Spectral Clustering

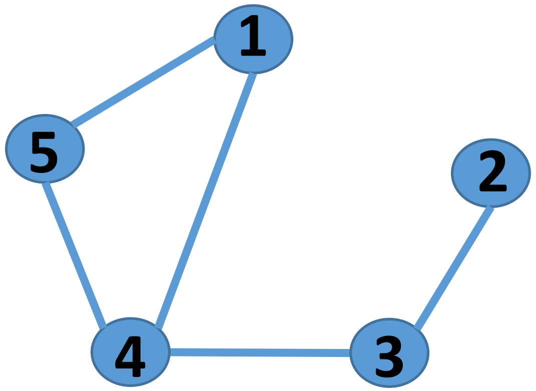
Veronika Strnadová-Neeley

Seminar on Top Algorithms in Computational Science

Main Reference: Ulrike Von Luxburg's *A Tutorial on Spectral Clustering*

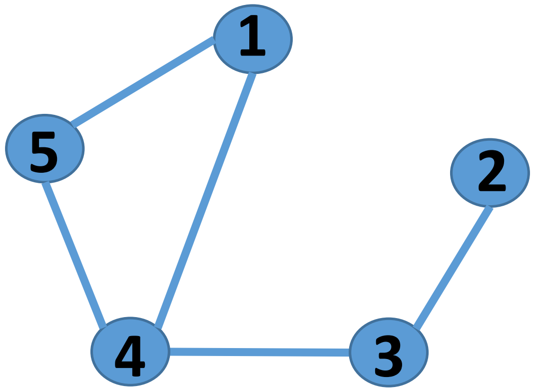
The Spectral Clustering Algorithm

Uses the eigenvalues and vectors of the *graph Laplacian matrix* in order to find clusters (or “partitions”) of the graph



The Spectral Clustering Algorithm

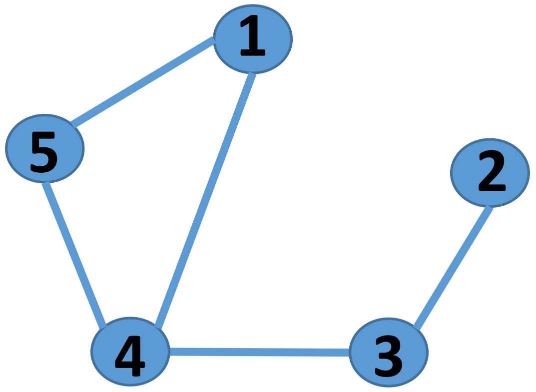
Uses the eigenvalues and vectors of the *graph Laplacian matrix* in order to find clusters (or "partitions") of the graph



$$\begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$

The Spectral Clustering Algorithm

Uses the eigenvalues and vectors of the *graph Laplacian matrix* in order to find clusters (or "partitions") of the graph



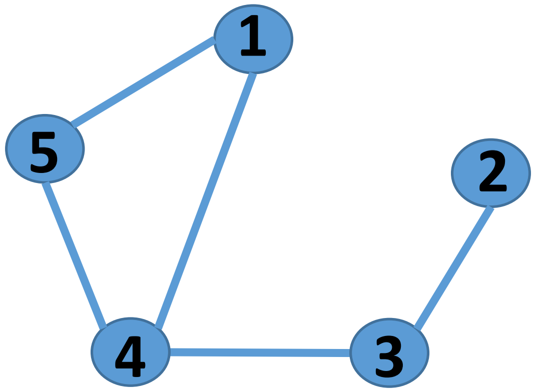
$$\begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$

