DDA4210/MAIR6002 Advanced Machine Learning
Lecture 05-I Graph Cut and Spectral Clustering

Tongxin Li

School of Data Science, CUHK-Shenzhen

Spring 2024

Overview

Unsupervised Learning

- Supervised learning
  - Use labeled data pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ to learn a function $\mathbf{y} = f(\mathbf{x})$.
- Unsupervised learning
  - Learn something useful from unlabeled data $\{\mathbf{x}_i\}_{i=1}^N$.

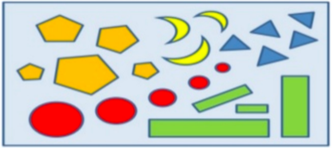
Clustering

- Clustering

  $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9, \mathbf{x}_{10}\}$

  $\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_8\}$ $\{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_{10}\}$ $\{\mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_9\}$

  - Unsupervised grouping of datapoints.
  - Knowledge discovery.
  - Useful when don't know what you're looking for.
- Basic idea of clustering
  - Group together similar instances.



Clustering Algorithms

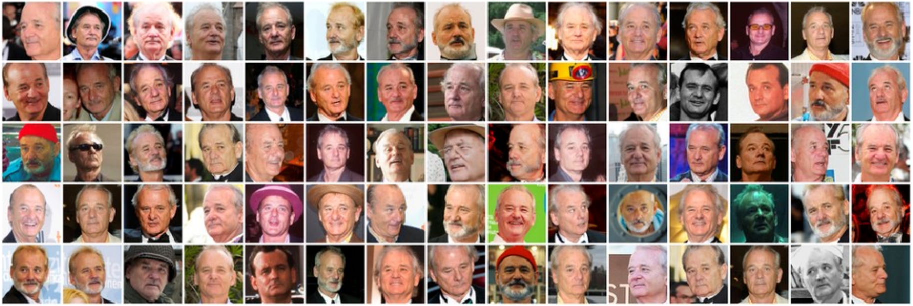
- Hierarchical clustering (intuitive, not included in this course)
- K-means clustering (learned in basic ML courses)
- Mixture of Gaussians (learned in basic ML courses)
- **Spectral clustering**
- Subspace clustering (not included in this course)
- Deep learning based clustering (not included in this course)

Applications of Clustering

- Image segmentation
  - Break up image into meaningful or perceptually similar regions.

Applications of Clustering

- Image clustering



Difficult!

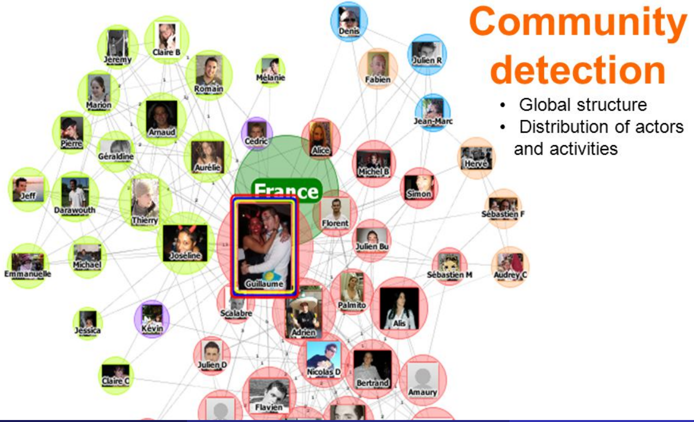
Applications of Clustering

- Image clustering

Very difficult!

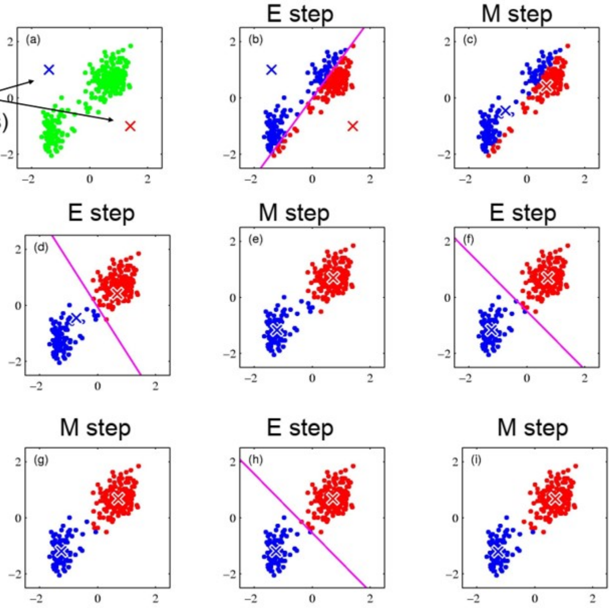
Applications of Clustering

- Gene and cell clustering
- Document clustering
- Recommendation system (How to do?)
- Social network analysis
- Community detection



