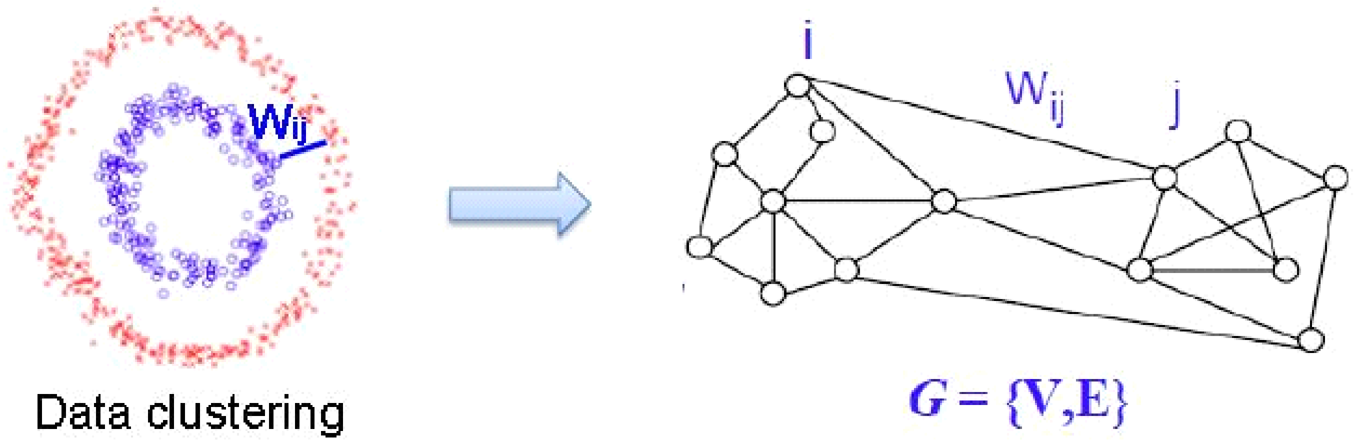
Graph-based methods

Similarity graph model:

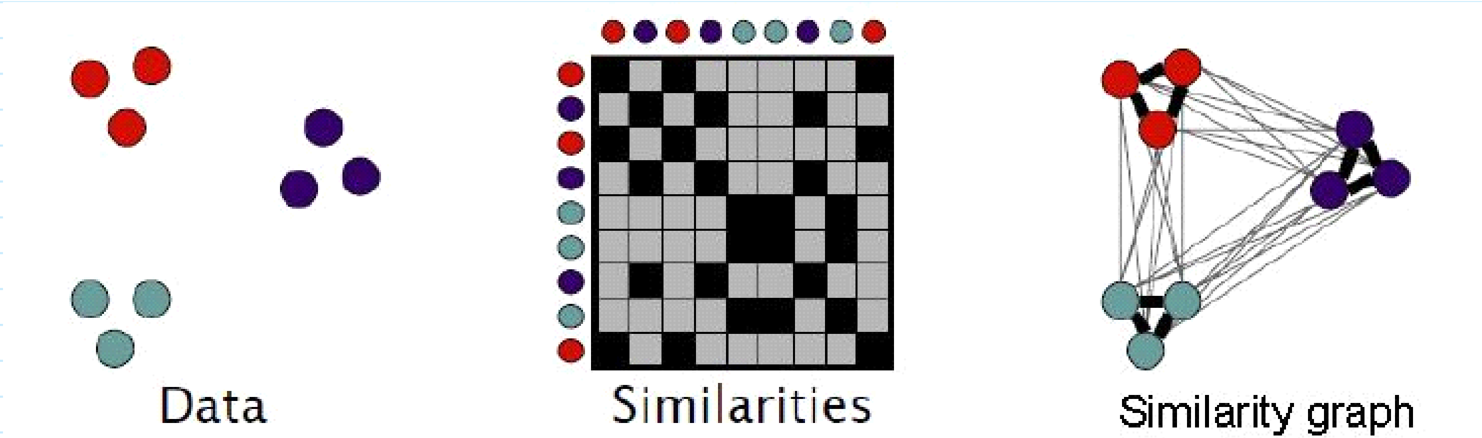
A graph $G=(V,E)$, Each vertex represents a data element

A similarity measure: $w: E \longrightarrow \mathbb{R}^+$

↳ Similarity matrix $\quad w(ij) =$ Proximity of i and j



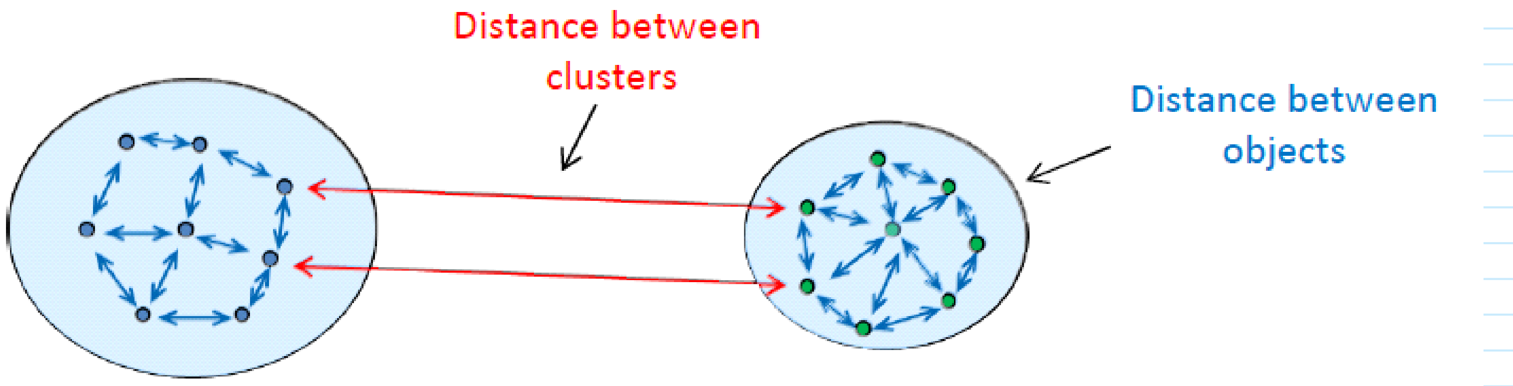
Converting data into a graph



Guassian Kernel similarity function

Each vertex $i \in V(G)$ is endowed with a feature vector $x_i \in \mathbb{R}^d$.

G.K.S.F. is defined as: $w_{ij} = e^{\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)}$



Different Similarity graphs:

1- ε-Neighborhood graph

$$i \sim j \iff \|x_i - x_j\| \leqslant \varepsilon$$

2- k-nearest neighbor graph

$i \sim j \iff$ i is among k nearest neighbor of j or
j is among k nearest neighbor of i

3- Fully connected graph

(complete graph)

Graph-based Methods of clustering:

1- Hierarchical Methods

→ Minimum Spanning Tree

2- Cut-based Methods

→ Sparsest cut, Normalized cut, ...