$$\lambda_1, \lambda_2, \ldots, \lambda_n$$

$$e_1, e_2, \ldots, e_n$$

The Spectral Clustering Algorithm

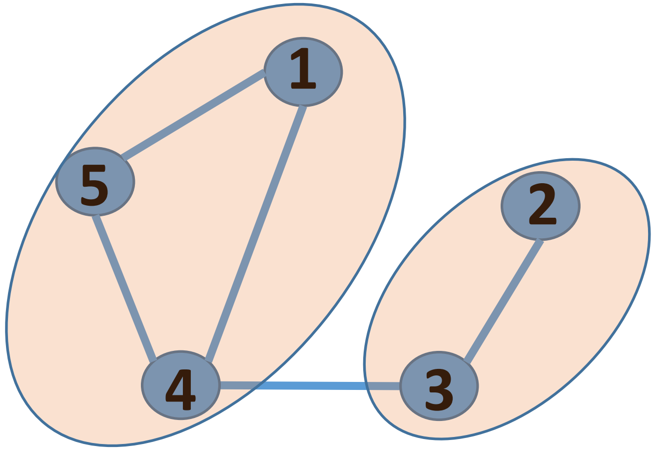
Uses the eigenvalues and vectors of the *graph Laplacian matrix* in order to find clusters (or “partitions”) of the graph



$$\begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$

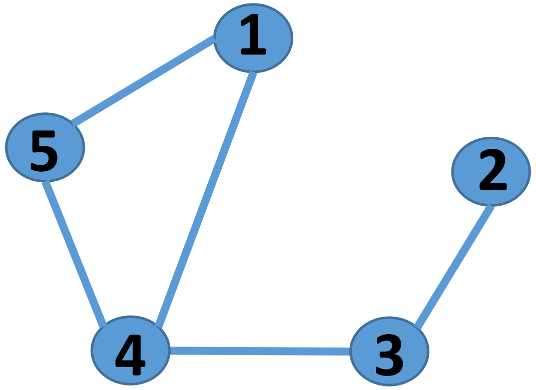
$$\lambda_1, \lambda_2, \ldots, \lambda_n$$

$$e_1, e_2, \ldots, e_n$$



A little history…

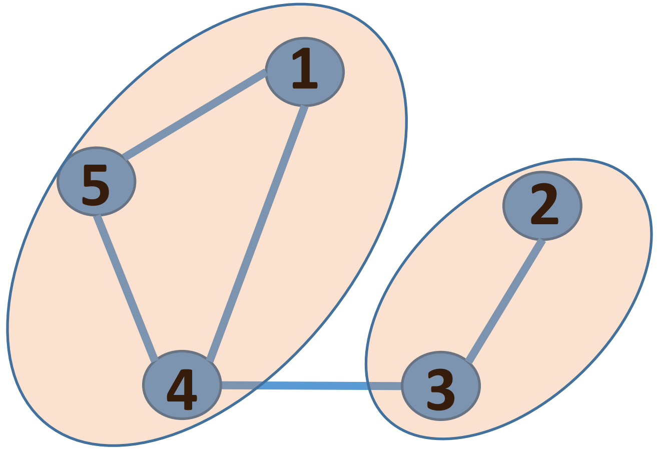
- 1973: *Algebraic Connectivity of Graph* M. Fiedler



$$\begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$

$$\lambda_1, \lambda_2, \dots, \lambda_n$$

$$e_1, e_2, \dots, e_n$$



A little history…

- 1973: *Algebraic Connectivity of Graph* M. Fiedler
- 1986: *Eigenvalues and expanders.* N. Alon.
- 1989: *Conductance and convergence of Markov chains-a combinatorial treatment of expanders*. M. Mihail

