

# A Tutorial on Spectral Clustering

By Ulrike von Luxburg, 2007

Max Planck Institute for Biological Cybernetics

Presented by Samy Coulombe, Shenyang (Andy) Huang

2020.3.24

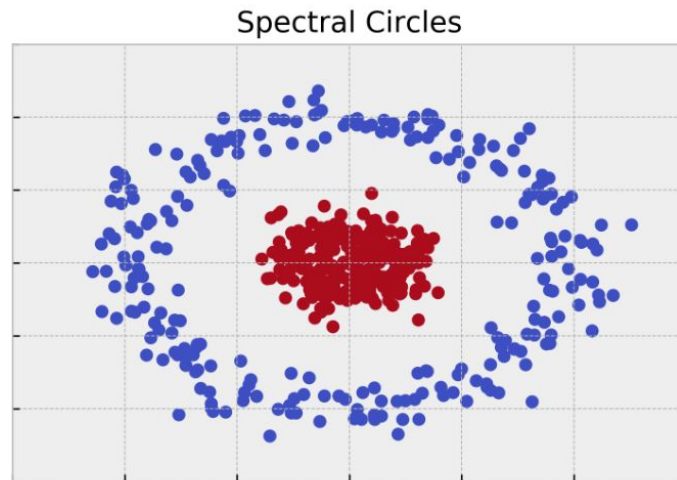
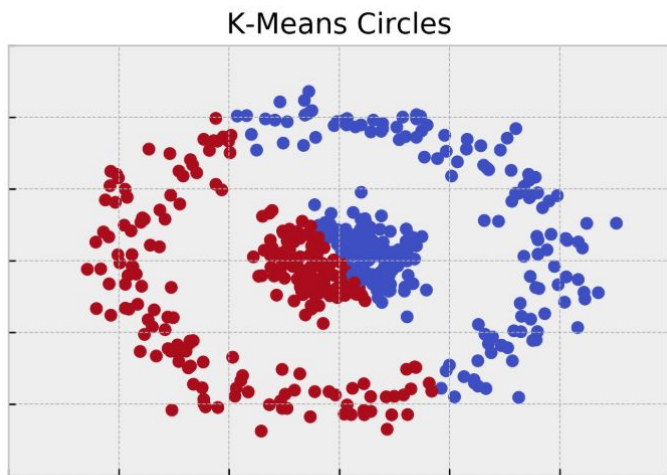
# Agenda

1. Motivation for the paper
2. Intro to similarity graphs and notation
3. Graph Laplacian and their basic properties
4. Spectral Clustering Algorithm
5. Connection to Perturbation theory

# Motivation and background (2007)

**Aim: a self-contained introduction to spectral clustering.**

- Graph Laplacian and Spectral Clustering algorithm
- Spectral Clustering and its connections to various areas



*Comparison between results of k-means clustering and spectral clustering  
(Colour indicates cluster assignment) [2]*