K-Means Clustering: Example

Initial Choice of Means (Parameters)

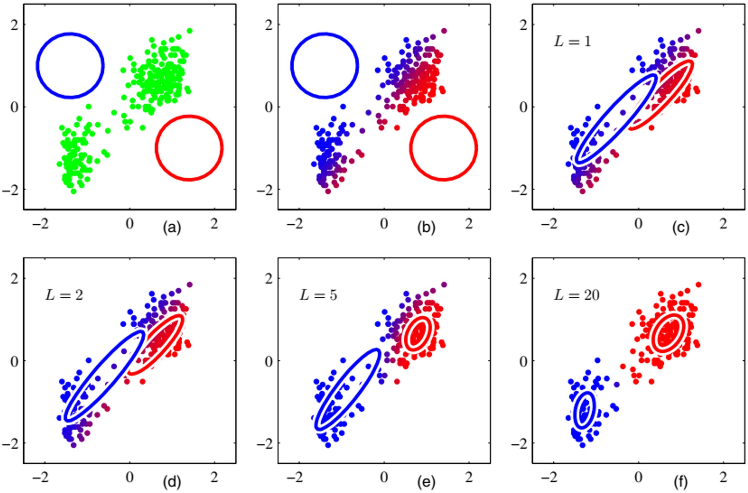


E step: parameters are fixed Distributions are optimized

M step: distributions are fixed Parameters are optimized

Final Clusters And Means

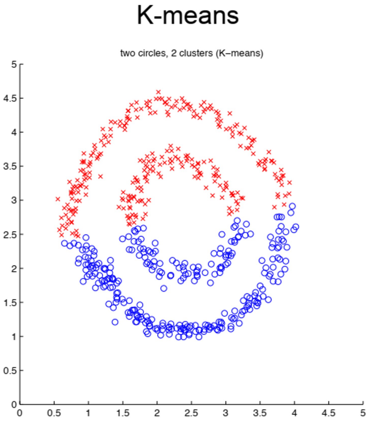
GMM: Example

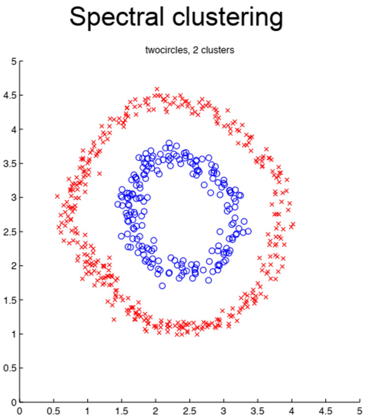


L: cycles of EM

This is the Old Faithful Geyser dataset [PRML, Bishop]

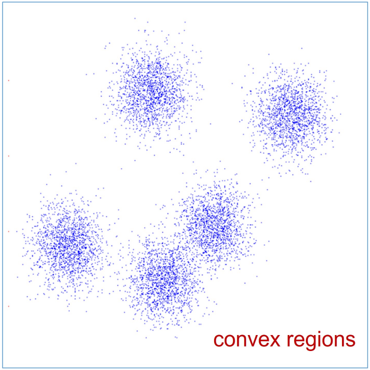
Main Limitation of K-means



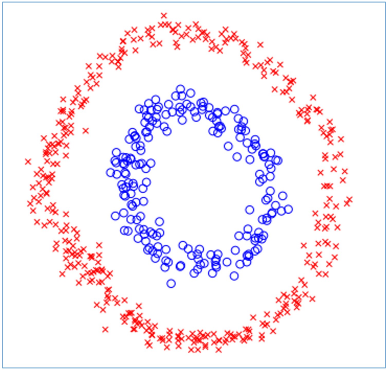


Clustering Criterion

- Two different clustering criteria
  - Compactness, e.g., k-means, Gaussian mixture models
  - Connectivity, e.g., spectral clustering