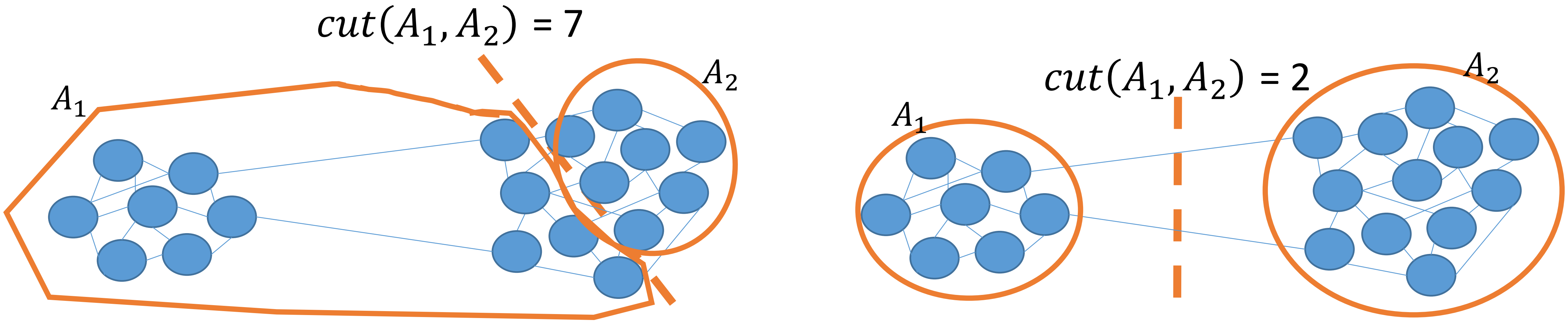
A little history…

- 1973: *Algebraic Connectivity of Graph* M. Fiedler
- 1986: *Eigenvalues and expanders.* N. Alon.
- 1989: *Conductance and convergence of Markov chains-a combinatorial treatment of expanders*. M. Mihail
- 1990 – *Partitioning Sparse Matrices with Eigenvectors of Graphs* A. Pothen et al.

A little history…

- 1973: *Algebraic Connectivity of Graph* M. Fiedler
- 1986: *Eigenvalues and expanders.* N. Alon.
- 1989: *Conductance and convergence of Markov chains-a combinatorial treatment of expanders*. M. Mihail
- 1990 – *Partitioning Sparse Matrices with Eigenvectors of Graphs* A. Pothen et al.
- 2000: *Normalized Cuts and Image Segmentation.* J. Shi and J. Malik
- 2001: *A Random Walks View of Spectral Segmentation*. M. Meila

A little history…

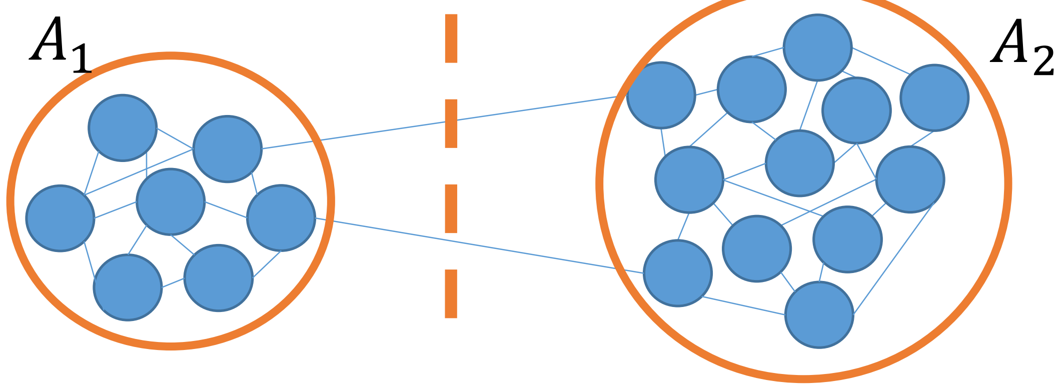


- 2000: *Normalized Cuts and Image Segmentation.* J. Shi and J. Malik
- 2001: *A Random Walks View of Spectral Segmentation*. M. Meila

A little history…

$$Ratiocut(A_1, A_2, \ldots, A_k) = \sum_{i=1,\ldots,k} \frac{cut(A_i, \overline{A_i})}{|A_i|}$$