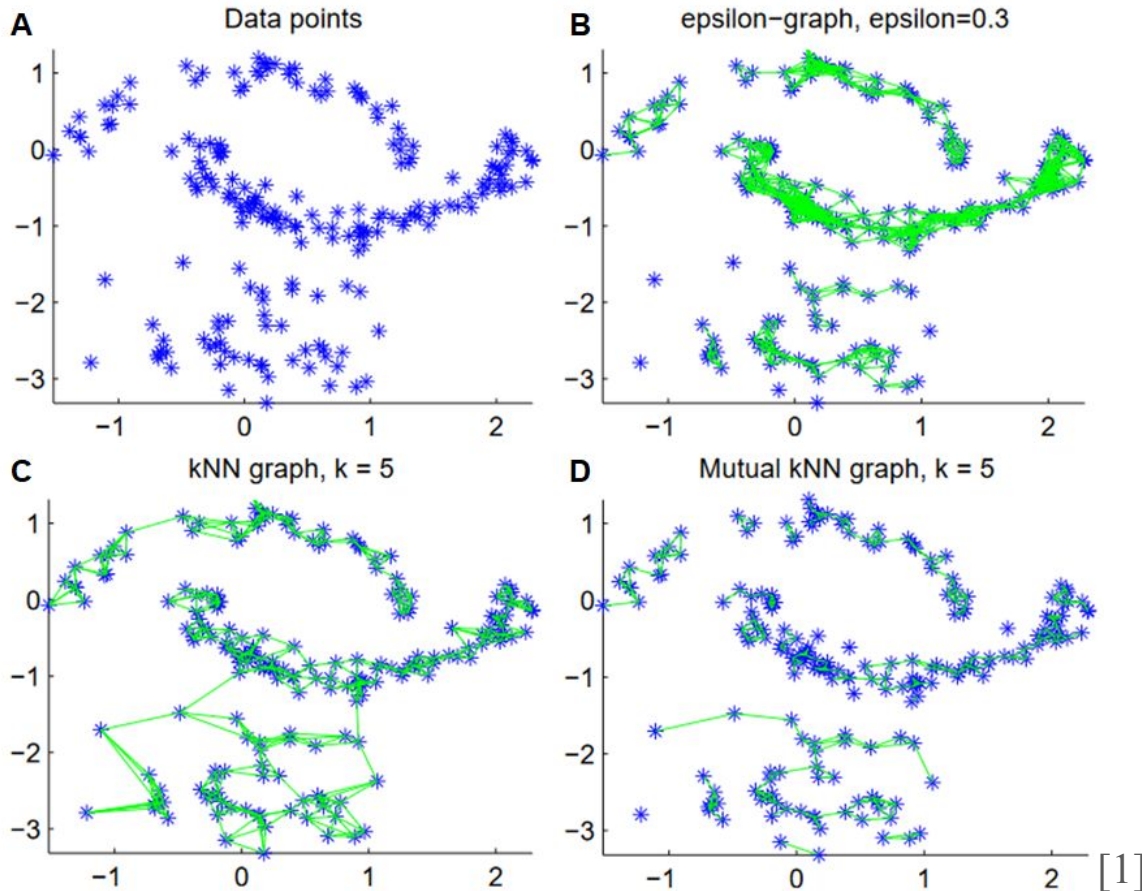
# Clustering Setting

**Goal:** 1). High within-group similarity,  
2). Low between-group similarity

- Given a set of data points,  $x_1, x_2, \dots, x_n$  and a similarity measure\*
  - $S : x \in \mathbb{R}^d, y \in \mathbb{R}^d \rightarrow \mathbb{R}^+$  ( $S_{i,j} \geq 0$ ) between all pairs of data points
- Construct a similarity graph:  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$

\*domain dependent choice

# Types of similarity graphs



## Legend

- A) Data points
- B)  **$\epsilon$ -neighborhood graph:**  
edge  $(v_i, v_j)$  exists if  
 $\text{distance}(v_i, v_j) \leq \epsilon$
- C) **kNN graph:**  
edge  $(v_i, v_j)$  exists if  
 $v_j$  is among  $v_i$ 's  $k$  nearest  
nodes
- D) **Mutual kNN graph:**  
edge  $(v_i, v_j)$  exists if  
 $v_j$  is among  $v_i$ 's  $k$  nearest  
nodes and vice versa.

