Spectral clustering

He Sun

THE UNIVERSITY of EDINBURGH

Recall: normalised graph Laplacian

Let $G = (V, E, w)$ be an undirected and weighted graph with n vertices and weight function $w : E \to \mathbb{R}_{\geq 0}$. Let $d_u = \sum_{u \sim v} w(u, v)$.

Recall: normalised graph Laplacian

Let $G = (V, E, w)$ be an undirected and weighted graph with n vertices and weight function $w : E \to \mathbb{R}_{\geq 0}$. Let $d_u = \sum_{u \sim v} w(u, v)$.

Normalised Laplacian Matrix

The normalised Laplacian matrix of G is defined by

$$\mathcal{L} \triangleq \mathbf{I} - \mathbf{D}^{-1/2} \cdot \mathbf{A}\mathbf{D}^{-1/2},$$

where $\mathbf{A}$ is the adjacency matrix of G.

Recall: normalised graph Laplacian

Let $G = (V, E, w)$ be an undirected and weighted graph with n vertices and weight function $w : E \to \mathbb{R}_{\geq 0}$. Let $d_u = \sum_{u \sim v} w(u, v)$.

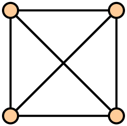
NORMALISED LAPLACIAN MATRIX

The normalised Laplacian matrix of G is defined by

$$\mathcal{L} \triangleq \mathbf{I} - \mathbf{D}^{-1/2} \cdot \mathbf{A}\mathbf{D}^{-1/2},$$

where $\mathbf{A}$ is the adjacency matrix of G.

Example:



$$\mathcal{L}_G = \begin{pmatrix} 1 & -1/3 & -1/3 & -1/3 \\ -1/3 & 1 & -1/3 & -1/3 \\ -1/3 & -1/3 & 1 & -1/3 \\ -1/3 & -1/3 & -1/3 & 1 \end{pmatrix}$$

Recall: normalised graph Laplacian

Let $G = (V, E, w)$ be an undirected and weighted graph with n vertices and weight function $w : E \to \mathbb{R}_{\geq 0}$. Let $d_u = \sum_{u \sim v} w(u, v)$.

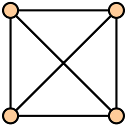
NORMALISED LAPLACIAN MATRIX

The normalised Laplacian matrix of G is defined by

$$\mathcal{L} \triangleq \mathbf{I} - \mathbf{D}^{-1/2} \cdot \mathbf{A}\mathbf{D}^{-1/2},$$

where $\mathbf{A}$ is the adjacency matrix of G.

Example:



$$\mathcal{L}_G = \begin{pmatrix} 1 & -1/3 & -1/3 & -1/3 \\ -1/3 & 1 & -1/3 & -1/3 \\ -1/3 & -1/3 & 1 & -1/3 \\ -1/3 & -1/3 & -1/3 & 1 \end{pmatrix}$$

Matrix $\mathcal{L}$ has eigenvalues $0 = \lambda_1 \leq \ldots \leq \lambda_n$ with corresponding eigenvectors

$$f_1, \ldots, f_n.$$

Recall: graph conductance

The conductance of a set S is defined by

$$\phi_G(S) \triangleq \frac{w(S, V \setminus S)}{\mathrm{vol}(S)},$$

where

$$w(S, V \setminus S) \triangleq \sum_{u \in S, v \in V \setminus S, u \sim v} w(u, v)$$

and $\mathrm{vol}(S) \triangleq \sum_{u \in S} d_u$.

Recall: graph conductance

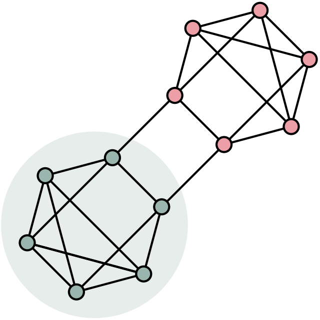
The conductance of a set S is defined by

$$\phi_G(S) \triangleq \frac{w(S, V \setminus S)}{\mathrm{vol}(S)},$$

where

$$w(S, V \setminus S) \triangleq \sum_{u \in S, v \in V \setminus S, u \sim v} w(u, v)$$

and $\mathrm{vol}(S) \triangleq \sum_{u \in S} d_u$.



$$\phi_G(S) = \frac{2}{4 \cdot 6} = \frac{1}{12}$$

Recall: graph conductance

The conductance of a set S is defined by

$$\phi_G(S) \triangleq \frac{w(S, V \setminus S)}{\mathrm{vol}(S)},$$

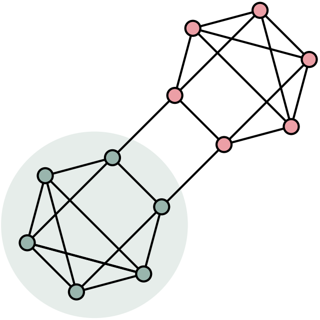
where

$$w(S, V \setminus S) \triangleq \sum_{u \in S, v \in V \setminus S, u \sim v} w(u, v)$$

and $\mathrm{vol}(S) \triangleq \sum_{u \in S} d_u$.

The conductance of a graph G is defined by

$$\phi_G \triangleq \min_{S : \mathrm{vol}(S) \leq \mathrm{vol}(V)/2} \phi_G(S).$$



$$\phi_G(S) = \frac{2}{4 \cdot 6} = \frac{1}{12}$$

Recall: graph conductance

The conductance of a set S is defined by

$$\phi_G(S) \triangleq \frac{w(S, V \setminus S)}{\mathrm{vol}(S)},$$

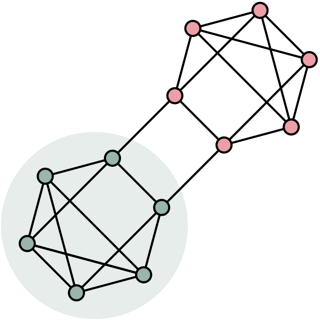
where

$$w(S, V \setminus S) \triangleq \sum_{u \in S, v \in V \setminus S, u \sim v} w(u, v)$$

and $\mathrm{vol}(S) \triangleq \sum_{u \in S} d_u$.