3- Spectral Methods

→ Eigenspaces of Laplacian

4- Combinations of above methods

Minimum Spanning Tree (MST)

Given a graph $G = (V, E)$ with weights:

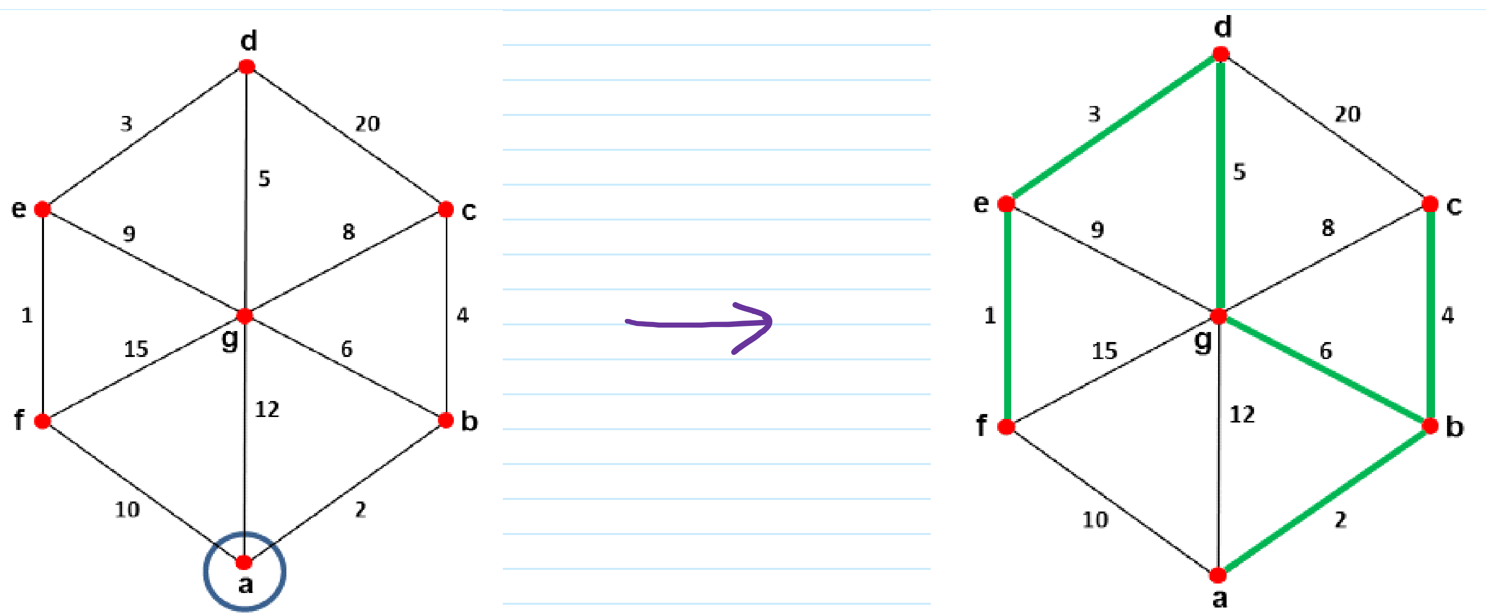
$w_{ij} = \| x_i - x_j \|$ (weights are distances)

MST is a connected spanning subgraph of G with minimum weight.

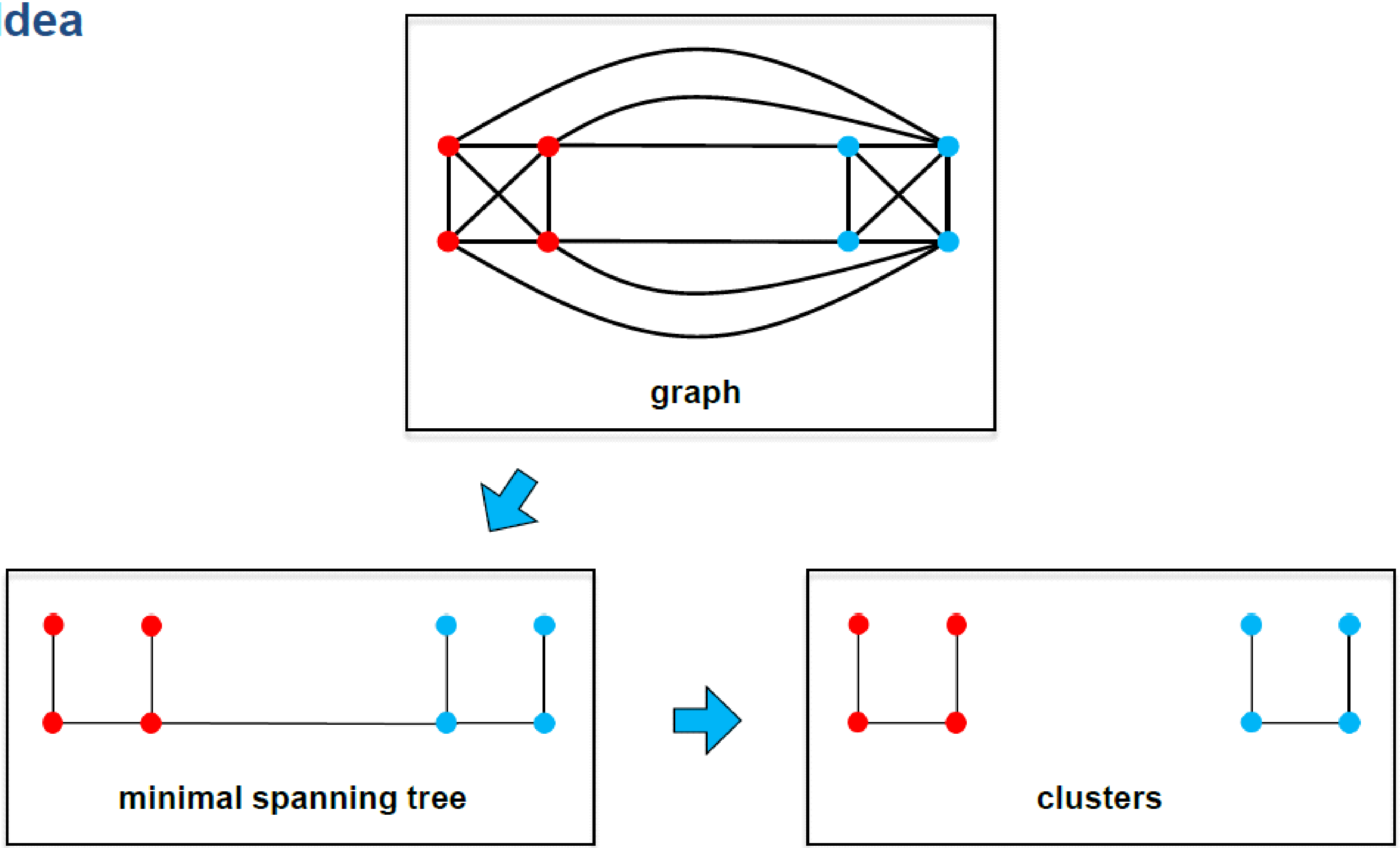
$$\min_{\substack{H \subseteq G \\ H \text{ connected} \\ \text{and spanning}}} \sum_{e \in E(H)} w(e)$$

MST can be computed in $O(n^2)$ using greedy algorithm.

