


SPECTRAL AND GRAPH CLUSTERING



- 
- clustering over graph data
 - goal is to cluster the nodes by using the edges and their weights, which represent the similarity between the incident nodes.
 - Graph clustering also has a very strong connection to spectral decomposition of graph-based matrices

Graphs and Matrices

Given a dataset $\mathbf{D} = \{\mathbf{x}_i\}_{i=1}^n$ consisting of n points in \mathbb{R}^d , let \mathbf{A} denote the $n \times n$ symmetric *similarity matrix* between the points, given as

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

where $\mathbf{A}(i,j) = a_{ij}$ denotes the similarity or affinity between points \mathbf{x}_i and \mathbf{x}_j .

We require the similarity to be symmetric and non-negative, that is, $a_{ij} = a_{ji}$ and $a_{ij} \geq 0$, respectively.

The matrix \mathbf{A} is the *weighted adjacency matrix* for the data graph. If all affinities are 0 or 1, then \mathbf{A} represents the regular adjacency relationship between the vertices.

example

Iris dataset

$n = 150$ points $\mathbf{x}_i \in \mathbb{R}^4$

$$a_{ij} = \exp \left\{ -\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right\}$$

top q nearest neighbors

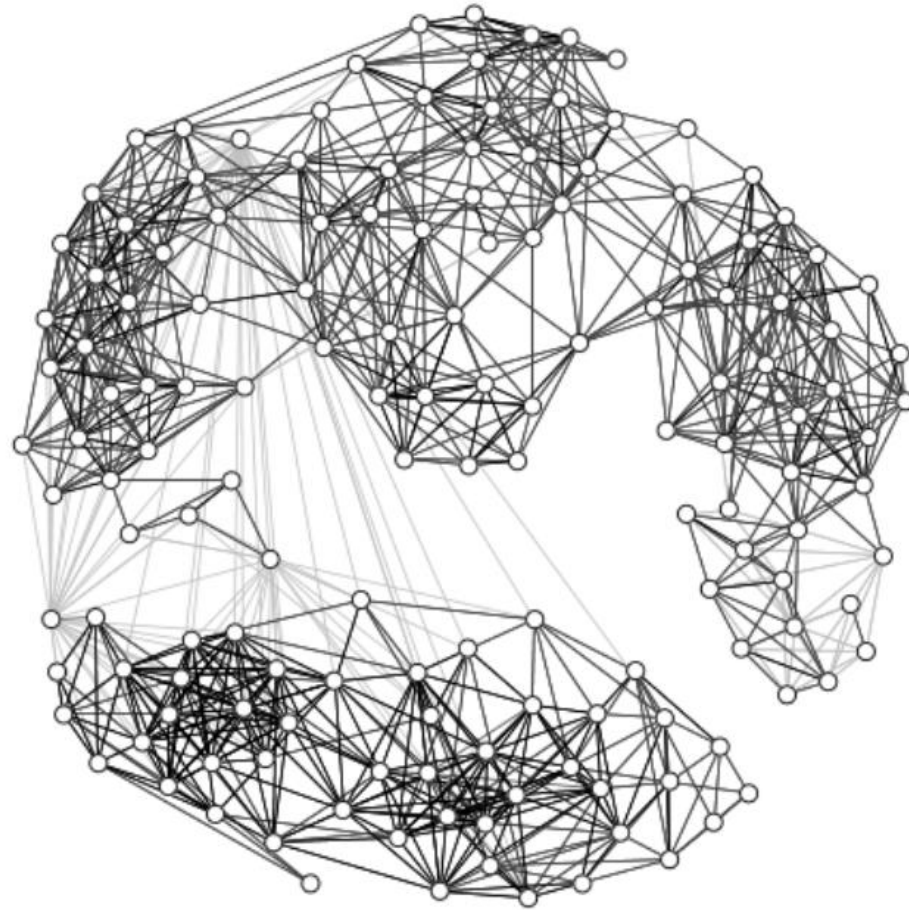
a_{iq} represents the similarity value between \mathbf{x}_i and its q th nearest neighbor

$q = 16$

$$N_q(\mathbf{x}_i) = \{\mathbf{x}_j \in V : a_{ij} \leq a_{iq}\}$$

An edge is added between nodes \mathbf{x}_i and \mathbf{x}_j if and only if both nodes are mutual nearest neighbors

