Spectral Clustering

PR & Machine Learning – CS5691

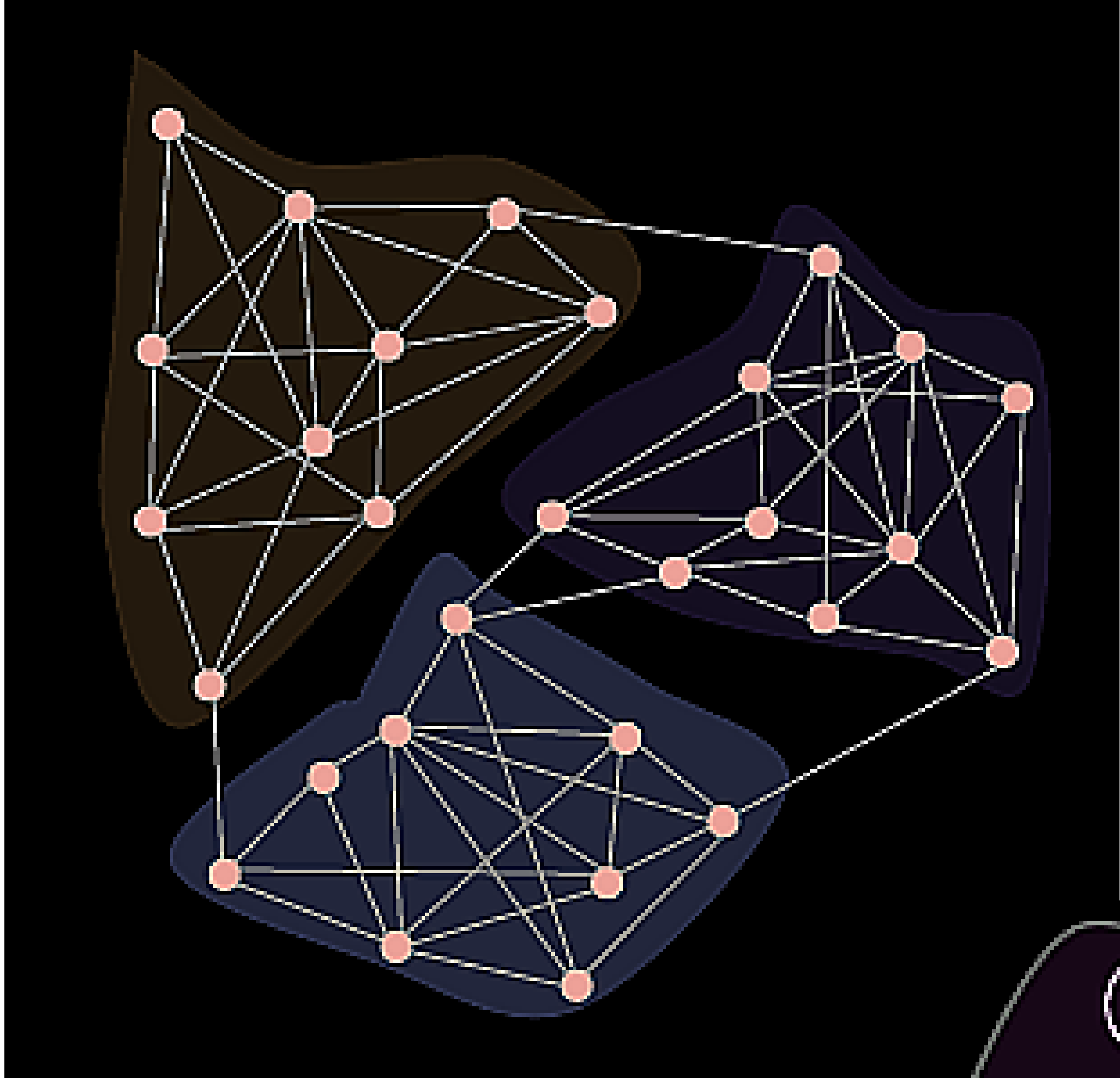
Content Credits:

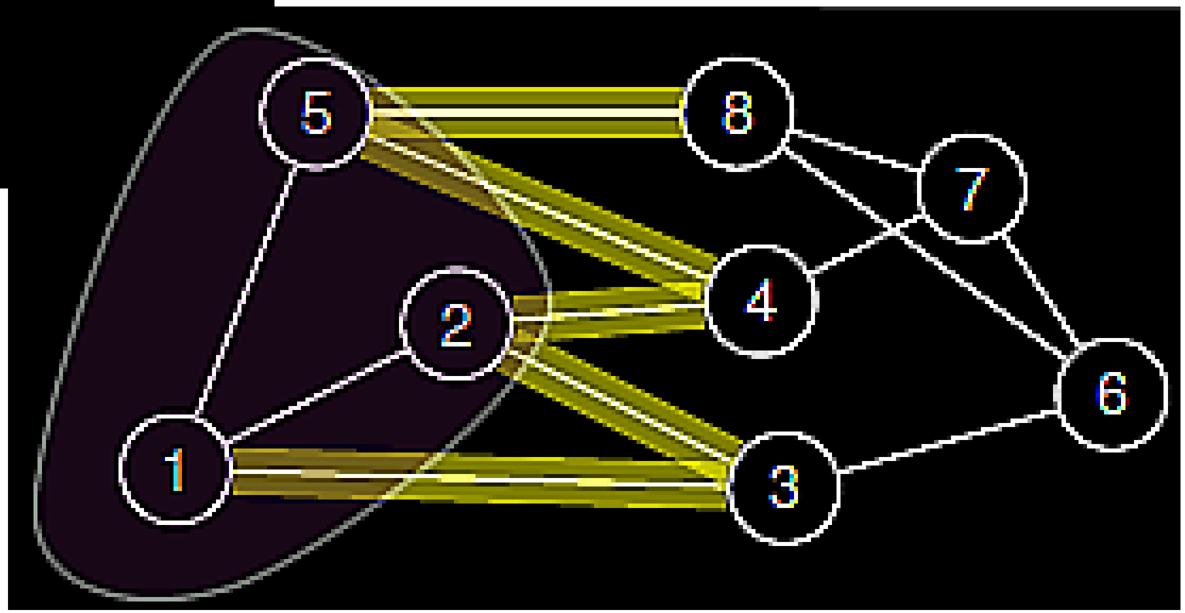
1. Von Luxburg, U., “A tutorial on spectral clustering.”; Statistics and Computing, 17(4), 395-416. Springer (2007); Technical Report No. TR-149, A Tutorial on Spectral Clustering – Aug. 2006.
2. Davide Eynard, “Notes on Spectral Clustering.”; (2012).

Similarity graph

- The objective of a clustering algorithm is partitioning data into groups such that:
  - Points in the **same** group are **similar**
  - Points in **different** groups are **dissimilar**

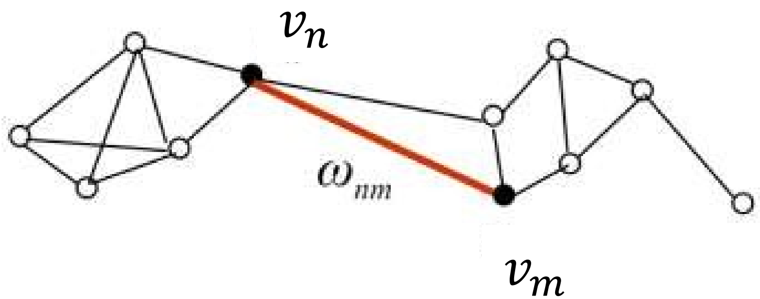
- **Similarity graph** $G = (V, E)$ [undirected graph]:
  - Vertices v_i and v_j are connected by a **weighted** edge if their similarity is above a given threshold
  - GOAL: find a partition of the graph such that:
    - edges **within** a group have **high weights**
    - edges **across** different groups have **low weights**



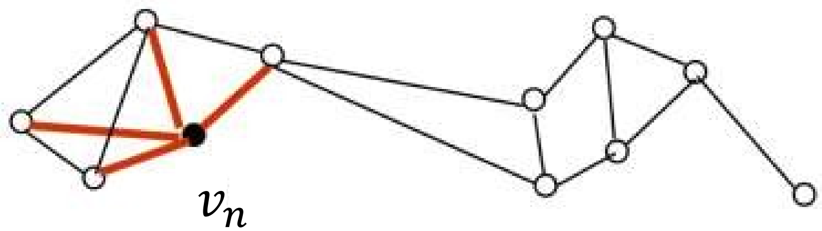

5
8
7
4
2
6
1
3

Weighted adjacency matrix

- Let $G(V, E)$ be an undirected graph with vertex set
$$V = \{v_1, \ldots, v_n\}$$
- **Weighted adjacency matrix** $\boldsymbol{W} = (w_{ij})_{i,j=1,\ldots,n}$
  - $w_{ij} \geq 0$ is the weight of the edge between v_i and v_j.
  - $w_{ij} = 0$ means that v_i and v_j are not connected by an edge
  - $w_{ij} = w_{ji}$