☆ Prim's algorithm ☆ Kruskal's algorithm



Idea



Hierarchical method:

1- Find Minimum Spanning tree.

2- Delete edges iteratively.

(obtained connected components = clusters)

Edge deletion Policies

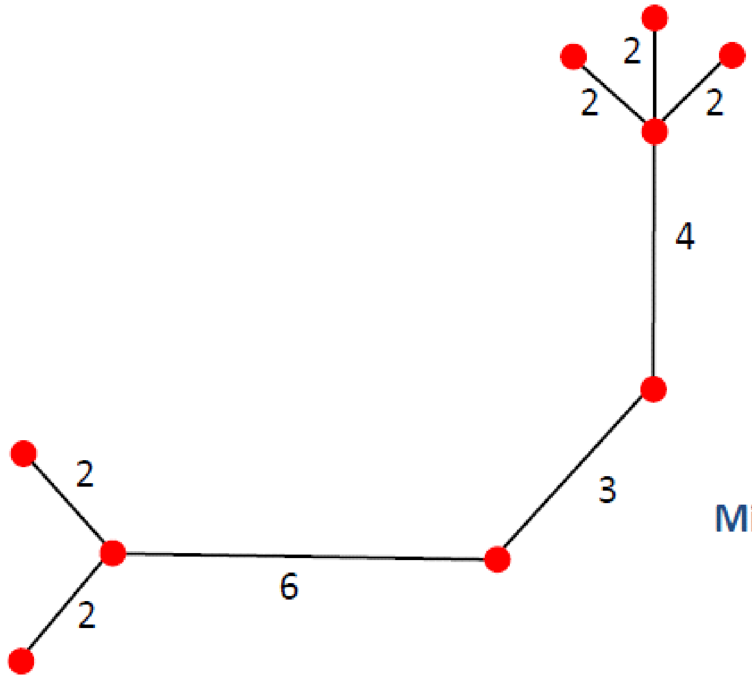
A. Delete edges with maximum weight.

B. Delete inconsistence edges.

Graph-based Clustering Methods

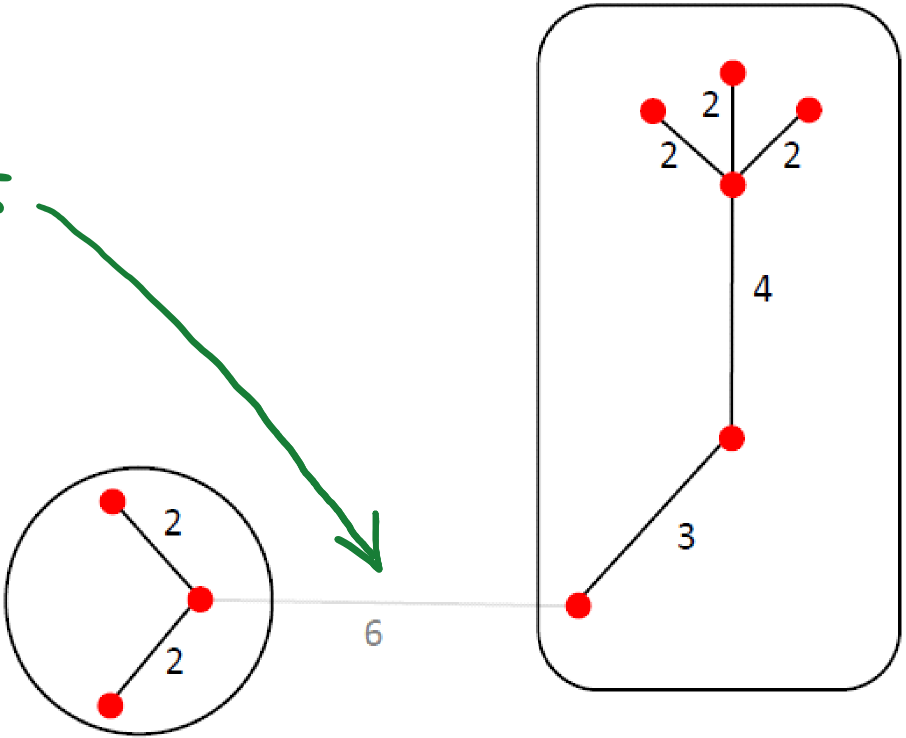
Thursday, April 14, 2022

PLAN A.