Compactness



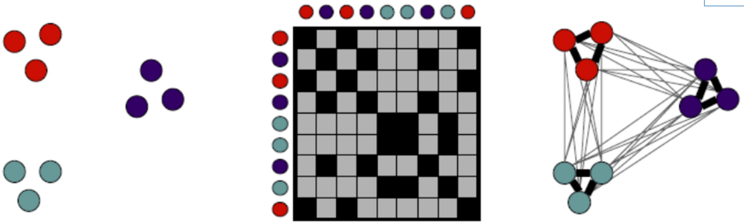
Connectivity

Graph Partition

Similarity Graph: G(V,E,W)

V – Vertices (Data points)
E – Edge if similarity > 0
W - Edge weights (similarities)

affinity matrix

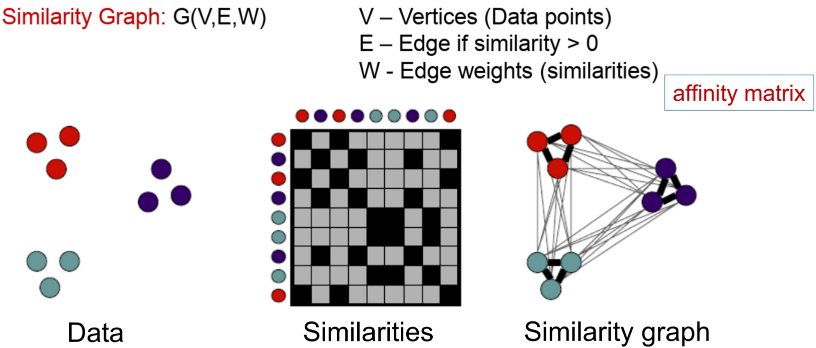


Data　　Similarities　　Similarity graph

$$V = \{v_1, v_2, \ldots, v_N\}, \quad E = \{e_1, e_2, \ldots, e_l\}, \quad W = \begin{bmatrix} & \vdots & \\ \cdots & w_{ij} & \cdots \\ & \vdots & \end{bmatrix}$$

W is usually nonnegative and symmetric, and $w_{ii} = 0$.

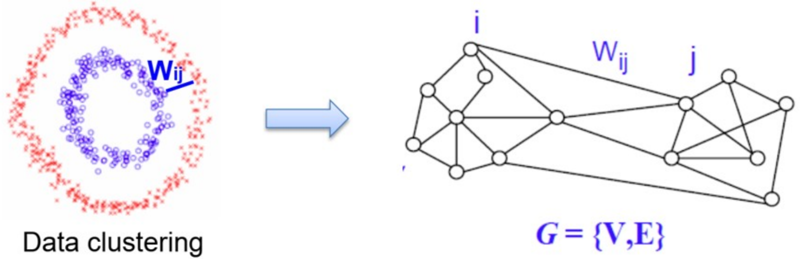
Graph Partition



- Similarity graph
  - Model local neighborhood relations between data points
  - Exist naturally or need to be constructed
- Graph partition: Partition the graph so that edges within a group have large weights and edges across groups have small weights.

Similarity Graph Construction

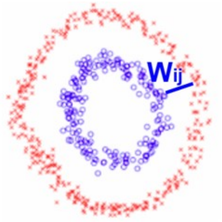
Given $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, construct a similarity graph.



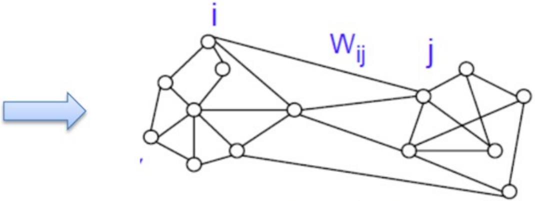
Data clustering

Similarity Graph Construction

Given $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, construct a similarity graph.



Data clustering



$G = \{V, E\}$

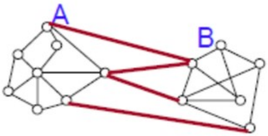
- k-nearest neighbor graph
- ϵ-neighborhood graph
- Gaussian kernel similarity function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

Minimum Cut

Minimum cut: Partition graph into two sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum.