The conductance of a graph G is defined by

$$\phi_G \triangleq \min_{S : \mathrm{vol}(S) \leq \mathrm{vol}(V)/2} \phi_G(S).$$



$$\phi_G(S) = \frac{2}{4 \cdot 6} = \frac{1}{12}$$

Cheeger's Inequality

$$\frac{\lambda_2}{2} \leq \phi_G \leq \sqrt{2\lambda_2}.$$

Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$

Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$

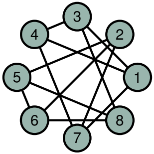
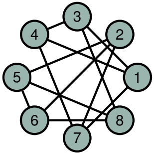


Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$

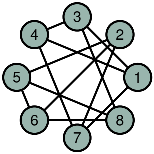


$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^\intercal$$

Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^\intercal$$

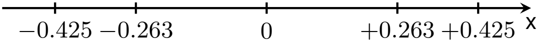
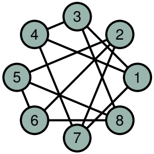


Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0&0&1&1&0&0&1&0\\ 0&0&0&0&1&1&1&0\\ 1&0&0&1&0&0&0&1\\ 1&0&1&0&0&0&1&0\\ 0&1&0&0&0&1&0&1\\ 0&1&0&0&1&0&0&1\\ 1&1&0&1&0&0&0&0\\ 0&0&1&0&1&1&0&0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1&0&-\frac{1}{3}&-\frac{1}{3}&0&0&-\frac{1}{3}&0\\ 0&1&0&0&-\frac{1}{3}&-\frac{1}{3}&-\frac{1}{3}&0\\ -\frac{1}{3}&0&1&-\frac{1}{3}&0&0&0&-\frac{1}{3}\\ -\frac{1}{3}&0&-\frac{1}{3}&1&0&0&-\frac{1}{3}&0\\ 0&-\frac{1}{3}&0&0&1&-\frac{1}{3}&0&-\frac{1}{3}\\ 0&-\frac{1}{3}&0&0&-\frac{1}{3}&1&0&-\frac{1}{3}\\ -\frac{1}{3}&-\frac{1}{3}&0&-\frac{1}{3}&0&0&1&0\\ 0&0&-\frac{1}{3}&0&-\frac{1}{3}&-\frac{1}{3}&0&1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^\intercal$$

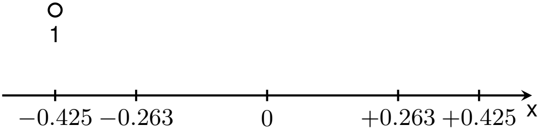
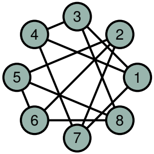


Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\intercal}$$

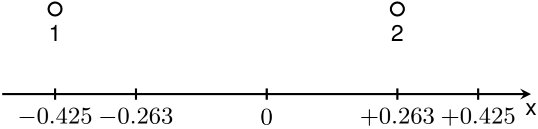
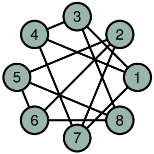


Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\intercal}$$

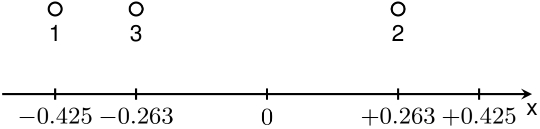
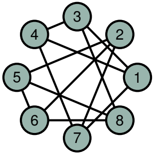


Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\intercal}$$

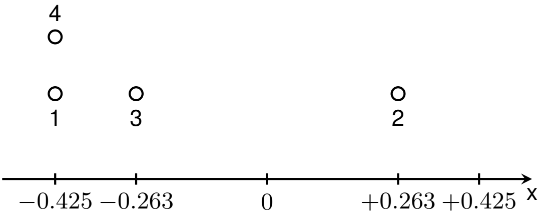
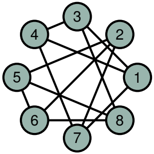


Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\intercal}$$

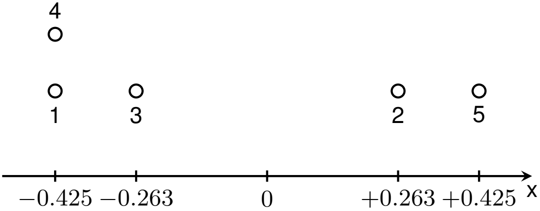
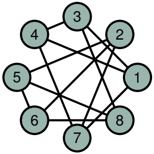


Illustration of the graph partitioning algorithm

$$\mathbf{A}=\begin{pmatrix}0&0&1&1&0&0&1&0\\0&0&0&0&1&1&1&0\\1&0&0&1&0&0&0&1\\1&0&1&0&0&0&1&0\\0&1&0&0&0&1&0&1\\0&1&0&0&1&0&0&1\\1&1&0&1&0&0&0&0\\0&0&1&0&1&1&0&0\end{pmatrix}\qquad \mathcal{L}=\begin{pmatrix}1&0&-\frac{1}{3}&-\frac{1}{3}&0&0&-\frac{1}{3}&0\\0&1&0&0&-\frac{1}{3}&-\frac{1}{3}&-\frac{1}{3}&0\\-\frac{1}{3}&0&1&-\frac{1}{3}&0&0&0&-\frac{1}{3}\\-\frac{1}{3}&0&-\frac{1}{3}&1&0&0&-\frac{1}{3}&0\\0&-\frac{1}{3}&0&0&1&-\frac{1}{3}&0&-\frac{1}{3}\\0&-\frac{1}{3}&0&0&-\frac{1}{3}&1&0&-\frac{1}{3}\\-\frac{1}{3}&-\frac{1}{3}&0&-\frac{1}{3}&0&0&1&0\\0&0&-\frac{1}{3}&0&-\frac{1}{3}&-\frac{1}{3}&0&1\end{pmatrix}$$



$$\lambda_2=\sqrt{5}/3\approx 0.75$$

$$v=(-0.425,+0.263,-0.263,-0.425,+0.425,+0.425,-0.263,+0.263)^{\mathsf{T}}$$

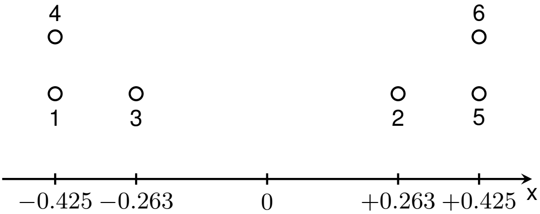
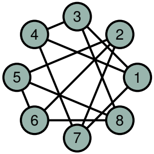


Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\mathsf{T}}$$

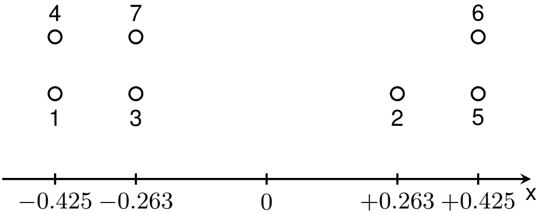
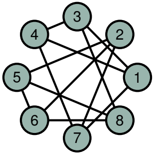


Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\mathsf{T}}$$

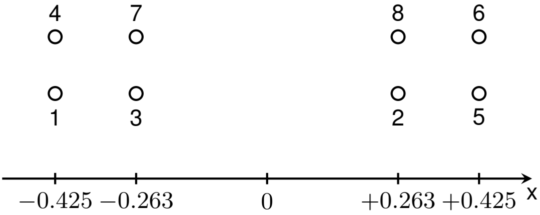
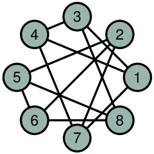


Illustration of the graph partitioning algorithm

$$\mathbf{A}=\begin{pmatrix}0&0&1&1&0&0&1&0\\0&0&0&0&1&1&1&0\\1&0&0&1&0&0&0&1\\1&0&1&0&0&0&1&0\\0&1&0&0&0&1&0&1\\0&1&0&0&1&0&0&1\\1&1&0&1&0&0&0&0\\0&0&1&0&1&1&0&0\end{pmatrix}\qquad \mathcal{L}=\begin{pmatrix}1&0&-\frac{1}{3}&-\frac{1}{3}&0&0&-\frac{1}{3}&0\\0&1&0&0&-\frac{1}{3}&-\frac{1}{3}&-\frac{1}{3}&0\\-\frac{1}{3}&0&1&-\frac{1}{3}&0&0&0&-\frac{1}{3}\\-\frac{1}{3}&0&-\frac{1}{3}&1&0&0&-\frac{1}{3}&0\\0&-\frac{1}{3}&0&0&1&-\frac{1}{3}&0&-\frac{1}{3}\\0&-\frac{1}{3}&0&0&-\frac{1}{3}&1&0&-\frac{1}{3}\\-\frac{1}{3}&-\frac{1}{3}&0&-\frac{1}{3}&0&0&1&0\\0&0&-\frac{1}{3}&0&-\frac{1}{3}&-\frac{1}{3}&0&1\end{pmatrix}$$



$$\lambda_2=\sqrt{5}/3\approx 0.75$$

$$v=(-0.425,+0.263,-0.263,-0.425,+0.425,+0.425,-0.263,+0.263)^{\mathsf{T}}$$

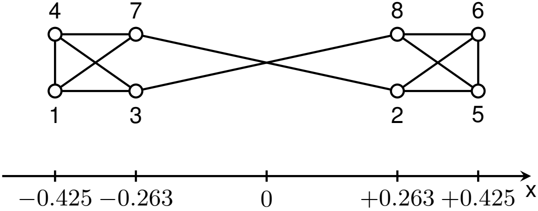
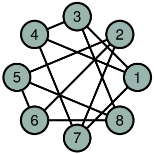


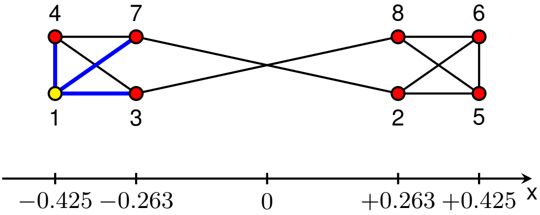
Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0&0&1&1&0&0&1&0\\ 0&0&0&0&1&1&1&0\\ 1&0&0&1&0&0&0&1\\ 1&0&1&0&0&0&1&0\\ 0&1&0&0&0&1&0&1\\ 0&1&0&0&1&0&0&1\\ 1&1&0&1&0&0&0&0\\ 0&0&1&0&1&1&0&0 \end{pmatrix} \quad \mathcal{L} = \begin{pmatrix} 1&0&-\frac{1}{3}&-\frac{1}{3}&0&0&-\frac{1}{3}&0\\ 0&1&0&0&-\frac{1}{3}&-\frac{1}{3}&-\frac{1}{3}&0\\ -\frac{1}{3}&0&1&-\frac{1}{3}&0&0&0&-\frac{1}{3}\\ -\frac{1}{3}&0&-\frac{1}{3}&1&0&0&-\frac{1}{3}&0\\ 0&-\frac{1}{3}&0&0&1&-\frac{1}{3}&0&-\frac{1}{3}\\ 0&-\frac{1}{3}&0&0&-\frac{1}{3}&1&0&-\frac{1}{3}\\ -\frac{1}{3}&-\frac{1}{3}&0&-\frac{1}{3}&0&0&1&0\\ 0&0&-\frac{1}{3}&0&-\frac{1}{3}&-\frac{1}{3}&0&1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\mathsf{T}}$$