  - **Degree of a vertex** $v_i \in V$: $\boldsymbol{d_i} = \sum_{j=1,\ldots,n} w_{ij}$
  - **Degree matrix** $\boldsymbol{D} = diag(d_1, \ldots, d_n)$

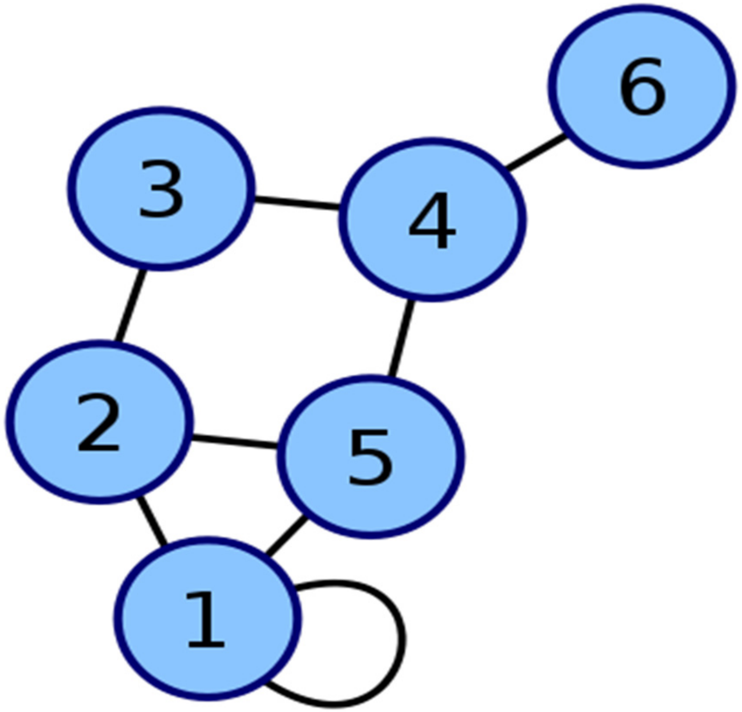


Similarity graphs - variants

- **ε-neighborhood:**
  - Connect all points whose pairwise distance is less than ε
- K-nearest neighbor graph
  - Connect vertex v_i with vertex v_j , if v_j is among the k-nearest neighbours of v_i.
- Fully connected
  - *all* points with similarity $w_{ij} > 0$ are connected.
  - use a similarity function like the **Gaussian**:

$$w_{ij} = w(v_i , v_j) = \exp(-\|v_i - v_j\|^2/(2\sigma^2))$$

The adjacency matrix of a finite graph G on n vertices is the $n \times n$ matrix where the non-diagonal entry a_{ij} is the number of edges from vertex i to vertex j, and the diagonal entry a_{ii}, depending on the convention, is either once (directed) or twice (undirected) the number of edges (loops) from vertex i to itself. In the special case of a finite simple graph, the adjacency matrix is a (0,1)-matrix with zeros on its diagonal. If the graph is undirected, the adjacency matrix is symmetric.



Labeled graph

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

Adjacency matrix

Graph Laplacians

- Graph Laplacian:
  - $\boldsymbol{L} = D - W$ *(symmetric and positive semi-definite)*
- Properties:
  - Smallest eigenvalue $\lambda_1 = 0$ with eigenvector $\mathbb{1}$
  - n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$
  - the multiplicity k of the eigenvalue 0 of $\boldsymbol{L}$ equals the number of connected components $A_1, \dots, A_k$ in the graph.