Iris Graph



Degree Matrix

For a vertex x_i , let d_i denote the *degree* of the vertex, defined as

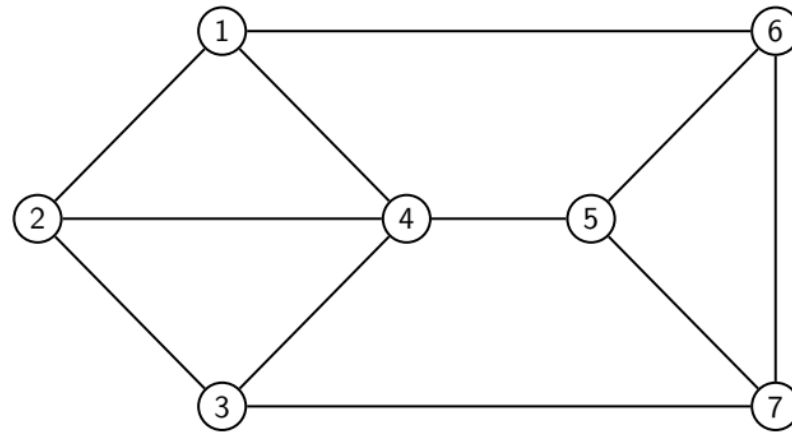
$$d_i = \sum_{j=1}^n a_{ij}$$

We define the *degree matrix* Δ of graph G as the $n \times n$ diagonal matrix:

$$\Delta = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^n a_{1j} & 0 & \cdots & 0 \\ 0 & \sum_{j=1}^n a_{2j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sum_{j=1}^n a_{nj} \end{pmatrix}$$

Δ can be compactly written as $\Delta(i, i) = d_i$ for all $1 \leq i \leq n$.

example



Graph Laplacian Matrix

The *Laplacian matrix* of a graph is defined as

$$\mathbf{L} = \mathbf{\Delta} - \mathbf{A}$$

Graph Laplacian Matrix

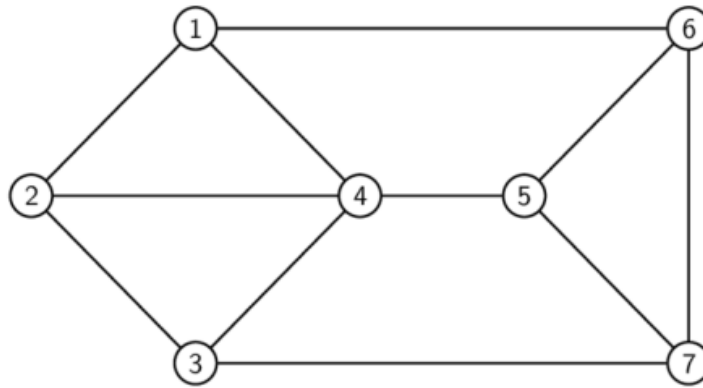
- L is a symmetric, positive semidefinite matrix

$$\mathbf{c}^T \mathbf{L} \mathbf{c} = \mathbf{c}^T (\mathbf{\Delta} - \mathbf{A}) \mathbf{c} = \mathbf{c}^T \mathbf{\Delta} \mathbf{c} - \mathbf{c}^T \mathbf{A} \mathbf{c}$$

Graph Laplacian Matrix

$$\begin{pmatrix} \sum_{j \neq 1} a_{1j} & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & \sum_{j \neq 2} a_{2j} & \cdots & -a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & \sum_{j \neq n} a_{nj} \end{pmatrix}$$

example




The graph Laplacian is given as

normalized symmetric Laplacian matrix

The *normalized symmetric Laplacian matrix* of a graph is defined as

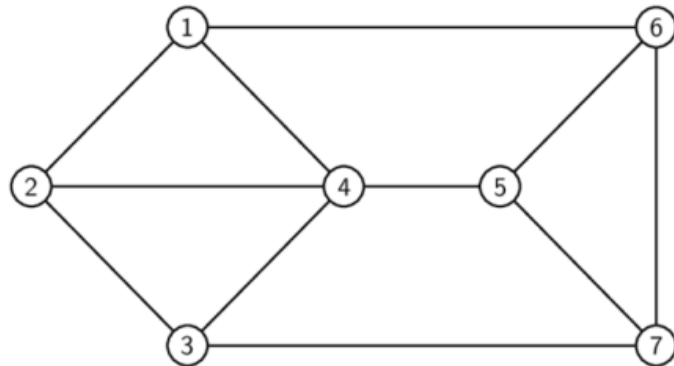
$$\mathbf{L}^s = \mathbf{\Delta}^{-1/2} \mathbf{L} \mathbf{\Delta}^{-1/2}$$

$$\mathbf{c}^T \mathbf{L}^s \mathbf{c} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \left(\frac{c_i}{\sqrt{d_i}} - \frac{c_j}{\sqrt{d_j}} \right)^2 \geq 0$$



