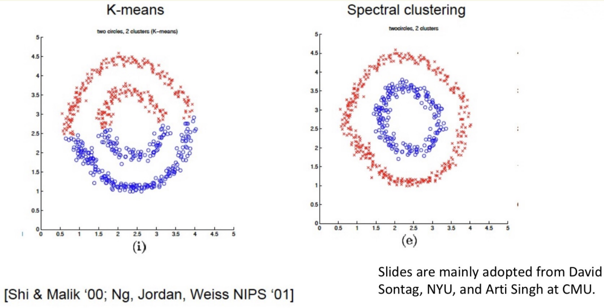


Spectral clustering

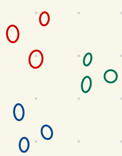
- Core idea: Form a graph and group points using graph structure.

1. Similarity graph

Notation : $G(V,E,W)$, where V: Vertices (Data Points), E : Edge if similarity >0, W: Edge weights (similarities)

Goal: Given data points $x_1, x_2, \dots, x_n$ and similarities $w(x_i, x_j)$, partition the data into groups so that points in a group are similar and points in different groups are dissimilar.

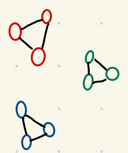
Ex)



Data

1	0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0	0
1	0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0	0
0	0	0	0	1	1	0	1	0
0	0	0	0	1	1	0	1	0
0	1	0	1	0	0	1	0	0
0	0	0	0	1	1	0	1	0
1	0	1	0	0	0	0	0	1

Similarities



Similarity graph

☆ Partition the graph so that edges within a group have large weights and edges across groups have small weights.

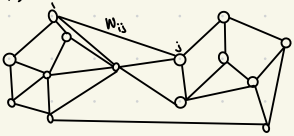
Similarity graph Construction. : Model local neighborhood relations between data points

E.g. Gaussian kernel similarity function

$$W_{ij} = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}$$

σ → Controls size of neighborhood

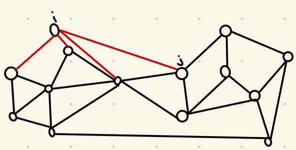
ex)



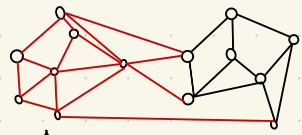
$G = \{V, E\}$

Graph Terminologies

- Degree of nodes : $d_i = \sum_j W_{ij}$

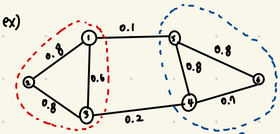


- Volume of a set : $vol(A) = \sum_{i \in A} d_i, \; A \subseteq V$



- Graph Cut: $cut(A,B) = \sum_{i \in A, j \in B} W_{ij}$

ex)



$$cut(A,B) = \sum_{i \in A, j \in B} W_{ij} = 0.3$$

- An intuitive goal is find the partition that minimizes the cut.

- Laplacian of a graph is defined as: $L = D - W$

Ex)

D =

	1	2	3	4	5	6
1	1.5	0	0	0	0	0
2	0	1.6	0	0	0	0
3	0	0	1.6	0	0	0
4	0	0	0	1.7	0	0
5	0	0	0	0	1.7	0
6	0	0	0	0	0	1.5

,

W =

	1	2	3	4	5	6
1	0	0.8	0.6	0	0.1	0
2	0.8	0	0.8	0	0	0
3	0.6	0.8	0	0.2	0	0
4	0	0	0.2	0	0.8	0.7
5	0.1	0	0	0.8	0	0.8
6	0	0	0	0.7	0.8	0

L =

	1	2	3	4	5	6
1	1.5	-0.8	-0.6	0	-0.1	0
2	-0.8	1.6	-0.8	0	0	0
3	-0.6	-0.8	1.6	-0.2	0	0
4	0	0	-0.2	1.7	-0.8	-0.7
5	-0.1	0	0	-0.8	1.7	-0.8
6	0	0	0	-0.7	-0.8	1.5

How can we formalize cut(A,B) as a tractable optimization problem?

⇒ Mathematically formalize cut(A,B)

1) Set $f_i = \begin{cases} \frac{1}{\sqrt{2}} & i \in A \\ -\frac{1}{\sqrt{2}} & i \in B \end{cases}$

2) $f^T L f = f^T D f - f^T W f$

$$= \sum_i d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij}$$

$$= \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2 = cut(A,B)$$

3) $cut(A,B) = f^T L f$

★ $\underset{f \in [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]^N}{\operatorname{argmin}} f^T L f$

• Normalized Cut : $Ncut(A,B) = \frac{Cut(A,B)}{Vol(A)} + \frac{Cut(A,B)}{Vol(B)}$

$= Cut(A,B) = \frac{Vol(A)+Vol(B)}{Vol(A)\,Vol(B)}$

• Normalized Cut and Graph Laplacian : $Ncut(A,B) = \frac{f^T L f}{f^T D f}$

Hence, in Spectral clustering, we need to solve the following:

$$\min f^T L f$$
$$\text{s.t } f^T D f = 1$$

how? : Solve the following eigenvalue problem: $(D^{-1}L) f = \lambda f$, and pick the second smallest eigenvector f.

Summary:

① Form a similarity Graph → ex) using Gaussian kernel similarity function

② Calculate D: Degree matrix, W = Weights matrix, and L: Laplacian matrix where $L = D - W$

③ Calculate eigen-decomposition of $D^{-1}Lv = \lambda v$

④ Choose the K-eigenvectors associated w/ the smallest eigenvalues (avoid the trivial one)

⑤ $X \in \mathbb{R}^{N \times d} \rightarrow V = [v_1, \dots, v_k] \in \mathbb{R}^{N \times k}$

⑥ Perform K-means clustering on V.