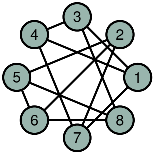


Sweep: 1

Conductance: 1

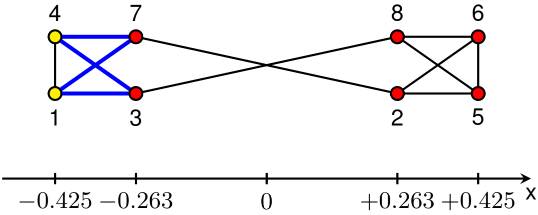
Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0&0&1&1&0&0&1&0\\ 0&0&0&0&1&1&1&0\\ 1&0&0&1&0&0&0&1\\ 1&0&1&0&0&0&1&0\\ 0&1&0&0&0&1&0&1\\ 0&1&0&0&1&0&0&1\\ 1&1&0&1&0&0&0&0\\ 0&0&1&0&1&1&0&0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1&0&-\frac{1}{3}&-\frac{1}{3}&0&0&-\frac{1}{3}&0\\ 0&1&0&0&-\frac{1}{3}&-\frac{1}{3}&-\frac{1}{3}&0\\ -\frac{1}{3}&0&1&-\frac{1}{3}&0&0&0&-\frac{1}{3}\\ -\frac{1}{3}&0&-\frac{1}{3}&1&0&0&-\frac{1}{3}&0\\ 0&-\frac{1}{3}&0&0&1&-\frac{1}{3}&0&-\frac{1}{3}\\ 0&-\frac{1}{3}&0&0&-\frac{1}{3}&1&0&-\frac{1}{3}\\ -\frac{1}{3}&-\frac{1}{3}&0&-\frac{1}{3}&0&0&1&0\\ 0&0&-\frac{1}{3}&0&-\frac{1}{3}&-\frac{1}{3}&0&1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\mathsf{T}}$$

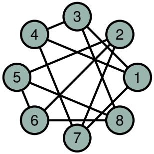


Sweep: 2

Conductance: 0.666

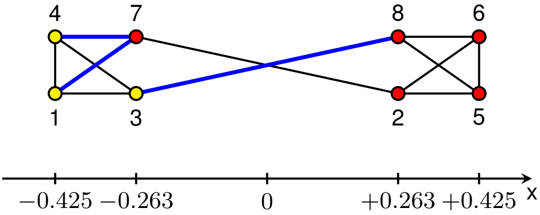
Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0&0&1&1&0&0&1&0\\ 0&0&0&0&1&1&1&0\\ 1&0&0&1&0&0&0&1\\ 1&0&1&0&0&0&1&0\\ 0&1&0&0&0&1&0&1\\ 0&1&0&0&1&0&0&1\\ 1&1&0&1&0&0&0&0\\ 0&0&1&0&1&1&0&0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1&0&-\frac13&-\frac13&0&0&-\frac13&0\\ 0&1&0&0&-\frac13&-\frac13&-\frac13&0\\ -\frac13&0&1&-\frac13&0&0&0&-\frac13\\ -\frac13&0&-\frac13&1&0&0&-\frac13&0\\ 0&-\frac13&0&0&1&-\frac13&0&-\frac13\\ 0&-\frac13&0&0&-\frac13&1&0&-\frac13\\ -\frac13&-\frac13&0&-\frac13&0&0&1&0\\ 0&0&-\frac13&0&-\frac13&-\frac13&0&1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\intercal}$$

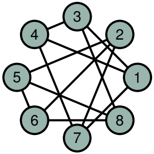


Sweep: 3

Conductance: 0.333

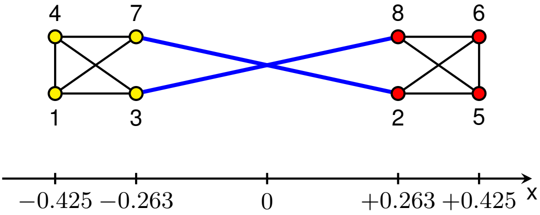
Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\intercal}$$

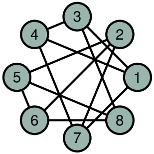


Sweep: 4

Conductance: 0.166

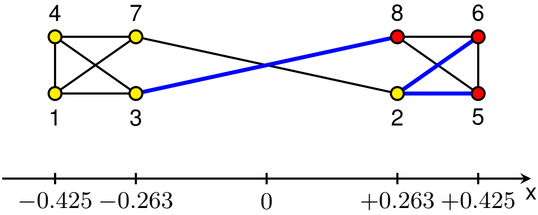
Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\intercal}$$

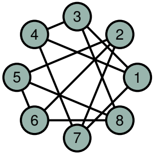


Sweep: 5

Conductance: 0.333

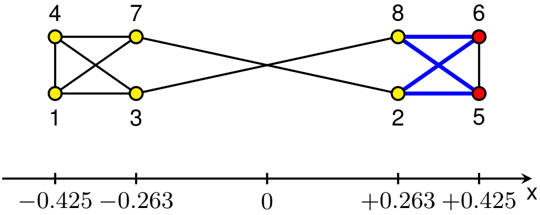
Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \quad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^{\mathsf{T}}$$

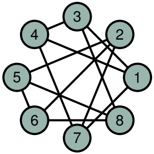


Sweep: 6

Conductance: 0.666

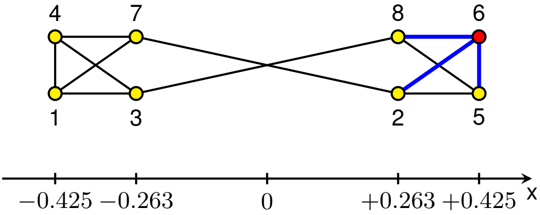
Illustration of the graph partitioning algorithm

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \qquad \mathcal{L} = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & 1 & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & 1 & -\frac{1}{3} & 0 & 0 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & 0 & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{1}{3} & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & 1 & 0 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 & 1 \end{pmatrix}$$



$$\lambda_2 = \sqrt{5}/3 \approx 0.75$$

$$v = (-0.425, +0.263, -0.263, -0.425, +0.425, +0.425, -0.263, +0.263)^\intercal$$