# Similarity Graphs and the Graph Laplacian

From the similarity graph  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ , we can define the following:

- $\mathbf{G}$ 's weighted adjacency matrix  $\mathbf{W}$
- The degree of a node  $i$ ,  $d_i = \sum_{j=1}^n w_{i,j}$ , and  $\mathbf{G}$ 's diagonal degree matrix  $\mathbf{D}_{ii} = d_i$

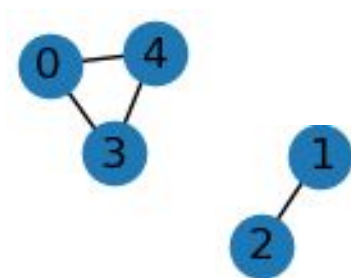
---

**Graph Laplacian ( $\mathbf{L}$ ) of  $\mathbf{G}$ :**  $\mathbf{L} = \mathbf{D} - \mathbf{W}$

- $\mathbf{D}$  is the degree matrix
- $\mathbf{W}$  is the weighted adjacency matrix

# Similarity Graphs and the Graph Laplacian

From the similarity graph  $G = (V, E)$ ,



- Degree matrix  $\mathbf{D}$ ,

$$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

-

- Adjacency matrix  $\mathbf{W}$ ,

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

=

- Graph Laplacian  $\mathbf{L}$

$$\begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$

# Properties of Graph Laplacian

**L** satisfies the following properties:

- vector  $f \in \mathbb{R}^n$ :

- $f^\top Lf = \frac{1}{2} \sum_{i,j=1}^n w_{i,j} (f_i - f_j)^2$

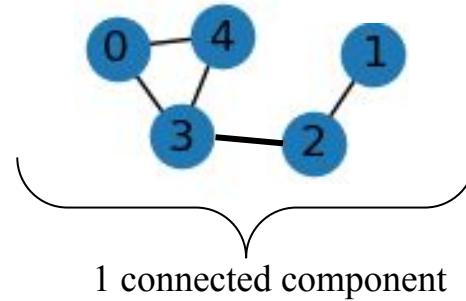
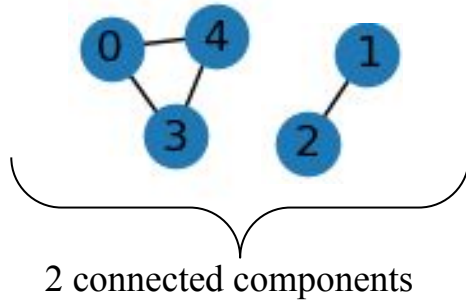
- **L** is symmetric and positive semi-definite
- **L** eigenvalues  $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$



# 0 eigenvalues and the connected components

**Proposition (number of connected components):** Let  $G$  be an undirected graph with non-negative weight matrix  $\mathbf{W}$ . Then,

- The multiplicity  $k$  of the eigenvalue 0 of  $L$  equals the number of connected components  $C_1, \dots, C_k$  in the graph.



- The eigenspace of eigenvalue 0 is spanned by  $k$  **indicator vectors** which map the  $n$  nodes to the  $k$  clusters.

# Proof

**Base Case:  $k = 1$  (connected graph)**

Assuming  $f$  is an **eigenvector with eigenvalue 0**.

By definition of eigenvalue and eigenvector:

$$Lf = \lambda f = 0 \cdot f = 0$$

We know from the properties of the Laplacian that:

$$f^\top Lf = \sum_{i,j=1}^n w_{i,j} (f_i - f_j)^2 = 0$$

# Proof

## Base Case: $k = 1$ (connected graph)

Assuming  $f$  is an eigenvector with eigenvalue 0. We know from property of Laplacian that:

$$f^\top L f = \sum_{i,j=1}^n w_{i,j} (f_i - f_j)^2 = 0$$

- If vertices  $v_i$  and  $v_j$  are connected ( $w_{i,j} > 0$ ), then  $f_i = f_j$
- Because all nodes are connected by a path, we must have  $f_1 = f_2 = \dots = f_n$
- Therefore  $f$  is a constant **normalized one vector**  $\mathbf{1}$  for a **connected component**

# Proof continued

## Case: $k$ connected components

- Assume the vertices are ordered according to the connected components they belong to.
- Weighted adjacency  $\mathbf{W}$  and Laplacian  $\mathbf{L}$  have a **block diagonal form**.