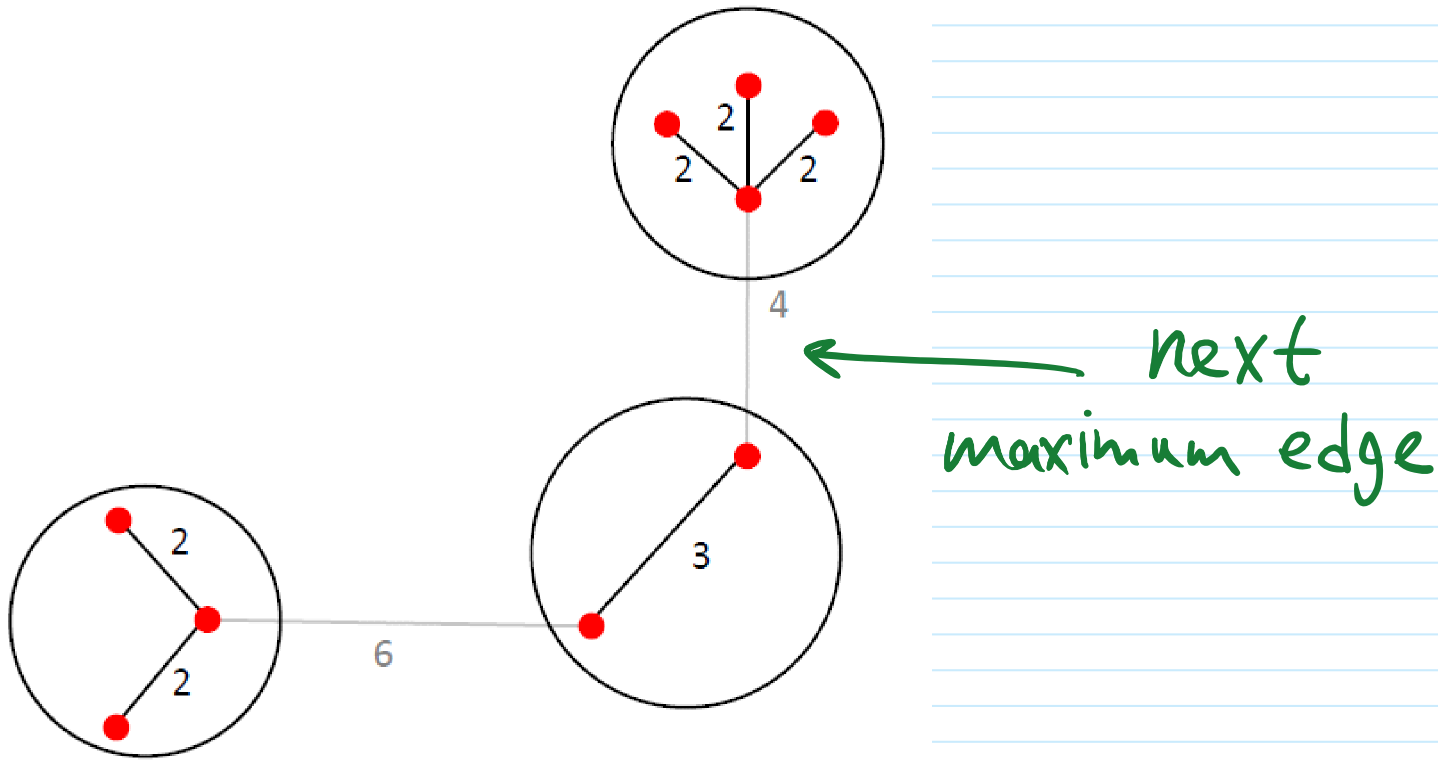


Minimum spanning tree

maximum weight



2 clusters

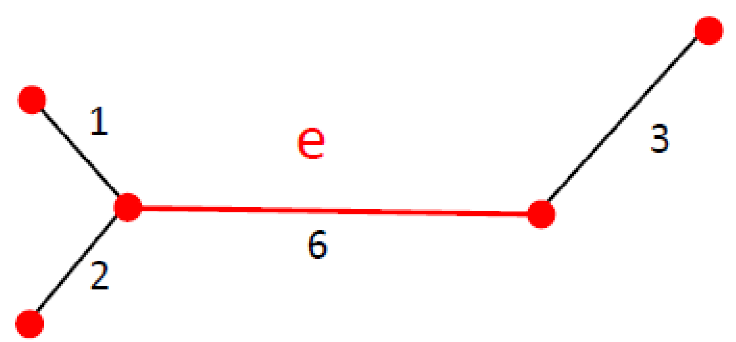


3 clusters

PLAN B. (Zahn's Algorithm)

An edge e is incosistence if its weight w_e is (much) larger than $\overline{w_e}$, where

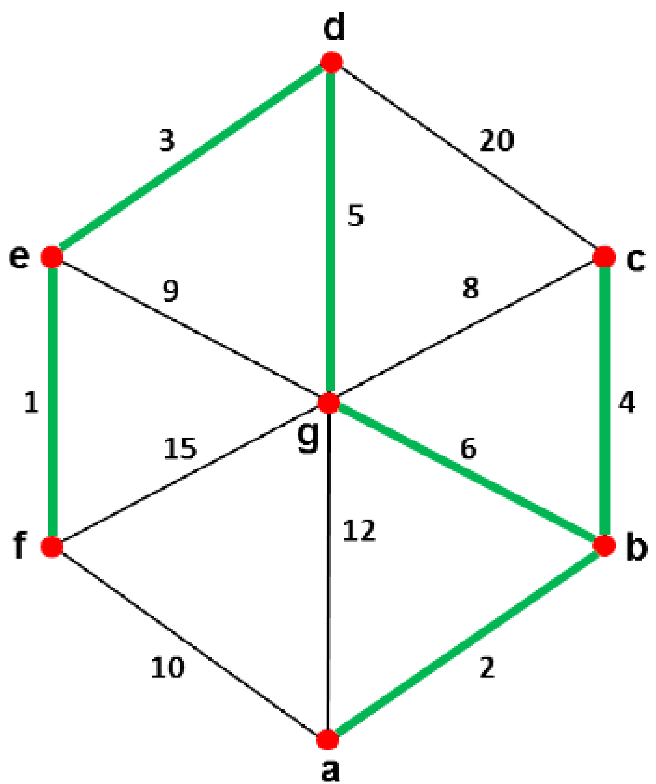
$\overline{w_e}$ = average of weights of edges adjacent to e.