Sweep: 7

Conductance: 1

Illustration of the graph partition on a large graph

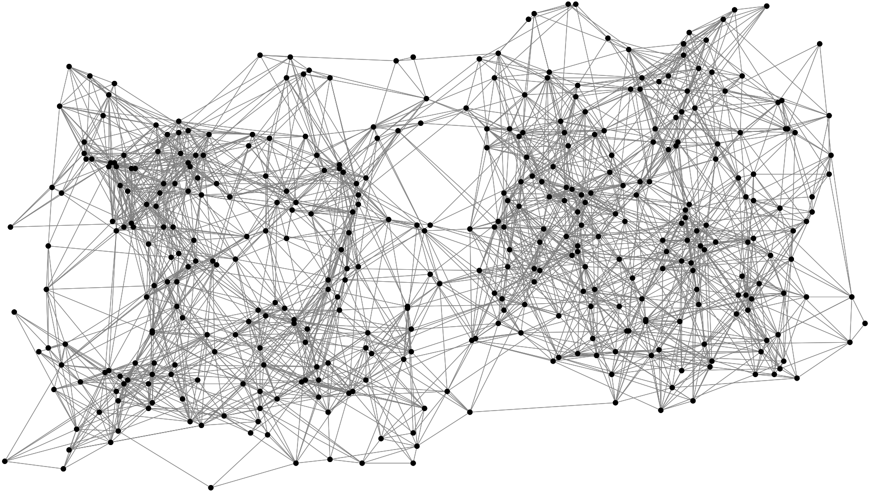


Illustration continues: the second eigenvector

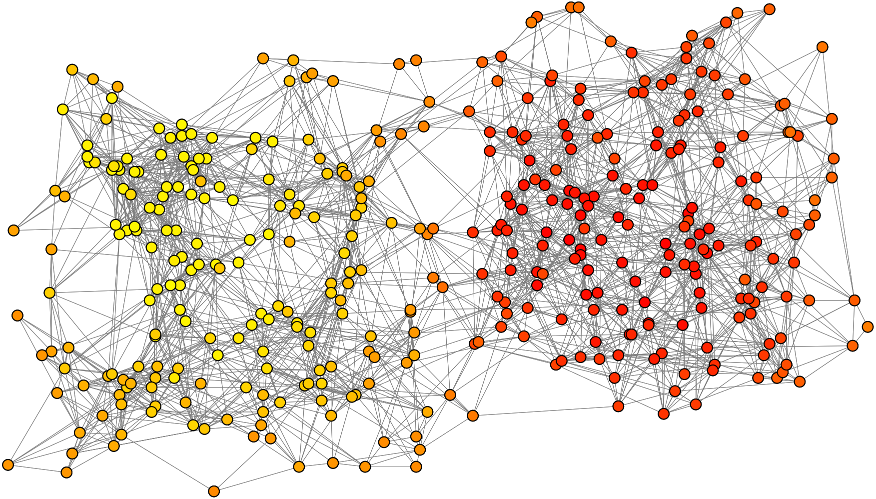
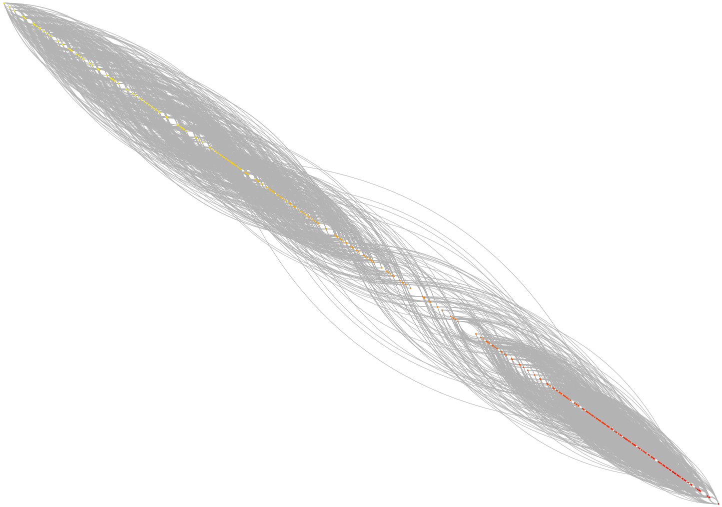
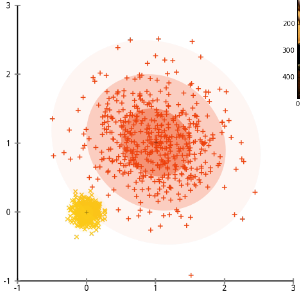


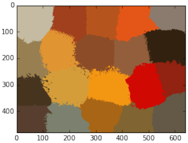
Illustration continues: vertices and edges after the embedding

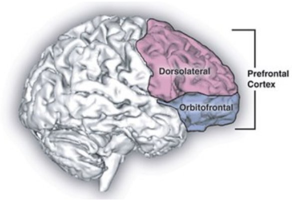


From graph partitioning to graph clustering

Clustering is the task of dividing objects in groups (clusters) so that similar objects are grouped together and dissimilar objects are separated in different groups.



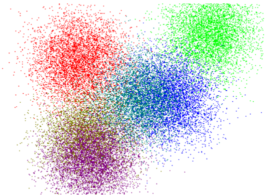




Numerous applications in image segmentation, community detection, bioinformatics, network analysis, among many others

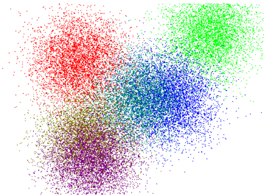
On the hardness of clustering

- One basic learning task in machine learning
  - For many applications training sets are unavailable
- The problem is inherently difficult to be formalised
  - There's no "ground truth"
  - A cluster structure can be defined in many different ways
  - "Impossibility theorem for clustering" (Kleinberg, NIPS '13)
- Most formalisations are actually NP-hard



On the hardness of clustering

- One basic learning task in machine learning
  - For many applications training sets are unavailable
- The problem is inherently difficult to be formalised
  - There's no "ground truth"
  - A cluster structure can be defined in many different ways
  - "Impossibility theorem for clustering" (Kleinberg, NIPS '13)
- Most formalisations are actually NP-hard

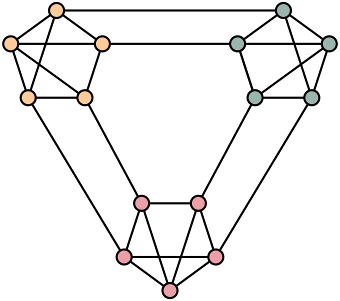


However,

- We have algorithms that "work in practice"
- The more well-clustered the data, the better the quality of the clustering produced

Graph clustering

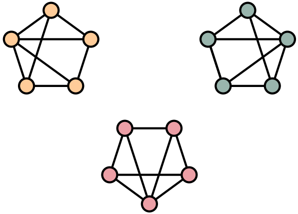
Partition the graph into clusters so that vertices in the same cluster have, on average, more connections among each other than with vertices in other clusters.