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{W}_k \end{bmatrix} \quad \mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{L}_k \end{bmatrix}$$

- Each  $L_i$  is the Laplacian of  **$i$ -th connected component**

# Proof continued

## Case: $k$ connected components

- both adjacency  $\mathbf{W}$  and Laplacian  $\mathbf{L}$  have a block diagonal form.

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{L}_k \end{bmatrix}$$

- The spectrum of block diagonal matrix  $\mathbf{L}$  is **the union of the spectrums** of  $L_i$
- From base case, every  $L_i$  has eigenvalue 0 with multiplicity 1, and the corresponding eigenvector is an **indicator vector** for the  $i$ -th connected component.

# Spectral Clustering algorithm

Input: Similarity matrix  $S \in \mathbb{R}^{n \times n}$  , number of clusters  $k$

1. Compute the Laplacian matrix  $L$
2. **Compute the first  $k$  smallest non-zero eigenvectors**  $v_1, \dots, v_k$
3. Let  $\mathbf{V} \in \mathbb{R}^{n \times k}$  be the matrix formed by the eigenvectors, each row vector  $y_i \in \mathbb{R}^k$  and these form data points to cluster  $C_1, \dots, C_k$
4. Cluster  $y_i$  with **k-means algorithm** into clusters

Output: Clusters  $(A_1, A_2, \dots, A_k)$  where  $A_i = \{j | y_j \in C_i\}$

# Spectral Clustering algorithm

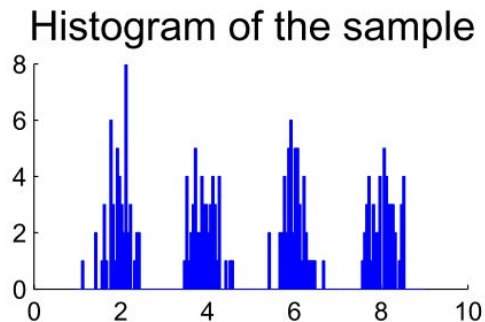
## Core idea of Spectral Clustering:

Project dataset into a nice **low dimensional embedding space**, then cluster

## Questions you might have:

1. Why use the  $k$  smallest eigenvectors of the Laplacian matrix?
  - a. See upcoming slides
2. Why use k-means for clustering?
  - a. Actually any other clustering algorithm can be used
  - b. But the Euclidean distance assumption of k-means is actually motivated here