For every vector $f \in \mathbb{R}^n$ we have

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

Normalized Laplacian:

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2};$$

$$L_{rw} = D^{-1} L = I - D^{-1} W \ \ \text{(random walk)}$$

RANDOM WALKS on GRAPHS

- $G = (V, E)$: a simple connected graph on n vertices
- $A(G)$: the adjacency matrix
- $D(G) = \mathrm{diag}(d_1, d_2, \ldots, d_n)$: the diagonal degree matrix
- $L = D - A$: the combinatorial Laplacian
- L is semi-definite and $\mathbf{1}$ is always an eigenvector for the eigenvalue 0.

Normalized Laplacian: $\mathcal{L} = I - D^{-1/2}AD^{-1/2}$.

- $\mathcal{L}$ is always semi-definite.
- 0 is always an eigenvalue of $\mathcal{L}$ with eigenvector $(\sqrt{d_1}, \ldots, \sqrt{d_n})'$.

- Laplacian eigenvalues: $\lambda_0, \ldots, \lambda_{n-1}$

$$0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1} \leq 2.$$

- $\lambda_{n-1} = 2$ if and only if G is bipartite.
- $\lambda_1 > 1$ if and only if G is the complete graph.

Spectral Clustering algorithm (1)

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

1. Construct a similarity graph as previously described. Let W be its weighted adjacency matrix.
2. Compute the unnormalized Laplacian $\boldsymbol{L}$
3. Compute the first k eigenvectors $u_1, \dots, u_k$ of $\boldsymbol{L}$
4. Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \dots, u_k$ as columns
5. For $i = 1, \dots, n$ let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U
6. Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \dots, C_k$.

- Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j | y_j \in C_i\}$.

Normalized Graph Laplacians

- *Symmetric*: $\boldsymbol{L}_{sym} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}} = I - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{W}\boldsymbol{D}^{-\frac{1}{2}}$
  - $\boldsymbol{L}_{rw} = \boldsymbol{D}^{-1}\boldsymbol{L} = I - \boldsymbol{D}^{-1}W$

- λ is an eigenvalue of L_{sym} with eigenvector u iff λ and u solve the **generalized eigenproblem** $\boldsymbol{L}u = \lambda\boldsymbol{D}u$
- 0 is an eigenvalue of $\boldsymbol{L}_{sym}$ with eigenvector $\boldsymbol{D}^{\frac{1}{2}}$
- $\boldsymbol{L}_{sym}$ and $\boldsymbol{L}_{rw}$ are positive semi-definite and have n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$

Then the multiplicity k of the eigenvalue 0 of both L_{rw} and L_{sym} equals the number of connected components $A_1, \ldots, A_k$ in the graph. For L_{rw}, the eigenspace of 0 is spanned by the indicator vectors $\mathbb{1}_{A_i}$ of those components. For L_{sym}, the eigenspace of 0 is spanned by the vectors $D^{1/2}\mathbb{1}_{A_i}$.

1. *For every* $f \in \mathbb{R}^n$ *we have*

$$f' L_{sym} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 .$$

2. λ *is an eigenvalue of* L_{rw} *with eigenvector* v *if and only if* λ *is an eigenvalue of* L_{sym} *with eigenvector* $w = D^{1/2}v$.

3. λ *is an eigenvalue of* L_{rw} *with eigenvector* v *if and only if* λ *and* v *solve the generalized eigenproblem* $Lv = \lambda Dv$.

4. 0 *is an eigenvalue of* L_{rw} *with the constant one vector* $\mathbb{1}$ *as eigenvector.* 0 *is an eigenvalue of* L_{sym} *with eigenvector* $D^{1/2}\mathbb{1}$.

5. L_{sym} *and* L_{rw} *are positive semi-definite and have* n *non-negative real-valued eigenvalues* $0 = \lambda_1 \leq \ldots \leq \lambda_n$.

Spectral Clustering algorithm (2)

- Input: Similarity matrix S $\in \mathbb{R}^{n\times n}$, number k of clusters to construct.
  1. Construct a similarity graph as previously described. Let W be its weighted adjacency matrix.
  2. Compute the normalized Laplacian L_{sym}
  3. Compute the first k eigenvectors $u_1, \ldots, u_k$ of $\boldsymbol{L}_{sym}$.
  4. normalize the eigenvectors
- Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | y_j \in C_i\}$.

Unnormalized spectral clustering

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

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $v_1, \dots, v_k$ of L.**
- Let $V \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $v_1, \dots, v_k$ as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of V.
- Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \dots, C_k$.

Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

Normalized spectral clustering according to Shi and Malik (2000)

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

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $v_1, \dots, v_k$ of the generalized eigenproblem $Lv = \lambda Dv$.**
- Let $V \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $v_1, \dots, v_k$ as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of V.
- Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \dots, C_k$.

Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

Normalized spectral clustering according to Ng, Jordan, and Weiss (2002)

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

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the normalized Laplacian L_{sym}.
- **Compute the first k eigenvectors $v_1, \ldots, v_k$ of L_{sym}.**
- Let $V \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $v_1, \ldots, v_k$ as columns.
- **Form the matrix $U \in \mathbb{R}^{n \times k}$ from V by normalizing the row sums to have norm 1,** that is $u_{ij} = v_{ij} / (\sum_k v_{ik}^2)^{1/2}$.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

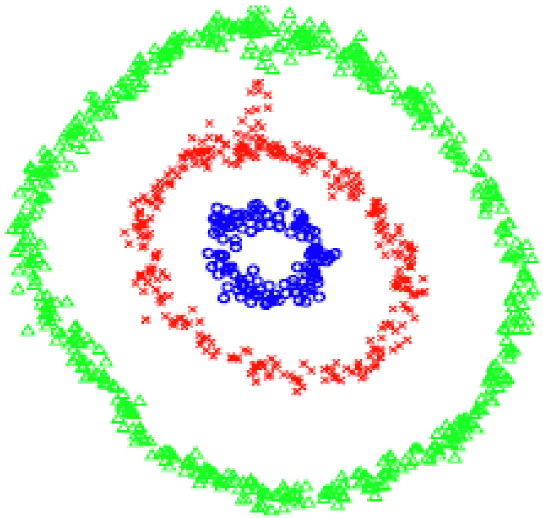
Unnormalized spectral clustering

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

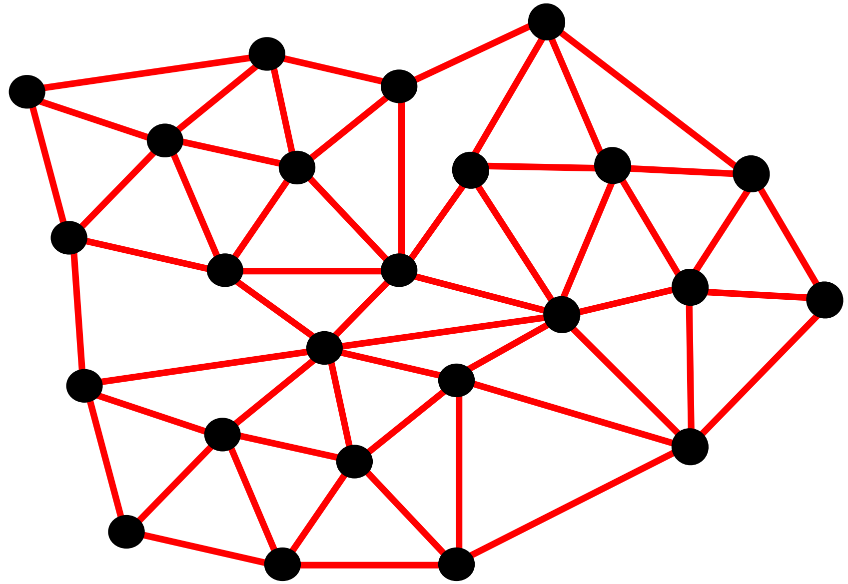
- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $v_1, \ldots, v_k$ of L.**
- Let $V \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $v_1, \ldots, v_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of V.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

spectral clustering (a la Ng-Jordan-Weiss)