\mathbf{L}^s is also positive semidefinite

example



CLUSTERING AS GRAPH CUTS

A *k*-way cut in a graph is a partitioning or clustering of the vertex set, given as $\mathcal{C} = \{C_1, \dots, C_k\}$. We require \mathcal{C} to optimize some objective function that captures the intuition that nodes within a cluster should have high similarity, and nodes from different clusters should have low similarity.

CLUSTERING AS GRAPH CUTS

Given a weighted graph G defined by its similarity matrix \mathbf{A} , let $S, T \subseteq V$ be any two subsets of the vertices. We denote by $W(S, T)$ the sum of the weights on all edges with one vertex in S and the other in T , given as

$$W(S, T) = \sum_{v_i \in S} \sum_{v_j \in T} a_{ij}$$

The *weight of the cut* or *cut weight* is defined as the sum of all the weights on edges between vertices in S and \bar{S} , given as $W(S, \bar{S})$.

CLUSTERING AS GRAPH CUTS

Given a clustering $\mathcal{C} = \{C_1, \dots, C_k\}$ comprising k clusters. Let $\mathbf{c}_i \in \{0, 1\}^n$ be the *cluster indicator vector* that records the cluster membership for cluster C_i , defined as

$$c_{ij} = \begin{cases} 1 & \text{if } v_j \in C_i \\ 0 & \text{if } v_j \notin C_i \end{cases}$$

The cluster size can be written as

$$|C_i| = \mathbf{c}_i^T \mathbf{c}_i = \|\mathbf{c}_i\|^2$$

$$\mathbf{c}_i^T \mathbf{c}_j = 0$$

CLUSTERING AS GRAPH CUTS

$$\text{vol}(C_i) = W(C_i, V) = \sum_{v_r \in C_i} d_r = \sum_{v_r \in C_i} c_{ir} d_r c_{ir} = \sum_{r=1}^n \sum_{s=1}^n c_{ir} \Delta_{rs} c_{is}$$

$$\text{vol}(C_i) = \mathbf{c}_i^T \Delta \mathbf{c}_i$$

CLUSTERING AS GRAPH CUTS

$$\begin{aligned} W(C_i, \overline{C_i}) &= \sum_{v_r \in C_i} \sum_{v_s \in V - C_i} a_{rs} = W(C_i, V) - W(C_i, C_i) \\ &= \mathbf{c}_i (\mathbf{\Delta} - \mathbf{A}) \mathbf{c}_i = \mathbf{c}_i^T \mathbf{L} \mathbf{c}_i \end{aligned}$$

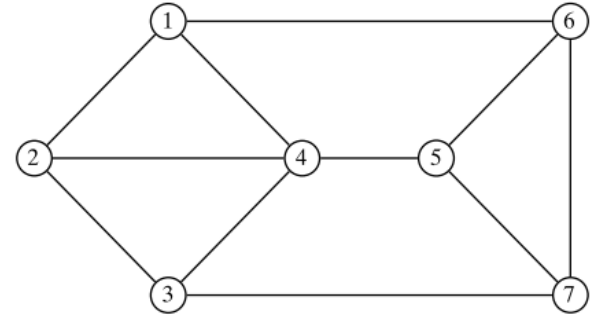
Example

$$C_1 = \{1, 2, 3, 4\}$$

$$C_2 = \{5, 6, 7\}$$

$$\mathbf{c}_1 = (1, 1, 1, 1, 0, 0, 0)^T$$

$$\mathbf{c}_2 = (0, 0, 0, 0, 1, 1, 1)^T$$



Clustering Objective Functions: Ratio Cut

The clustering objective function can be formulated as an optimization problem over the k -way cut $\mathcal{C} = \{C_1, \dots, C_k\}$.