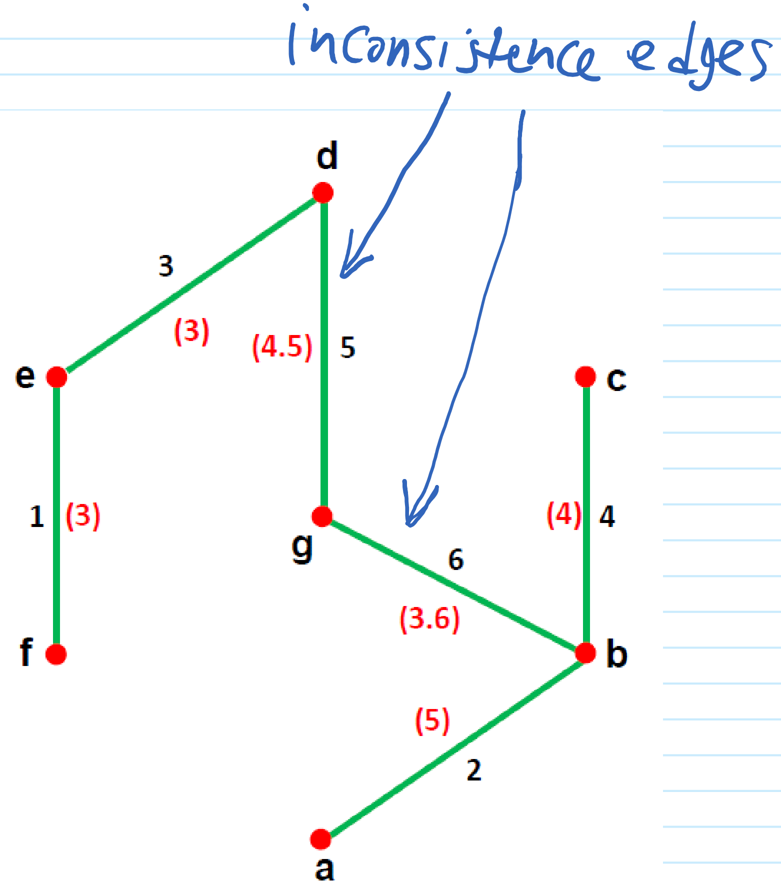


$$w_e = 6$$

$$\overline{w_e} = \frac{3+2+1}{3} = 3$$

$w_e > \overline{w_e} \Rightarrow e$ is inconsistence


d
3
20
5
e
c
9
8
1
4
g
15
6
f
b
12
10
2
a


inconsistence edges
d
3
(3)
(4.5)
5
e
c
1
(3)
(4)
4
g
6
(3.6)
f
b
(5)
2
a

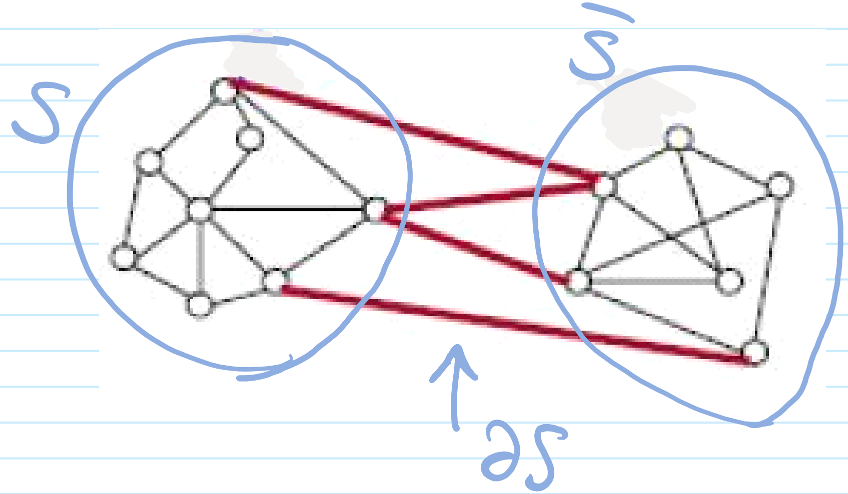
Cut-based Methods

$G=(V,E)$, $w: E \to \mathbb{R}^+$ is similarity measure

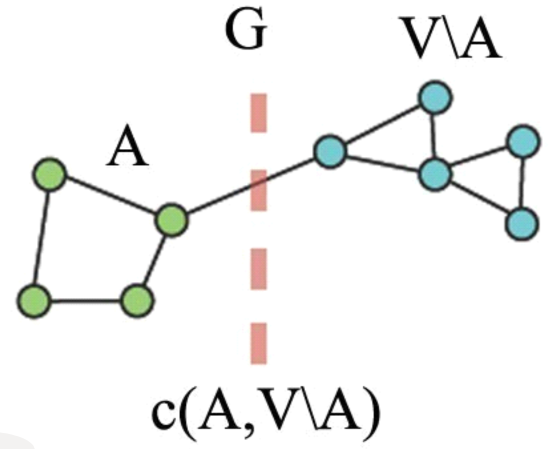
$\forall\, S \subseteq V(G)$, $\partial S = \{ e = uv \in E(G) \mid u \in S,\ v \in \bar{S} \}$

A cut

$$w(\partial S) = \sum_{e \in \partial S} w(e)$$



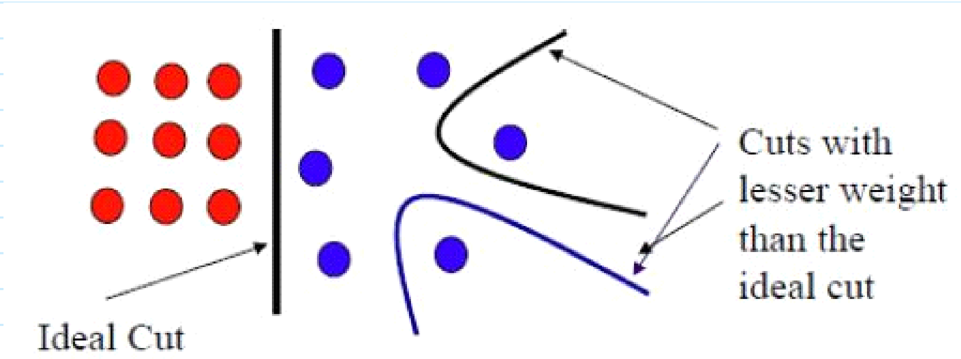
Min Cut: $\min\limits_{\substack{S \neq \phi \\ S \neq V(G)}} w(\partial S)$

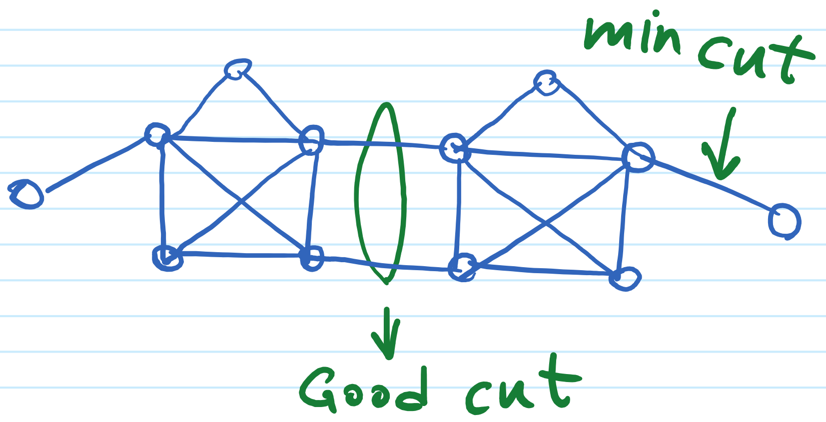


Clusters = Connected Components after removing edges in cut

Min Cut can be found efficiently in $O(n^3)$.
(Ford-Fulkerson's Algorithm)

Min Cut is not so good!





We have to Normalize the cut!

Sparsest cut

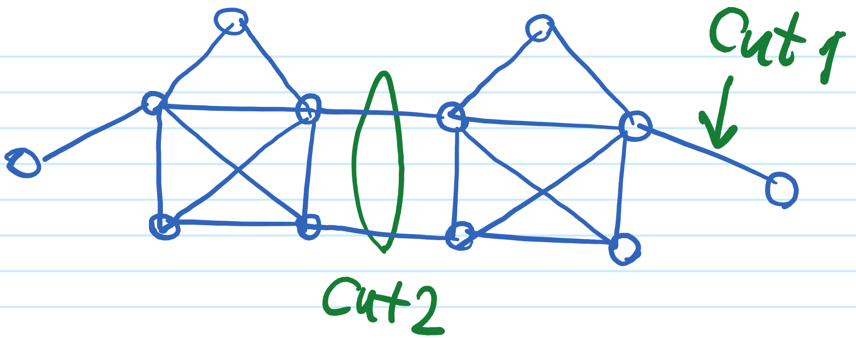
$$h(G) = \min_{\substack{S \neq \phi \\ \neq V(G)}} \frac{1}{2}\left(\frac{w(\partial S)}{|S|} + \frac{w(\partial S)}{|\bar{S}|} \right)$$

Or

$$\iota(G) = \min_{\substack{S \neq \phi \\ \neq V(G)}} \max\left(\frac{w(\partial S)}{|S|}, \frac{w(\partial S)}{|\bar{S}|} \right)$$

These parameters are called Conductance or Edge expansion. The problem is called Isoperimetric Problem.

Exp.



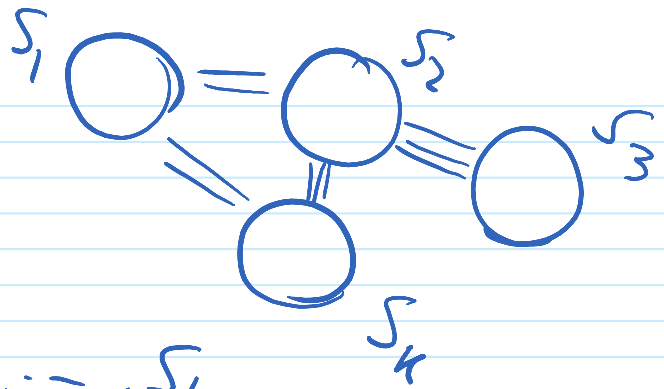
edge weights = 1

Cut 1: $\frac{1}{2}\left(\frac{w(\partial S)}{|S|}+\frac{w(\partial S)}{|\bar{S}|}\right)=\frac{1}{2}\left(\frac{1}{1}+\frac{1}{11}\right)=\frac{6}{11}$

cut 2: $\frac{1}{2}\left(\frac{w(\partial S)}{|S|}+\frac{w(\partial S)}{|\bar{S}|}\right)=\frac{1}{2}\left(\frac{2}{6}+\frac{2}{6}\right)=\frac{1}{3}<\frac{6}{11}$

Cut 2 is the sparsest cut.