# Spectral Clustering Algorithm in Action



**Toy example:** 200 random points drawn from a mixture of 4 Gaussians

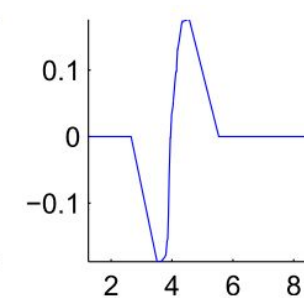
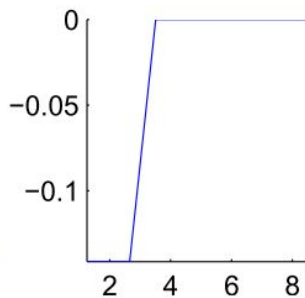
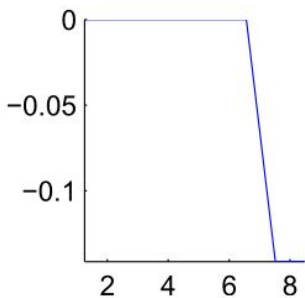
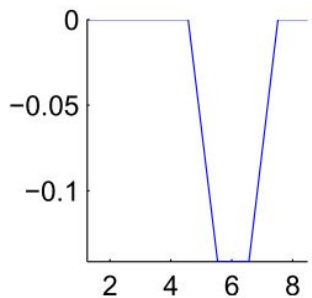
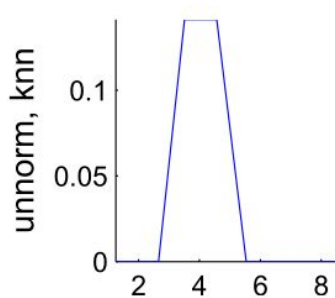
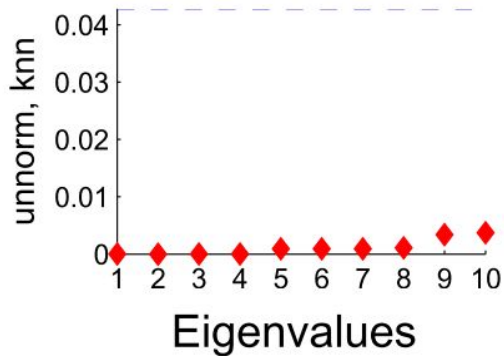
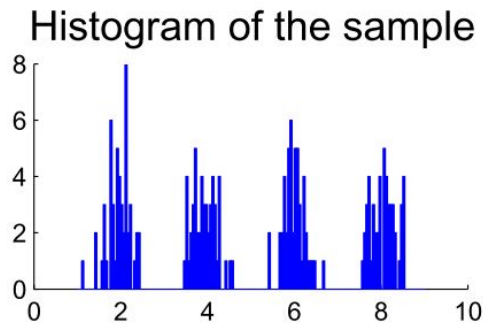
**Similarity function:** Gaussian similarity  $s(x_i, x_j) = \exp\left(-\frac{|x_i - x_j|^2}{2\sigma^2}\right)$  (non-Euclidean)

Construct K-nearest neighbor graph

spectral clustering using k-means detected the correct clusterings



# Spectral Clustering Algorithm in Action: Eigenvalue and eigenvectors of Laplacian



Eigenvector 1

Eigenvector 2

Eigenvector 3

Eigenvector 4

Eigenvector 5

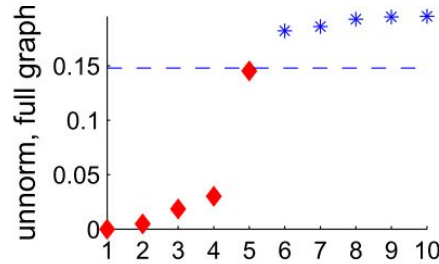
# Perturbation theory point of view

- ideally:  **$k$  disconnected components**
- reality: **a connected graph** with small between-cluster connectivity
- **$k$  smallest eigenvectors still similar to ideal case**

Consider a perturbed symmetric matrix  $\tilde{A} = A + H$

# Davis-Kahan Theorem

- **Distance between two Eigenspaces**  $V_1, \tilde{V}_1 : d(V_1, \tilde{V}_1)$
- **Spectral Gap** is  $\delta = \min\{|\lambda - s|; \lambda \text{ is eigenvalue of } A, \lambda \notin S_1, s \in S_1\}$



- **Davis-Kahan theorem** tells us that:  $d(V_1, \tilde{V}_1) \leq \frac{\|H\|}{\delta}$ 
  - The norm is the Frobenius norm or the Two-norm

# Comments on perturbation theory

For spectral clustering to work properly

- **There should be an ideal case** where  $A$  is block diagonal
- $A$  should be a symmetric matrix
- The entries of eigenvectors **away from 0**.

# More covered in the paper

1. Connection to **Graph Cut** problems

$$\text{RatioCut}(A_1, \dots, A_k) = \sum_{i=1}^k \frac{\text{cut}(A_i, \overline{A_i})}{|A_i|}$$

2. Symmetric and Random-Walk **Normalized Graph Laplacians** (normalize by degree)
3. Connection to **Random Walk Matrices**
4. Many **practical tips** for spectral clustering

Thank you for listening :)

# References

1. [A Tutorial on Spectral Clustering](#)
2. Image taken from <https://towardsdatascience.com/spectral-clustering-aba2640c0d5b>