Ratio Cut

The *ratio cut* objective is defined over a k -way cut as follows:

$$\min_{\mathcal{C}} J_{rc}(\mathcal{C}) = \sum_{i=1}^k \frac{W(C_i, \overline{C_i})}{|C_i|} = \sum_{i=1}^k \frac{\mathbf{c}_i^T \mathbf{L} \mathbf{c}_i}{\mathbf{c}_i^T \mathbf{c}_i} = \sum_{i=1}^k \frac{\mathbf{c}_i^T \mathbf{L} \mathbf{c}_i}{\|\mathbf{c}_i\|^2}$$

for binary cluster indicator vectors \mathbf{c}_i , the
ratio cut objective is
NP-hard

Clustering Objective Functions: Ratio Cut

An obvious relaxation is to allow \mathbf{c}_i to take on any real value. In this case, we can rewrite the objective as

$$\min_{\mathcal{C}} J_{rc}(\mathcal{C}) = \sum_{i=1}^k \frac{\mathbf{c}_i^T \mathbf{L} \mathbf{c}_i}{\|\mathbf{c}_i\|^2} = \sum_{i=1}^k \left(\frac{\mathbf{c}_i}{\|\mathbf{c}_i\|} \right)^T \mathbf{L} \left(\frac{\mathbf{c}_i}{\|\mathbf{c}_i\|} \right) = \sum_{i=1}^k \mathbf{u}_i^T \mathbf{L} \mathbf{u}_i$$
$$\mathbf{u}_i^T \mathbf{u}_i = 1$$

Clustering Objective Functions: Ratio Cut

we introduce the Lagrange multiplier λ_i

for each cluster C_i . We have

$$\frac{\partial}{\partial \mathbf{u}_i} \left(\sum_{i=1}^k \mathbf{u}_i^T \mathbf{L} \mathbf{u}_i + \sum_{i=1}^n \lambda_i (1 - \mathbf{u}_i^T \mathbf{u}_i) \right) = \mathbf{0}, \text{ which implies that}$$

Eigen

Clustering Objective Functions: Ratio Cut

we should choose the k smallest eigenvalues, and the corresponding eigenvectors, so that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$$

$$\min_{\mathcal{C}} J_{rc}(\mathcal{C}) = \mathbf{u}_n^T \mathbf{L} \mathbf{u}_n + \cdots + \mathbf{u}_{n-k+1}^T \mathbf{L} \mathbf{u}_{n-k+1}$$

$$= \lambda_n + \cdots + \lambda_{n-k+1}$$

$$0 = \lambda_n \leq \lambda_{n-1} \leq \lambda_{n-k+1}$$

Normalized cut

Normalized cut is similar to ratio cut, except that it divides the cut weight of each cluster by the volume of a cluster instead of its size

$$\min_{\mathcal{C}} J_{nc}(\mathcal{C}) = \sum_{i=1}^k \frac{W(C_i, \overline{C_i})}{vol(C_i)} = \sum_{i=1}^k \frac{\mathbf{c}_i^T \mathbf{L} \mathbf{c}_i}{\mathbf{c}_i^T \mathbf{\Delta} \mathbf{c}_i}$$

Spectral Clustering Algorithm

problem we face is that the eigenvectors u_i are not binary, and thus it is not immediately clear how we can assign points to clusters

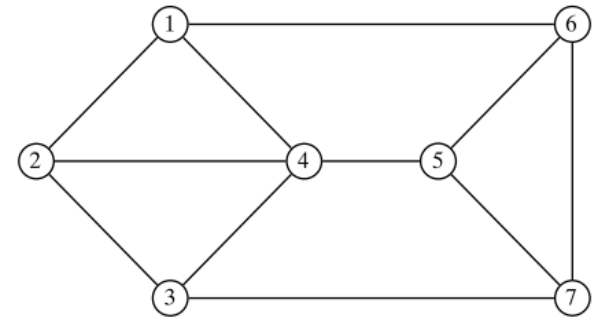
$u_n, u_{n-1}, \dots, u_{n-k+1}$ relaxed cluster indicator vector

Spectral Clustering (D, k):

- 1 Compute the similarity matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$
- 2 **if** *ratio cut* **then** $\mathbf{B} \leftarrow \mathbf{L}$
- 3 **else if** *normalized cut* **then** $\mathbf{B} \leftarrow \mathbf{L}^s$
- 4 Solve $\mathbf{B}\mathbf{u}_i = \lambda_i\mathbf{u}_i$ for $i = n, \dots, n - k + 1$, where
$$\lambda_n \leq \lambda_{n-1} \leq \dots \leq \lambda_{n-k+1}$$
- 5 $\mathbf{U} \leftarrow (\mathbf{u}_n \quad \mathbf{u}_{n-1} \quad \dots \quad \mathbf{u}_{n-k+1})$
- 6 $\mathbf{Y} \leftarrow$ normalize rows of \mathbf{U}
- 7 $\mathcal{C} \leftarrow \{C_1, \dots, C_k\}$ via K-means on \mathbf{Y}