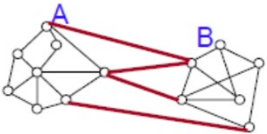
$$\text{cut}(A, B) := \sum_{i \in A, j \in B} w_{ij}$$



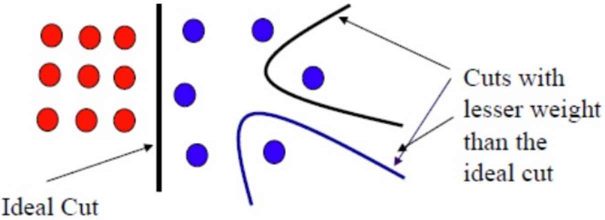
Minimum Cut

Minimum cut: Partition graph into two sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum.

$$\mathrm{cut}(A, B) := \sum_{i \in A, j \in B} w_{ij}$$



- Easy to solve $O(|V||E|)$ algorithm
- Not satisfactory partition? Often isolates vertices



Normalized Cut

Normalized cut: Partition graph into two sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum & sizes of A and B are very similar.

Let $\text{vol}(A) = \sum_{i \in A} d_i$, where $d_i = \sum_{j=1}^{N} w_{ij}$. Define the objective function as

$$\text{Ncut}(A, B) := \text{cut}(A, B) \left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)} \right)$$

Normalized Cut

Normalized cut: Partition graph into two sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum & sizes of A and B are very similar.

Let $\text{vol}(A) = \sum_{i \in A} d_i$, where $d_i = \sum_{j=1}^{N} w_{ij}$. Define the objective function as

$$\text{Ncut}(A, B) := \text{cut}(A, B) \left(\frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)} \right)$$

- Ncut is NP-hard to solve
- Spectral clustering is a relaxation

Degree Matrix and Graph Laplacian

- Given a graph with similarity matrix

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{bmatrix}$$

- The degree matrix of the graph is defined as

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_N \end{bmatrix}$$

where $d_j = \sum_{i=1}^{N} w_{ij}$. d_j is the degree of vertex j of the graph.

- The graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$

Normalized Cut and Graph Laplacian (optional)

Recall $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_N)$

$$\text{Let } \mathbf{u} = [u_1, u_2, \ldots, u_N]^\top \quad \text{with} \quad u_i = \begin{cases} \frac{1}{\mathrm{vol}(A)}, & \text{if } i \in A \\ -\frac{1}{\mathrm{vol}(B)}, & \text{if } i \in B \end{cases}$$

$$\mathbf{u}^\top \mathbf{L} \mathbf{u} = \frac{1}{2} \sum_{ij} w_{ij} (u_i - u_j)^2 = \sum_{i \in A, j \in B} w_{ij} \left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)} \right)^2$$

$$\mathbf{u}^\top \mathbf{D} \mathbf{u} = \sum_i d_i u_i^2 = \sum_{i \in A} \frac{d_i}{\mathrm{vol}(A)^2} + \sum_{j \in B} \frac{d_j}{\mathrm{vol}(B)^2} = \frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)}$$

Normalized Cut and Graph Laplacian (optional)

Recall $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_N)$

$$\text{Let } \mathbf{u} = [u_1, u_2, \ldots, u_N]^\top \quad \text{with} \quad u_i = \begin{cases} \frac{1}{\mathrm{vol}(A)}, & \text{if } i \in A \\ -\frac{1}{\mathrm{vol}(B)}, & \text{if } i \in B \end{cases}$$

$$\mathbf{u}^\top \mathbf{L} \mathbf{u} = \frac{1}{2} \sum_{ij} w_{ij}(u_i - u_j)^2 = \sum_{i \in A, j \in B} w_{ij} \left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)} \right)^2$$

$$\mathbf{u}^\top \mathbf{D} \mathbf{u} = \sum_i d_i u_i^2 = \sum_{i \in A} \frac{d_i}{\mathrm{vol}(A)^2} + \sum_{j \in B} \frac{d_j}{\mathrm{vol}(B)^2} = \frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)}$$

Then we have

$$\frac{\mathbf{u}^\top \mathbf{L} \mathbf{u}}{\mathbf{u}^\top \mathbf{D} \mathbf{u}} = \sum_{i \in A, j \in B} w_{ij} \left(\frac{1}{\mathrm{vol}(A)} + \frac{1}{\mathrm{vol}(B)} \right) = \mathrm{Ncut}(A, B)$$