Graph clustering via eigenvectors

Lemma (Problem 1 of Homework 5)

Graph G has exactly k connected components iff $\lambda_k = 0$ and $\lambda_{k+1} > 0$.

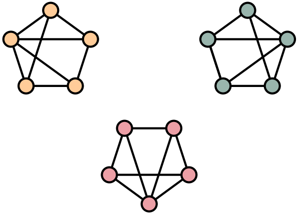


Graph clustering via eigenvectors

Lemma (Problem 1 of Homework 5)

Graph G has exactly k connected components iff $\lambda_k = 0$ and $\lambda_{k+1} > 0$.

Let $S_1, \ldots, S_k$ be G's connected components.



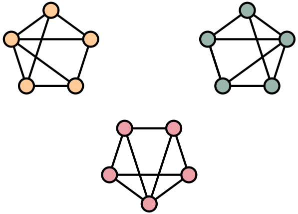
Graph clustering via eigenvectors

Lemma (Problem 1 of Homework 5)

Graph G has exactly k connected components iff $\lambda_k = 0$ and $\lambda_{k+1} > 0$.

Let $S_1, \ldots, S_k$ be G's connected components.
For any $1 \leq i \leq k$ let

$$\chi_i(v) = \begin{cases} \sqrt{d_v} & \text{if } v \in S_i, \\ 0 & \text{otherwise.} \end{cases}$$



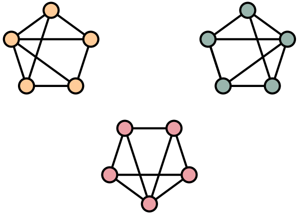
Graph clustering via eigenvectors

Lemma (Problem 1 of Homework 5)

Graph G has exactly k connected components iff $\lambda_k = 0$ and $\lambda_{k+1} > 0$.

Let $S_1, \ldots, S_k$ be G's connected components.
For any $1 \leq i \leq k$ let

$$\chi_i(v) = \begin{cases} \sqrt{d_v} & \text{if } v \in S_i, \\ 0 & \text{otherwise.} \end{cases}$$



It holds that $\{f_1, \ldots, f_k\} = \{\chi_1, \ldots, \chi_k\}$, i.e., the k eigenvectors can be applied to find k connected components.

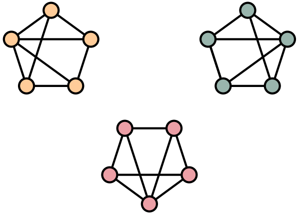
Graph clustering via eigenvectors

LEMMA (PROBLEM 1 OF HOMEWORK 5)

Graph G has exactly k connected components iff $\lambda_k = 0$ and $\lambda_{k+1} > 0$.

Let $S_1, \ldots, S_k$ be G's connected components.
For any $1 \leq i \leq k$ let

$$\chi_i(v) = \begin{cases} \sqrt{d_v} & \text{if } v \in S_i, \\ 0 & \text{otherwise.} \end{cases}$$



It holds that $\{f_1, \ldots, f_k\} = \{\chi_1, \ldots, \chi_k\}$, i.e., the k eigenvectors can be applied to find k connected components.

This situation can be informally viewed as, if $\lambda_k / \lambda_{k+1} = 0$, the structure of k clusters is completed encoded in the bottom k eigenvectors.

Graph clustering via eigenvectors (cont.)

Lemma (Problem 1 of Homework 5)

Graph G has exactly k connected components iff $\lambda_k = 0$ and $\lambda_{k+1} > 0$.

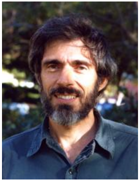
Graph clustering via eigenvectors (cont.)

Lemma (Problem 1 of Homework 5)

Graph G has exactly k connected components iff $\lambda_k = 0$ and $\lambda_{k+1} > 0$.

Davis-Kahan Theorem (1970, very informal statement in our setting)

As long as not too many edges between different clusters are added, the k eigenvectors do not change too much.



Chandler Davis

(mathematician, writer, politician)

William Kahan

(mathematician, received Turing Award in '89)

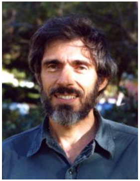
Graph clustering via eigenvectors (cont.)

Lemma (Problem 1 of Homework 5)

Graph G has exactly k connected components iff $\lambda_k = 0$ and $\lambda_{k+1} > 0$.

Davis-Kahan Theorem (1970, very informal statement in our setting)

As long as <u>not too many edges between different clusters are added</u>, the k eigenvectors do not change too much.



Chandler Davis

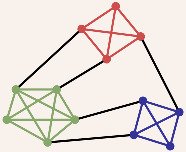
(mathematician, writer, politician)

William Kahan

(mathematician, received Turing Award in '89)

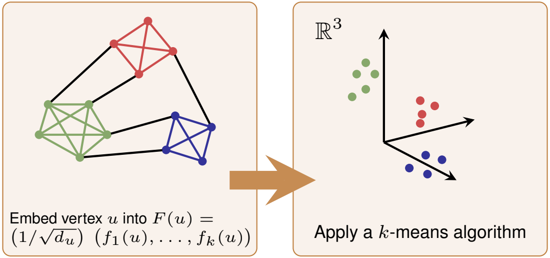
As long as λ_k/λ_{k+1} is small, the k eigenvectors can be used to find k clusters.