data



similarity graph

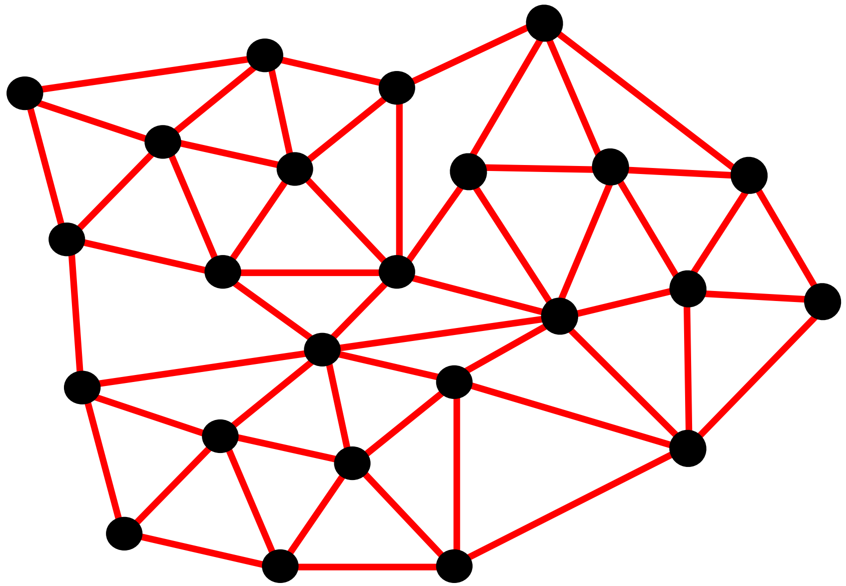
edges have weights **w(i,j)**

e.g. $w(i,j) = e^{-\|x_i - x_j\|^2}$

$$A_{ij} = w(i,j)$$

$$A_{ij} = w(i,j)$$

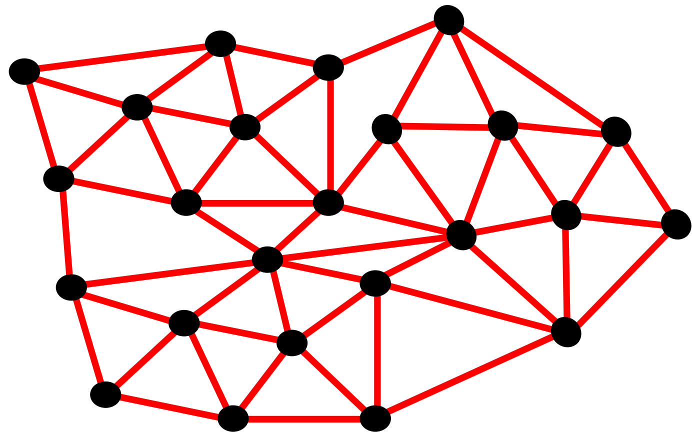
$$D_{ii} = \sum_{j=1}^{n} w(i,j)$$



diagonal matrix **D**

Normalized Laplacian: $L = I - D^{-1/2} A D^{-1/2}$

$$v^T L v = \sum_{\{i,j\} \in E} w(i,j)(v_i - v_j)^2$$



Normalized Laplacian: $L = I - D^{-1/2} A D^{-1/2}$

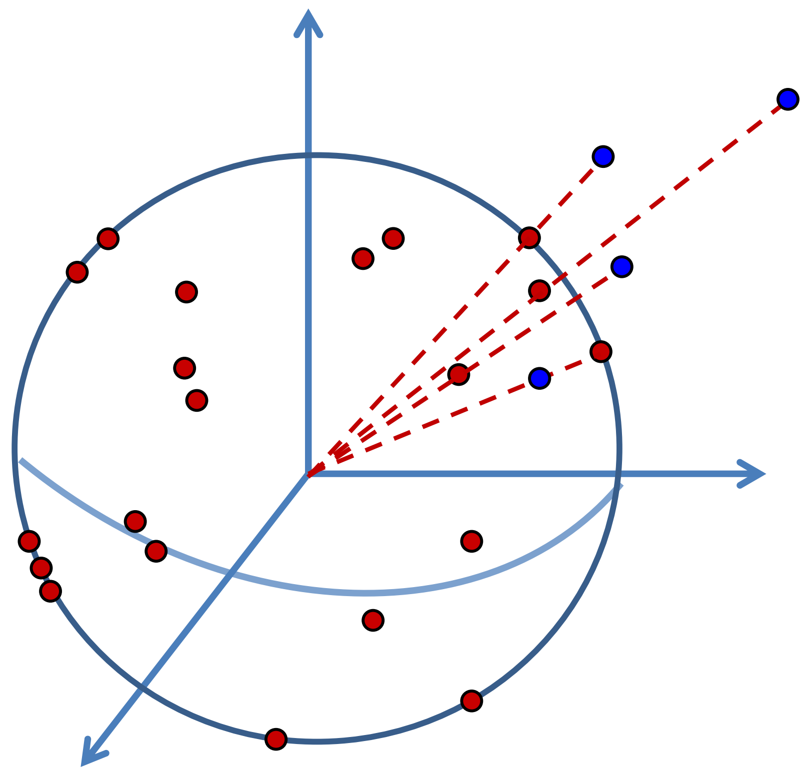
Normalized Laplacian: $L = I - D^{-1/2} A D^{-1/2}$