Normalized Cut and Graph Laplacian

Ncut is equivalent to the minimization of $\frac{\mathbf{u}^\top \mathbf{L}\mathbf{u}}{\mathbf{u}^\top \mathbf{D}\mathbf{u}}$, i.e.,

$$\min_{A,B} \text{Ncut}(A,B) \Longleftrightarrow \min_{\mathbf{u}} \frac{\mathbf{u}^\top \mathbf{L}\mathbf{u}}{\mathbf{u}^\top \mathbf{D}\mathbf{u}}, \quad \mathbf{u} \in \mathbb{R}^N,\ u_i = \begin{cases} \frac{1}{\text{vol}(A)}, & \text{if } i \in A \\ -\frac{1}{\text{vol}(B)}, & \text{if } i \in B \end{cases}$$

[1] Detailed derivation can be found in: *Shi and Malik. Normalized Cuts and Image Segmentation. 2000.*

Normalized Cut and Graph Laplacian

Ncut is equivalent to the minimization of $\frac{\mathbf{u}^\top \mathbf{L}\mathbf{u}}{\mathbf{u}^\top \mathbf{D}\mathbf{u}}$, i.e.,

$$\min_{A,B} \text{Ncut}(A,B) \Longleftrightarrow \min_{\mathbf{u}} \frac{\mathbf{u}^\top \mathbf{L}\mathbf{u}}{\mathbf{u}^\top \mathbf{D}\mathbf{u}}, \quad \mathbf{u} \in \mathbb{R}^N,\ u_i = \begin{cases} \frac{1}{\text{vol}(A)}, & \text{if } i \in A \\ -\frac{1}{\text{vol}(B)}, & \text{if } i \in B \end{cases}$$

Equivalent to[1]: $\min_{\mathbf{u}} \frac{\mathbf{u}^\top \mathbf{L}\mathbf{u}}{\mathbf{u}^\top \mathbf{D}\mathbf{u}} \quad \text{s.t.} \quad \mathbf{u}^\top \mathbf{D}\mathbf{1} = \mathbf{0},\ u_i \in \{1, -b\}$

* *b* is some positive constant.

Relaxation: $\mathbf{u}-$second eigenvector of generalized eigenvalue problem

$$\mathbf{L}\mathbf{u} = \lambda \mathbf{D}\mathbf{u}$$

Obtain cluster assignments by thresholding **u** at 0

[1]Detailed derivation can be found in: *Shi and Malik. Normalized Cuts and Image Segmentation. 2000.*

Normalized Cut and Graph Laplacian

$$\min_{A,B} \text{Ncut}(A,B) \iff \min_{\mathbf{u}} \frac{\mathbf{u}^\top \mathbf{L}\mathbf{u}}{\mathbf{u}^\top \mathbf{D}\mathbf{u}} \quad \text{s.t.} \quad \mathbf{u}^\top \mathbf{D}\mathbf{1} = \mathbf{0}, \; u_i \in \{1, -b\}$$

- Relaxation: Let $\mathbf{u}$ be the eigenvector corresponding to the second smallest eigenvalue of the generalized eigenvalue problem $\mathbf{L}\mathbf{u} = \lambda \mathbf{D}\mathbf{u}$
- Equivalent to eigenvector corresponding to the second smallest eigenvalue of the normalized Laplacian

$$\widetilde{\mathbf{L}} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$$

Normalized Cut and Graph Laplacian

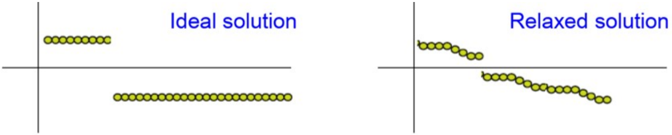
$$\min_{A,B} \text{Ncut}(A,B) \iff \min_{\mathbf{u}} \frac{\mathbf{u}^\top \mathbf{L}\mathbf{u}}{\mathbf{u}^\top \mathbf{D}\mathbf{u}} \quad \text{s.t. } \mathbf{u}^\top \mathbf{D}\mathbf{1} = \mathbf{0},\ u_i \in \{1, -b\}$$

- Relaxation: Let **u** be the eigenvector corresponding to the second smallest eigenvalue of the generalized eigenvalue problem $\mathbf{L}\mathbf{u} = \lambda \mathbf{D}\mathbf{u}$
- Equivalent to eigenvector corresponding to the second smallest eigenvalue of the normalized Laplacian

$$\widetilde{\mathbf{L}} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$$

- Obtain binary partition as follows:

$$i \in A \quad \text{if } u_i \geq 0, \quad i \in B \quad \text{if } u_i < 0$$



- It can be extended to multiple clusters $\longrightarrow$ Spectral Clustering

Spectral Clustering Algorithm

Input: data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, number K of clusters