Spectral clustering



Embed vertex u into $F(u) = \left(1/\sqrt{d_u}\right) \left(f_1(u), \ldots, f_k(u)\right)$

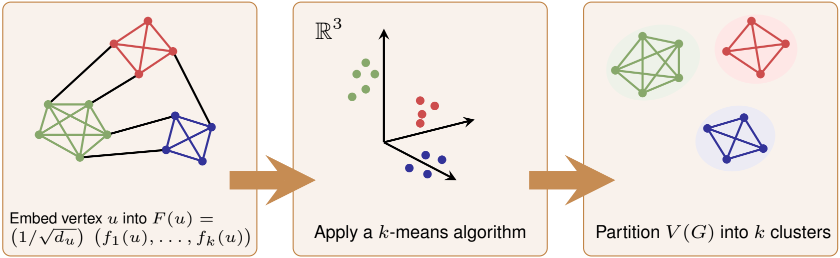
Spectral clustering



k-means clustering

- Input: Set of n points $x_1, \ldots, x_n$, where $x_i \in \mathbb{R}^d$, and parameter k.
- Goal: Assign the n points to k clusters such that total distance $\sum_{i=1}^{k} \sum_{x \in S_i} \|x - c_i\|^2$ is minimised, where c_i is the centre of cluster S_i.

Spectral clustering



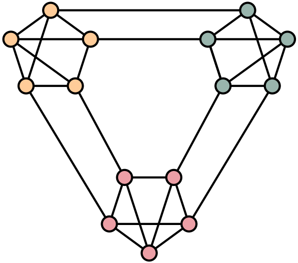
k-means clustering

- Input: Set of n points $x_1, \ldots, x_n$, where $x_i \in \mathbb{R}^d$, and parameter k.
- Goal: Assign the n points to k clusters such that total distance $\sum_{i=1}^{k} \sum_{x \in S_i} \|x - c_i\|^2$ is minimised, where c_i is the centre of cluster S_i.

Practical consideration: choosing the right value of k

The k-way expansion constant is defined by

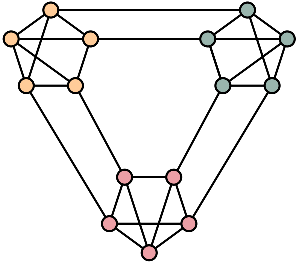
$$\rho(k) = \min_{\text{partition } A_1,\ldots,A_k} \max_{1\leq i\leq k} \phi_G(A_i).$$



Practical consideration: choosing the right value of k

The k-way expansion constant is defined by

$$\rho(k) = \min_{\text{partition } A_1,\ldots,A_k} \max_{1\le i\le k} \phi_G(A_i).$$



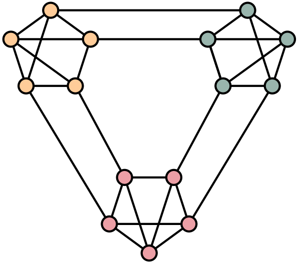
Higher-Order Cheeger's Inequality

$$\frac{\lambda_k}{2} \le \rho(k) \le O(k^3)\sqrt{\lambda_k}.$$

Practical consideration: choosing the right value of k

The k-way expansion constant is defined by

$$\rho(k) = \min_{\text{partition } A_1,\ldots,A_k} \max_{1\le i\le k} \phi_G(A_i).$$



Higher-Order Cheeger's Inequality

$$\frac{\lambda_k}{2} \le \rho(k) \le O(k^3)\sqrt{\lambda_k}.$$

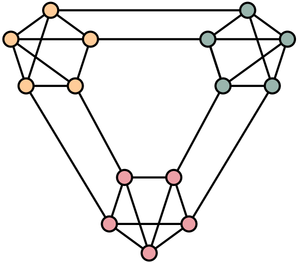
A large gap between λ_{k+1} and λ_k implies that

- existence of k clusters, each of which has low conductance $\le \rho(k)$.

Practical consideration: choosing the right value of k

The k-way expansion constant is defined by

$$\rho(k) = \min_{\text{partition } A_1,\ldots,A_k} \max_{1\le i\le k} \phi_G(A_i).$$



Higher-Order Cheeger's Inequality

$$\frac{\lambda_k}{2} \le \rho(k) \le O(k^3)\sqrt{\lambda_k}.$$

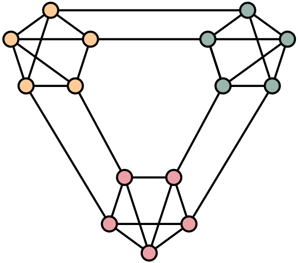
A large gap between λ_{k+1} and λ_k implies that

- existence of k clusters, each of which has low conductance $\le \rho(k)$.
- any $(k+1)$-way partition contains a set with conductance at least $\lambda_{k+1}/2$.

Practical consideration: choosing the right value of k

The k-way expansion constant is defined by

$$\rho(k) = \min_{\text{partition } A_1,\ldots,A_k} \max_{1\le i\le k} \phi_G(A_i).$$



Higher-Order Cheeger's Inequality

$$\frac{\lambda_k}{2} \le \rho(k) \le O(k^3)\sqrt{\lambda_k}.$$

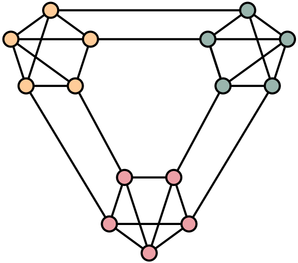
A large gap between λ_{k+1} and λ_k implies that

- existence of k clusters, each of which has low conductance $\le \rho(k)$.
- any $(k+1)$-way partition contains a set with conductance at least $\lambda_{k+1}/2$.
- Graph G has exactly k clusters.

Practical consideration: choosing the right value of k

The k-way expansion constant is defined by

$$\rho(k) = \min_{\text{partition } A_1,\ldots,A_k} \max_{1\le i\le k} \phi_G(A_i).$$



Higher-Order Cheeger's Inequality

$$\frac{\lambda_k}{2} \le \rho(k) \le O(k^3)\sqrt{\lambda_k}.$$

A large gap between λ_{k+1} and λ_k implies that

- existence of k clusters, each of which has low conductance $\le \rho(k)$.
- any $(k+1)$-way partition contains a set with conductance at least $\lambda_{k+1}/2$.
- Graph G has exactly k clusters.

Example: Assume the eigenvalues are $0,\ 0.20, 0.22,\ 0.5,\ 0.55, \ldots$, then $k = 3$.