example

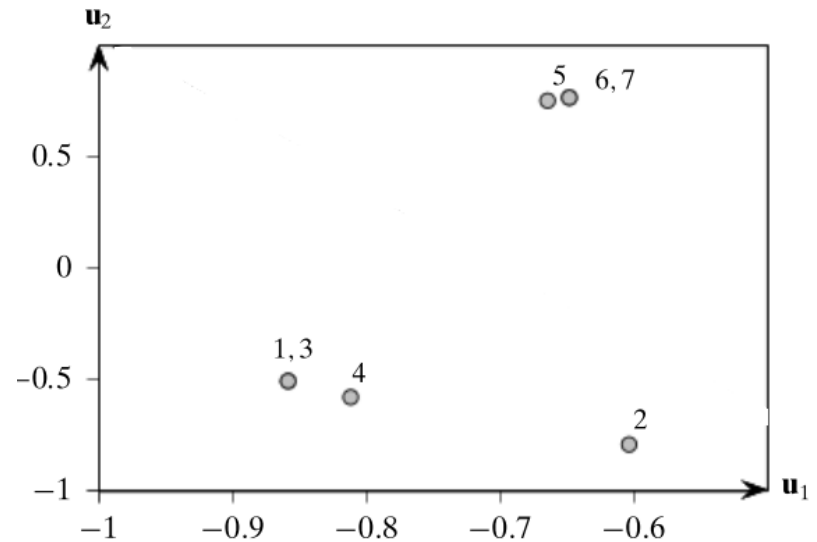
we want to find $k = 2$ clusters. For the normalized Laplacian matrix, we compute the eigenvectors, v_7 and v_6 , corresponding to the two smallest eigenvalues, $\lambda_7 = 0$ and $\lambda_6 = 0.517$.



$$\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 \\ -0.378 & -0.226 \\ -0.378 & -0.499 \\ -0.378 & -0.226 \\ -0.378 & -0.272 \\ -0.378 & 0.425 \\ -0.378 & 0.444 \\ -0.378 & 0.444 \end{pmatrix}$$

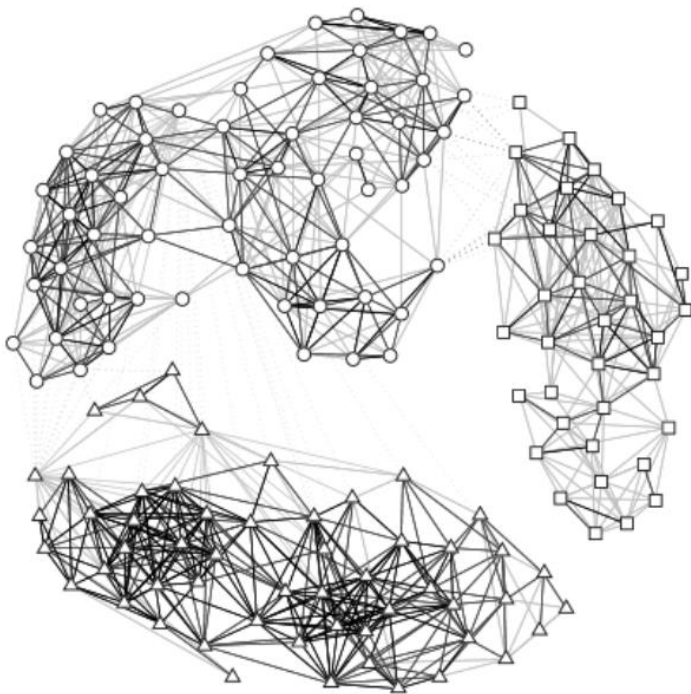
example

$$\mathbf{Y} = \begin{pmatrix} -0.859 & -0.513 \\ -0.604 & -0.797 \\ -0.859 & -0.513 \\ -0.812 & -0.584 \\ -0.664 & 0.747 \\ -0.648 & 0.761 \\ -0.648 & 0.761 \end{pmatrix}$$



example

- ❖ apply spectral clustering on the Iris graph: $k = 3$ clusters
- ❖ Normalized Cut
- ❖ Comparing them with the true Iris classes



	setosa	virginica	versicolor
C_1 (triangle)	50	0	4
C_2 (square)	0	36	0
C_3 (circle)	0	14	46