Better Criteria, however unfortunately both problems are NP-hard! :(



Multi-way Normalized cut

Given graph G and number k.

Partition $V(G)$ into k sets $S_1, S_2, \dots, S_k$

minimizing $$h_k(G) = \frac{1}{k}\left(\frac{w(\partial S_1)}{|S_1|} + \cdots + \frac{w(\partial S_k)}{|S_k|}\right)$$

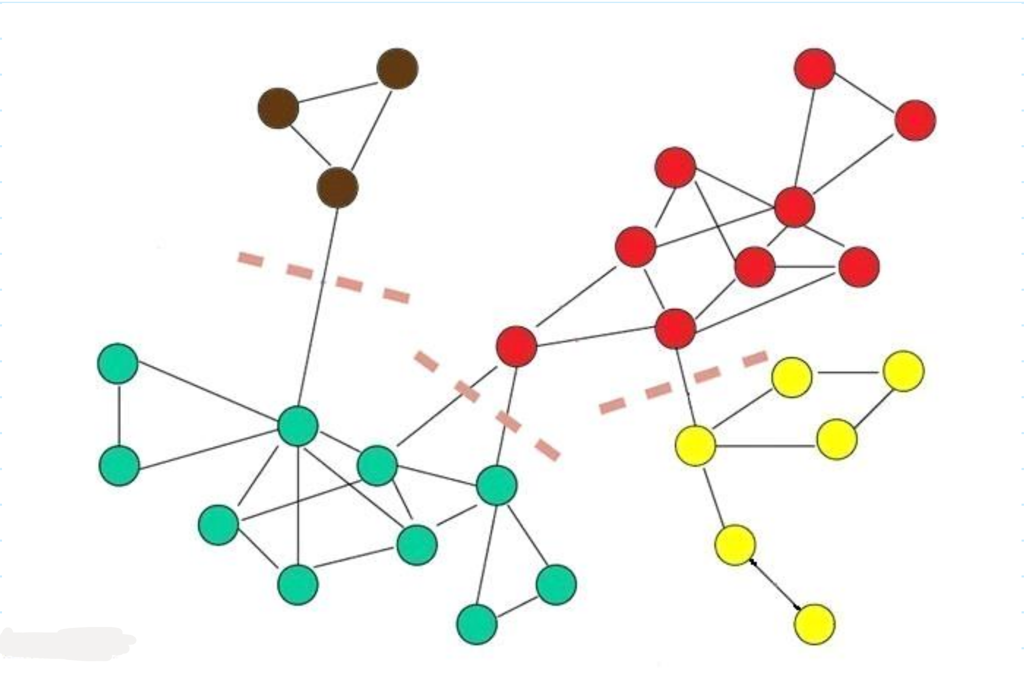
or

minimizing $$l_k(G) = \max\left(\frac{w(\partial S_1)}{|S_1|}, \dots, \frac{w(\partial S_k)}{|S_k|}\right)$$

Graph-based Clustering Methods

Thursday, April 14, 2022

$k = 4$



$$h_4(G) = \frac{1}{4}\left(\frac{1}{3} + \frac{3}{10} + \frac{3}{9} + \frac{1}{6}\right)$$

$$c_4(G) = \max\left\{\frac{1}{3}, \frac{3}{10}, \frac{3}{9}, \frac{1}{6}\right\} = \frac{1}{3}$$

Spectral clustering

Given graph $G=(V,E)$ and similarity measure

Similarity matrix : $W=(w_{ij})$, $w_{ij}=\begin{cases} weight(e) & \text{if } e=ij \in E(G) \\ 0 & \text{o.w.} \end{cases}$

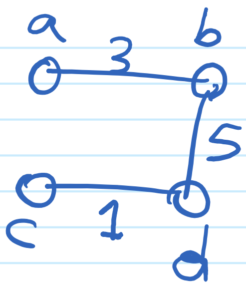
Diagonal matrix : $D=(d_{ij})$, $d_{ij}=\begin{cases} w(i) & i=j \\ 0 & \text{o.w} \end{cases}$

$$w(i)=\sum_{e\sim i} w(e)$$

Laplacian Matrix, $\boxed{\mathcal{L}=D-W}$

Graph-based Clustering Methods

Thursday, April 14, 2022



$$\mathcal{L} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 3 & -3 & 0 & 0 \\ -3 & 8 & 0 & -5 \\ 0 & 0 & 1 & -1 \\ 0 & -5 & -1 & 6 \end{bmatrix} \end{array}$$

If $\mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \Rightarrow \mathcal{L}\mathbf{1} = 0$

☆ $\mathcal{L}$ is positive semidefinite and its eigenvalues are $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. First eigenvector $= \mathbf{1}$

$\mathbb{R}^n$ has an orthonormal basis of eigenvectors $v_1 = \mathbf{1}, v_2, \cdots, v_n$

★ G is connected if and only if $\lambda_2 > 0$

we will see that graphs with larger λ_2 are more connected

λ_2 = Algebraic Connectivity, Spectral gap

How is λ_2 related to connectivity?

For every vector $f \in \mathbb{R}^n$, define

Rayleigh quotient: $\dfrac{f^t L f}{f^t f}$

$\boxed{\text{Variational Theorem}}$

$$\lambda_2 = \min_{f \perp 1} \frac{f^t L f}{f^t f}$$

Graph-based Clustering Methods

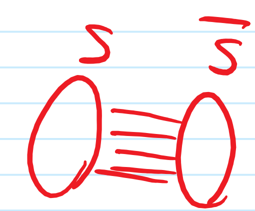
Thursday, April 14, 2022

$$f^t \mathcal{L} f = f^t D f - f^t W f = \sum_{i=1}^{n} f_i^2 d_i - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$\Rightarrow \boxed{f^t \mathcal{L} f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2}$$

Can you see why $\lambda_2 > 0$ if G is connected ?!