Practical consideration: building the graph from dataset

Construction of a similarity graph

Given the set X of points $x_1, \ldots, x_n$, where $x_i \in \mathbb{R}^d$, the similarity graph $G = (V, E, w)$ of X is constructed as follows:

Practical consideration: building the graph from dataset

CONSTRUCTION OF A SIMILARITY GRAPH

Given the set X of points $x_1, \ldots, x_n$, where $x_i \in \mathbb{R}^d$, the similarity graph $G = (V, E, w)$ of X is constructed as follows:

- Every point $x_i \in X$ corresponds to a vertex called $v_i \in V$.

Practical consideration: building the graph from dataset

Construction of a similarity graph

Given the set X of points $x_1, \ldots, x_n$, where $x_i \in \mathbb{R}^d$, the similarity graph $G = (V, E, w)$ of X is constructed as follows:

- Every point $x_i \in X$ corresponds to a vertex called $v_i \in V$.
- Any pair of vertices v_i and v_j are connected with the weight

$$w(v_i, v_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right).$$

Practical consideration: building the graph from dataset

Construction of a similarity graph

Given the set X of points $x_1, \ldots, x_n$, where $x_i \in \mathbb{R}^d$, the similarity graph $G = (V, E, w)$ of X is constructed as follows:

- Every point $x_i \in X$ corresponds to a vertex called $v_i \in V$.
- Any pair of vertices v_i and v_j are connected with the weight

$$w(v_i, v_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right).$$

- The choice of σ depends on applications. Usually $\sigma \in [0.05, 10]$.

Practical consideration: building the graph from dataset

Construction of a similarity graph

Given the set X of points $x_1, \ldots, x_n$, where $x_i \in \mathbb{R}^d$, the similarity graph $G = (V, E, w)$ of X is constructed as follows:

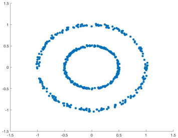
- Every point $x_i \in X$ corresponds to a vertex called $v_i \in V$.
- Any pair of vertices v_i and v_j are connected with the weight

$$w(v_i, v_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right).$$

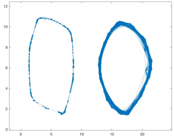
- The choice of σ depends on applications. Usually $\sigma \in [0.05, 10]$.

Example: Set $\sigma = 0.1$, and only edges with weight ≥ 0.01 shown.

original graph



similarity graph



Application of spectral clustering in image segmentation

- Every pixel p is characterised by its position in the image and its RGB value, hence every pixel p corresponds to $x_p = (x, y, r, g, b)$.
- Construct a similarity graph.
- Apply spectral clustering.

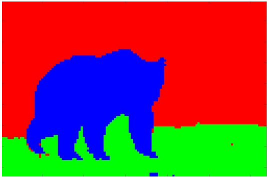
Original image

Application of spectral clustering in image segmentation

- Every pixel p is characterised by its position in the image and its RGB value, hence every pixel p corresponds to $x_p = (x, y, r, g, b)$.
- Construct a similarity graph.
- Apply spectral clustering.

Original image

Output ($\sigma = 10$)



Concluding remarks

- Spectral clustering is one of the most commonly used clustering algorithms in data science.

Concluding remarks

- Spectral clustering is one of the most commonly used clustering algorithms in data science.
  - It's easy to describe, and easy to implement. Many spectral clustering packages are available.

Concluding remarks

- Spectral clustering is one of the most commonly used clustering algorithms in data science.
  - It's easy to describe, and easy to implement. Many spectral clustering packages are available.
- Spectral clustering was first proposed in the literature around early 90s, and has been extensively studied since then.

Concluding remarks

- Spectral clustering is one of the most commonly used clustering algorithms in data science.
  - It's easy to describe, and easy to implement. Many spectral clustering packages are available.
- Spectral clustering was first proposed in the literature around early 90s, and has been extensively studied since then.
  - Why spectral clustering works in practice can be informally explained by the Davis-Kahan Theorem, one celebrated result in matrix perturbation theory developed in 1970.

Concluding remarks

- Spectral clustering is one of the most commonly used clustering algorithms in data science.
  - It's easy to describe, and easy to implement. Many spectral clustering packages are available.
- Spectral clustering was first proposed in the literature around early 90s, and has been extensively studied since then.
  - Why spectral clustering works in practice can be informally explained by the Davis-Kahan Theorem, one celebrated result in matrix perturbation theory developed in 1970.
  - A theoretical and quantitative analysis on the performance of spectral clustering is shown very recently (Peng, Sun, Zanetti, SICOMP 2017).

Concluding remarks

- Spectral clustering is one of the most commonly used clustering algorithms in data science.
  - It's easy to describe, and easy to implement. Many spectral clustering packages are available.
- Spectral clustering was first proposed in the literature around early 90s, and has been extensively studied since then.
  - Why spectral clustering works in practice can be informally explained by the Davis-Kahan Theorem, one celebrated result in matrix perturbation theory developed in 1970.
  - A theoretical and quantitative analysis on the performance of spectral clustering is shown very recently (Peng, Sun, Zanetti, SICOMP 2017).
- Several interesting questions remain open:

Concluding remarks

- Spectral clustering is one of the most commonly used clustering algorithms in data science.
  - It's easy to describe, and easy to implement. Many spectral clustering packages are available.
- Spectral clustering was first proposed in the literature around early 90s, and has been extensively studied since then.
  - Why spectral clustering works in practice can be informally explained by the Davis-Kahan Theorem, one celebrated result in matrix perturbation theory developed in 1970.
  - A theoretical and quantitative analysis on the performance of spectral clustering is shown very recently (Peng, Sun, Zanetti, SICOMP 2017).
- Several interesting questions remain open:
  - A better way to construct a similarity graph?

Concluding remarks

- Spectral clustering is one of the most commonly used clustering algorithms in data science.
  - It's easy to describe, and easy to implement. Many spectral clustering packages are available.
- Spectral clustering was first proposed in the literature around early 90s, and has been extensively studied since then.
  - Why spectral clustering works in practice can be informally explained by the Davis-Kahan Theorem, one celebrated result in matrix perturbation theory developed in 1970.
  - A theoretical and quantitative analysis on the performance of spectral clustering is shown very recently (Peng, Sun, Zanetti, SICOMP 2017).
- Several interesting questions remain open:
  - A better way to construct a similarity graph?
  - An easier way to determine the number of clusters?