Compute first **k** eigenvectors: $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_k}$

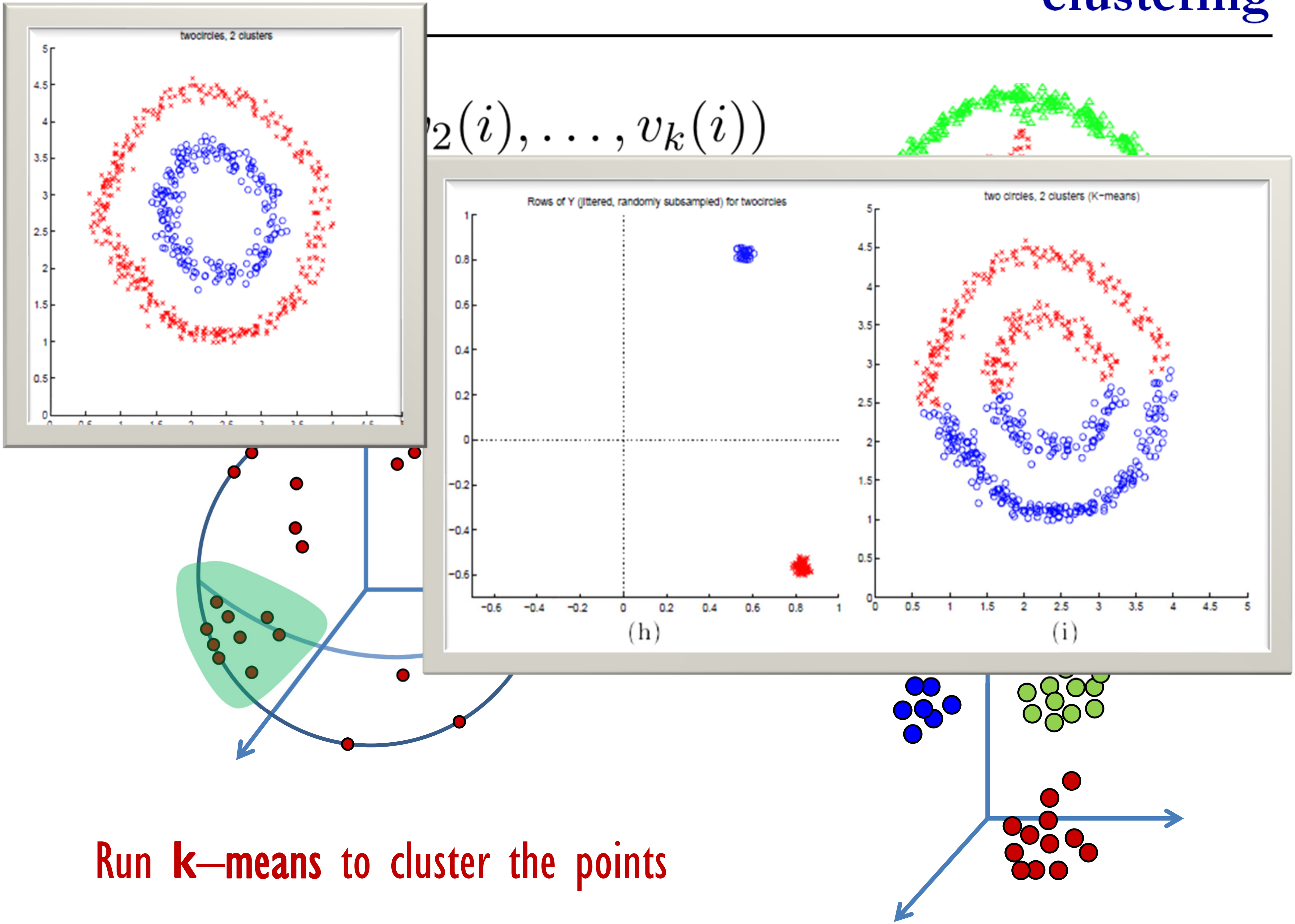
$$Lv_j = \lambda_j v_j \qquad \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k$$

$$F(i) = \frac{(v_1(i), v_2(i), \ldots, v_k(i))}{\sqrt{\sum_{j=1}^{k} v_j(i)^2}}$$