$Ratiocut(A_1, A_2)$ = 0.226

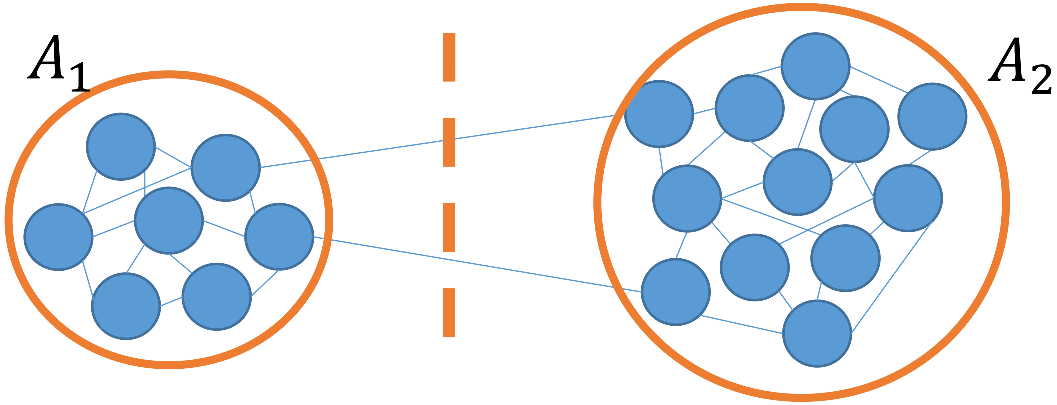


- 2000: *Normalized Cuts and Image Segmentation.* J. Shi and J. Malik
- 2001: *A Random Walks View of Spectral Segmentation*. M. Meila

A little history…

$$Ratiocut(A_1, A_2, \dots, A_k) = \sum_{i=1,\dots,k} \frac{cut(A_i, \overline{A_i})}{|A_i|}$$

$Ratiocut(A_1, A_2)$ = 0.226



- 2000: *Normalized Cuts and Image Segmentation.* J. Shi and J. Malik
- 2001: *A Random Walks View of Spectral Segmentation*. M. Meila

A little history…

- 1973: *Algebraic Connectivity of Graph* M. Fiedler
- 1986: *Eigenvalues and expanders.* N. Alon.
- 1989: *Conductance and convergence of Markov chains-a combinatorial treatment of expanders*. M. Mihail
- 1990 – *Partitioning Sparse Matrices with Eigenvectors of Graphs* A. Pothen et al.
- 2000: *Normalized Cuts and Image Segmentation.* J. Shi and J. Malik
- 2001: *A Random Walks View of Spectral Segmentation*. M. Meila
- 2010: *Power Iteration Clustering*. F. Lin and W. Cohen

Definition: Unnormalized Graph Laplacian Matrix

$$L = D \; - W$$

Where D is the diagonal degree matrix
and $W_{ij} \geq 0$ weight of edge (i, j), $i \neq j$

Definition: Unnormalized Graph Laplacian Matrix

$$L = D - W$$

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

$$\mathrm{W} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Definition: Unnormalized Graph Laplacian Matrix

$$L = D - W$$

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

Definition: Unnormalized Graph Laplacian Matrix

$$L = D - W$$

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

Definition: Unnormalized Graph Laplacian Matrix

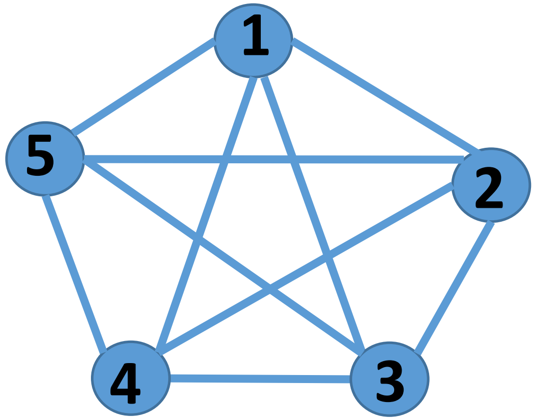
$$L = D - W$$

$$L = \begin{pmatrix} 3 & -3 & 0 & 0 & 0 \\ -3 & 4 & -1 & 0 & 0 \\ 0 & -1 & 5 & -4 & 0 \\ 0 & 0 & -4 & 9 & -5 \\ 0 & 0 & 0 & -5 & 5 \end{pmatrix}$$