- Step 1. Construct a similarity matrix $\mathbf{W}$
  e.g. use $w_{ij} = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right)$

  k-nearest neighbor graph, or ϵ-neighborhood graph

- Step 2. Compute the Laplacian matrix $\mathbf{L}$ (or normalized $\mathbf{L}$)
  - $\mathbf{L} = \mathbf{D} - \mathbf{W}$
  - $\widetilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$ (normalized)
  - $\widehat{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ (symmetric normalized, recommended)

Spectral Clustering Algorithm

- Step 3. Perform eigenvalue decomposition on **L** (or normalized **L**) and use the first K eigenvectors to form a matrix **Z**

$$\widehat{\mathbf{L}} = \mathbf{V}\Sigma\mathbf{V}^\top, \quad \mathbf{Z} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K]^\top \in \mathbb{R}^{K \times N}$$

- Step 4. Normalize the columns of **Z** to unit L_2 norm , i.e.,

$$\mathbf{z}_i \leftarrow \mathbf{z}_i / \left\| \mathbf{z}_i \right\|, \quad i = 1, \ldots, N$$

Spectral Clustering Algorithm

- Step 3. Perform eigenvalue decomposition on **L** (or normalized **L**) and use the first K eigenvectors to form a matrix **Z**

$$\widehat{\mathbf{L}} = \mathbf{V}\Sigma\mathbf{V}^\top, \quad \mathbf{Z} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K]^\top \in \mathbb{R}^{K\times N}$$

- Step 4. Normalize the columns of **Z** to unit L_2 norm , i.e.,

$$\mathbf{z}_i \leftarrow \mathbf{z}_i / \|\mathbf{z}_i\|, \quad i = 1, \ldots, N$$

- Step 5. Perform K-means on $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}$

Output: K of clusters of **Z** or **X**

Property of Graph Laplacian Matrix

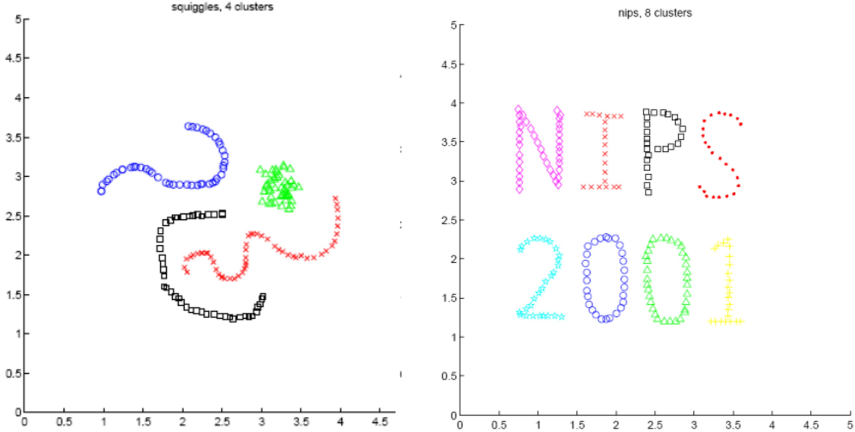
$$\mathbf{L} = \mathbf{D} - \mathbf{W} \quad \text{or} \quad \widehat{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$$

- Symmetric and positive semi-definite
- The eigenvalues satisfy

$$0 = \lambda_1 \le \lambda_2 \le \lambda_3 \le \cdots \le \lambda_{N-1} \le \lambda_N$$

- If the number of zero eigenvalues is K, the graph has K connected components, corresponding to K clusters.