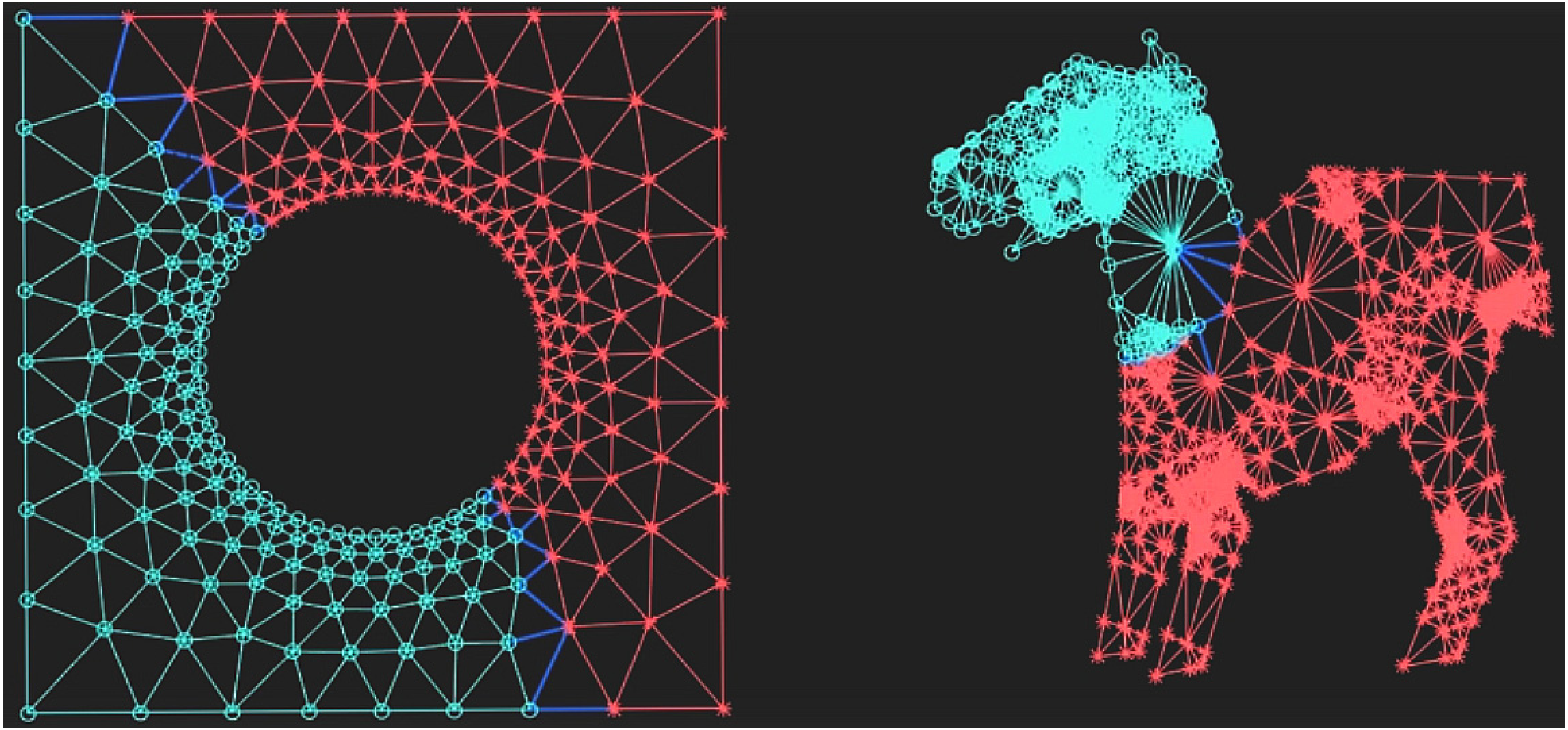
$$F : V \to \mathbb{R}^k$$



$${}_2(i), \ldots, v_k(i))$$



Run **k**–means to cluster the points



https://www.youtube.com/watch?v=FtgU4xDJzEk