Now, define f as $f_i = \begin{cases} |\bar{S}| & i \in S \\ -|S| & i \notin S \end{cases}$



Then, $\sum_{i=1}^{n} f_i = |S||\bar{S}| - |S||\bar{S}| = 0$

$$\Rightarrow f \perp \mathbb{1}$$

$$V.T. \Rightarrow \lambda_2 \leq \frac{f^t L f}{f^t f} = \frac{\frac{1}{2}\sum_{i,j} w_{ij}(f_i - f_j)^2}{\sum f_i^2}$$

$$= \frac{w(\partial S)\,(|S|+|\bar{S}|)^2}{|\bar{S}|^2|S| + |S|^2|\bar{S}|}$$

$$= \frac{w(\partial S)\,(|S|+|\bar{S}|)}{|S|\,|\bar{S}|}$$

$$= \left(\frac{w(\partial S)}{|S|} + \frac{w(\partial S)}{|\bar{S}|} \right)$$

$$= 2\,h(G)$$

Higher λ_2

$\Downarrow$

Higher Conductance

In fact

(Poly time solvable) $\lambda_2 = \min\limits_{f \perp 1} \frac{f^t \mathcal{L} f}{f^t f}$

(NP-hard) $h(G) = \min\limits_{\substack{f \perp 1 \\ f \text{ is two-valued}}} \frac{f^t \mathcal{L} f}{f^t f}$

Spectral clustering: Use λ_2 and its eigenvector to approximate $h(G)$

Cheeger's inequality:

$$\frac{1}{c} h^2(G) \leq \lambda_2 \leq 2h(G) \quad \text{(Alon-Milman '85)}$$

$c = \max_i w(i)$

That's why λ_2 is called algebraic connectivity!

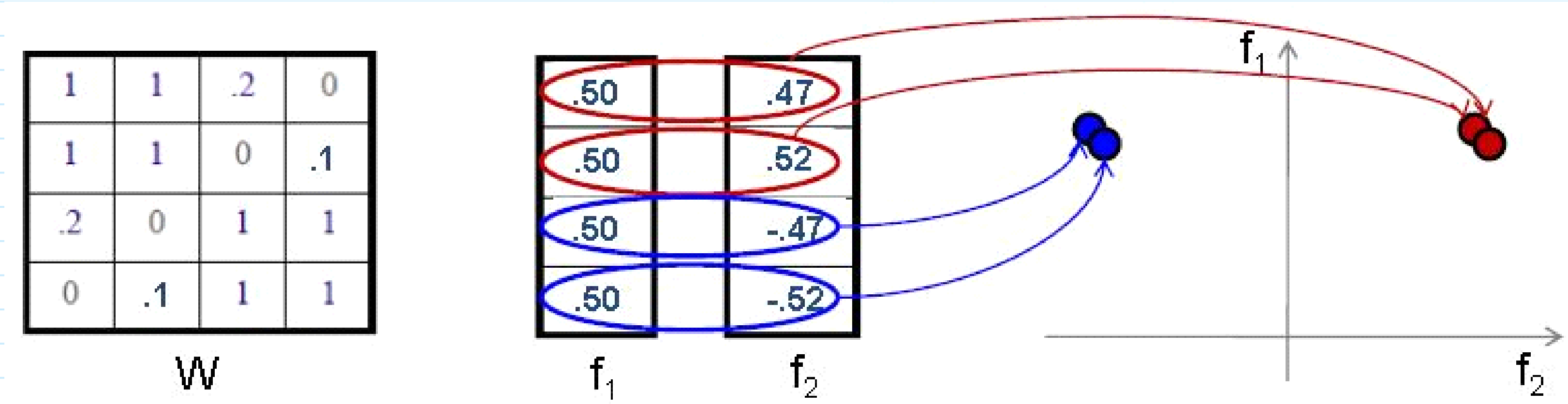
Regular graphs with large λ_2 is called Ramanujan Graphs.

Spectral clustering Algorithm (2-clustering)

1- Compute Laplacian Matrix.

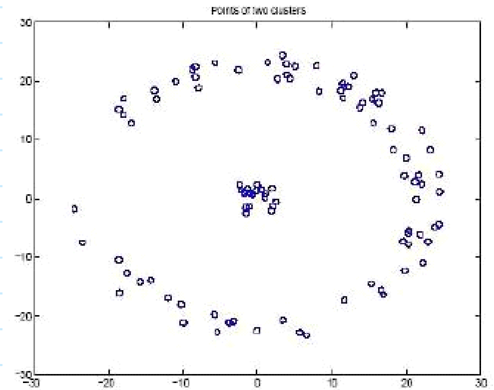
2- Compute eigenvector v corresponding to λ_2.

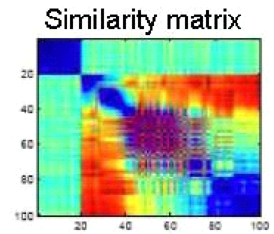
3- Define $S=\{i \mid v_i \geq 0\}$, $\bar{S}=\{i \mid v_i<0\}$

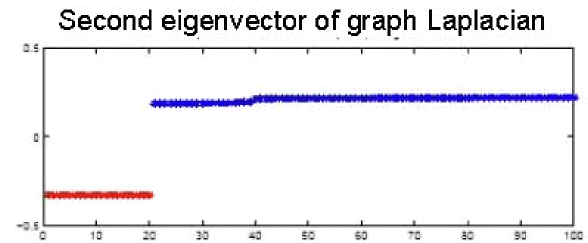


Graph-based Clustering Methods

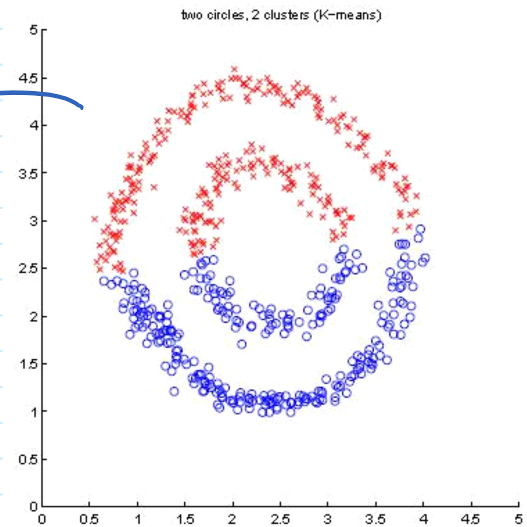
Thursday, April 14, 2022

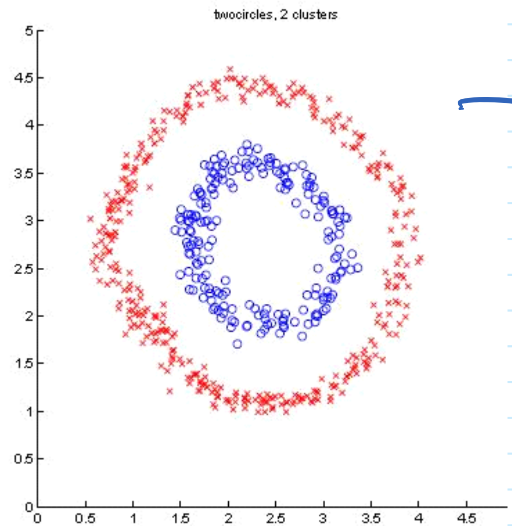






K-means





Spectral clustering

Spectral clustering (k-clustering)

Generalized Cheeger's inequality

$$\frac{f(k)}{c} h_k^2(G) \leq \lambda_k \leq h_k(G).$$

$c = \max_i w(i)$

(J. Lee, S. Oveis Gharan, L. Trevisan '14)

Algorithm: (Jordan, Weiss '02, Shi, Malik '00)

1- Compute Laplacian Matrix.