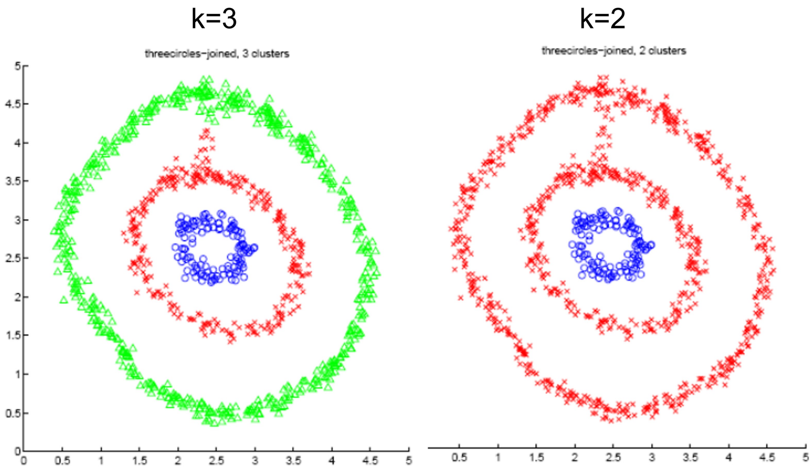
Examples of Spectral Clustering



Images from Ng et al. 2001

Examples of Spectral Clustering

- Influence of K

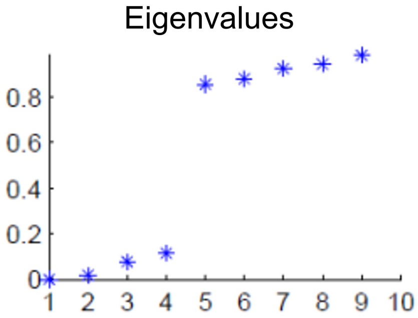


Images from Ng et al. 2001

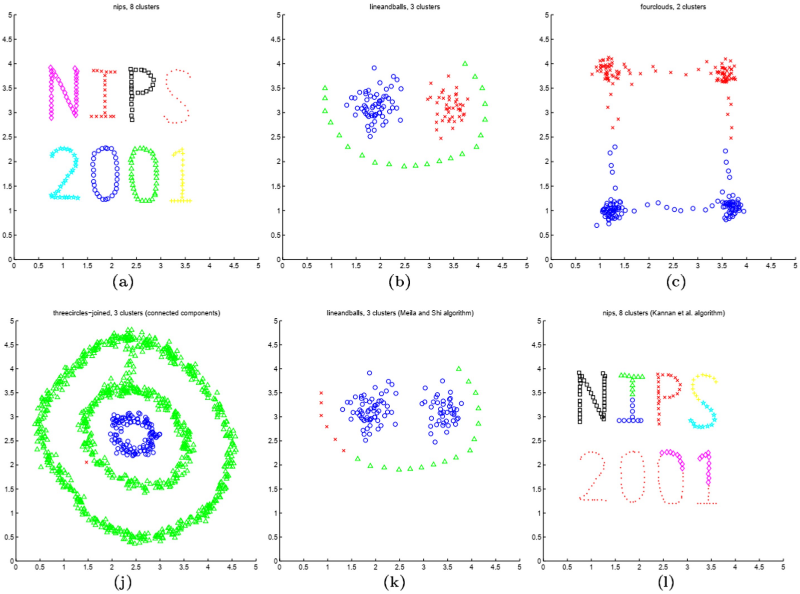
Determine K in Spectral Clustering

- Use the k that maximizes the eigengap (difference between consecutive eigenvalues)

$$\Delta_j = \left|\lambda_{j+1} - \lambda_j\right|, \qquad K^* = \arg\max_j \Delta_j$$



More Examples of Spectral Clustering



Images from Ng et al. 2001

Characteristics of Spectral Clustering[2]

- High clustering accuracy in real applications
  - Often outperform k-means
- High computational cost, not applicable to big data
  - Space complexity: $O(N^2)$
  - Time complexity: $O(N^3)$

[2] More about spectral clustering can be found in: *A Tutorial on Spectral Clustering. Ulrike von Luxburg. 2007.*

Characteristics of Spectral Clustering[2]

- High clustering accuracy in real applications
  - Often outperform k-means
- High computational cost, not applicable to big data
  - Space complexity: $O(N^2)$
  - Time complexity: $O(N^3)$
- Not easy to determine the similarity matrix
  - *kNN*, ϵ-neighborhood, Gaussian kernel, etc
  - Which method and what hyperparameter?

[2] More about spectral clustering can be found in: *A Tutorial on Spectral Clustering. Ulrike von Luxburg. 2007.*

Learning Outcomes

- Know the definitions of cut and Ncut
- Know the main steps of spectral clustering
- Know the property of graph Laplacian matrix
- Know the advantage and disadvantage of spectral clustering