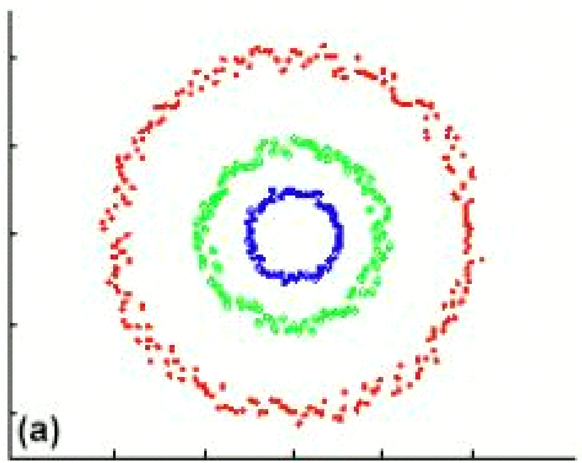
2- Compute first k eigenvectors $v_1, \ldots, v_k$ corresponding to first k eigenvalues $\lambda_1, \ldots, \lambda_k$

3- Construct matrix $M = [v_1 \cdots v_k]_{n \times k}$

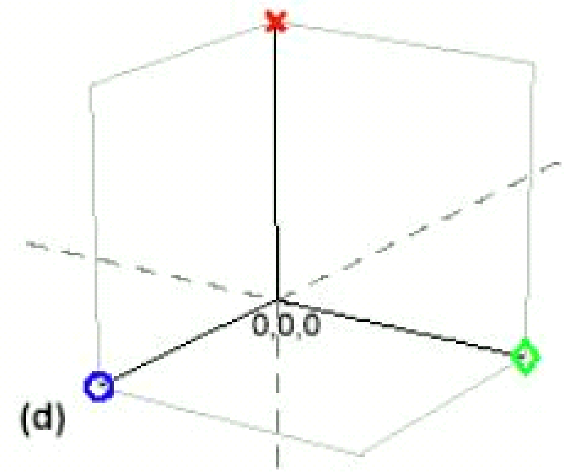
Let $x_1, \ldots, x_n \in \mathbb{R}^k$ be n rows of M.

Cluster $\{x_1, \ldots, x_n\}$ using k-means algorithm.

Original data

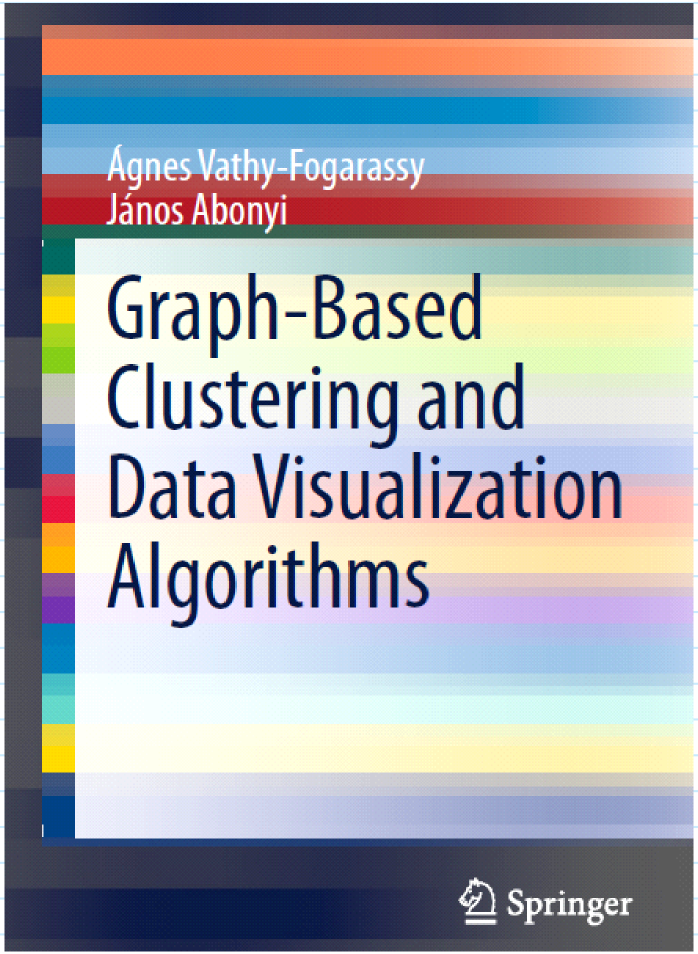


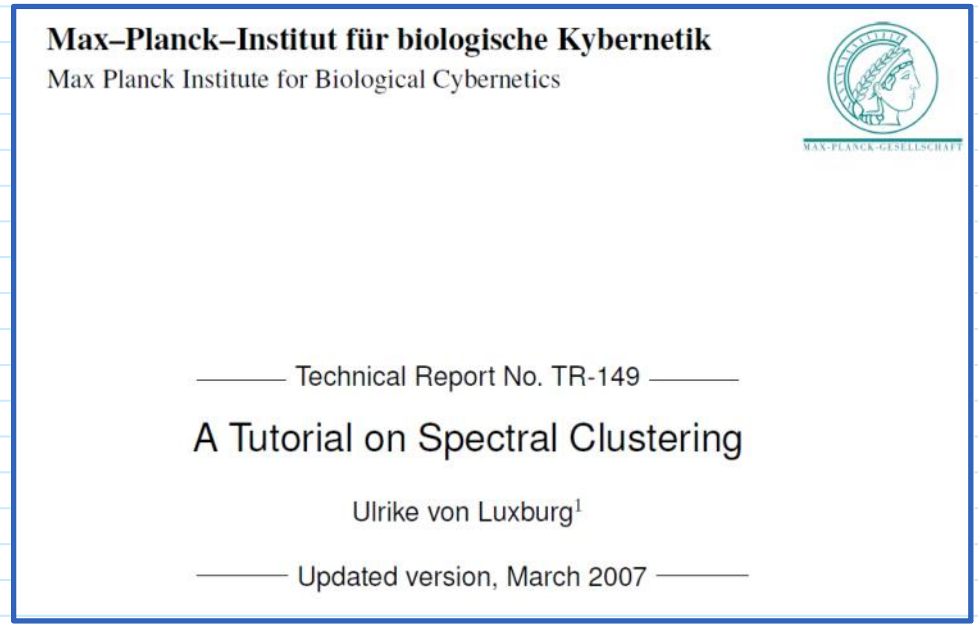
Projected data



Graph-based Clustering Methods

Thursday, April 14, 2022





Max–Planck–Institut für biologische Kybernetik

Max Planck Institute for Biological Cybernetics

Technical Report No. TR-149

A Tutorial on Spectral Clustering

Ulrike von Luxburg[1]

Updated version, March 2007

BULLETIN (New Series) OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 43, Number 4, October 2006, Pages 439–561
S 0273-0979(06)01126-8
Article electronically published on August 7, 2006

EXPANDER GRAPHS AND THEIR APPLICATIONS

SHLOMO HOORY, NATHAN LINIAL, AND AVI WIGDERSON