Note: In a weighted graph, $D_{ii} = \sum_{j=1}^{n} w_{ij}$

Definition: Unnormalized Graph Laplacian Matrix

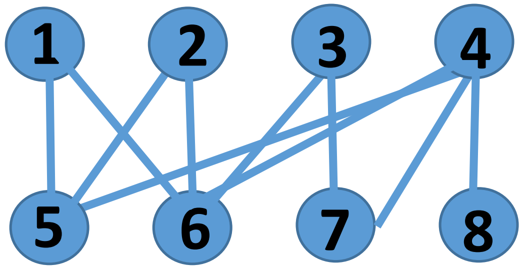
$$L = D - W$$



$$L = \begin{pmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix}$$

Definition: Unnormalized Graph Laplacian Matrix

$$L = D - W$$

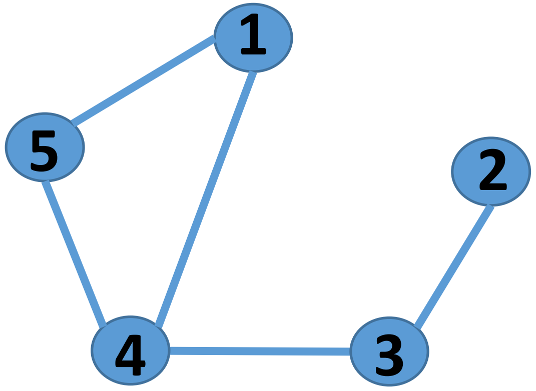


$$L = \begin{pmatrix} 2 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 2 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 4 & -1 & -1 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 0 & 4 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Properties of the Graph Laplacian

1. $\boldsymbol{f}^T \boldsymbol{L}\boldsymbol{f} = \frac{1}{2}\boldsymbol{\Sigma}_{\boldsymbol{i},\boldsymbol{j}=\mathbf{1}\dots\boldsymbol{n}} \boldsymbol{w}_{\boldsymbol{ij}}(\boldsymbol{f}_{\boldsymbol{i}} - \boldsymbol{f}_{\boldsymbol{j}})^2$ for all $\boldsymbol{f} \boldsymbol{\epsilon} \mathbb{R}^{\boldsymbol{n}}$

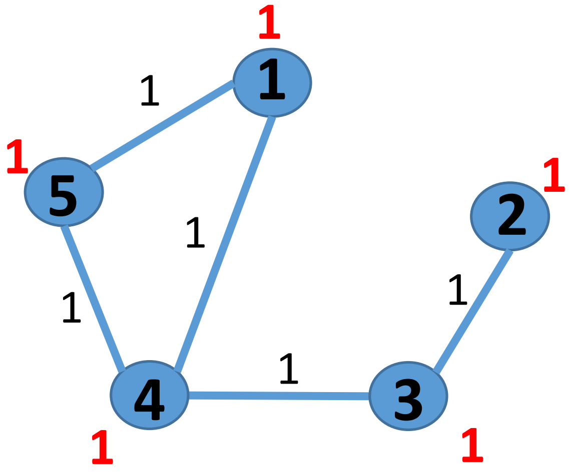
Note that in this property relates the difference in weights assigned to vertices to the quadratic form of L



1. $f^T L f = \frac{1}{2} \Sigma_{i,j=1\ldots n} w_{ij}(f_i - f_j)^2$ for all $f \epsilon \mathbb{R}^n$

$$f = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \Rightarrow f^T L f = 0$$

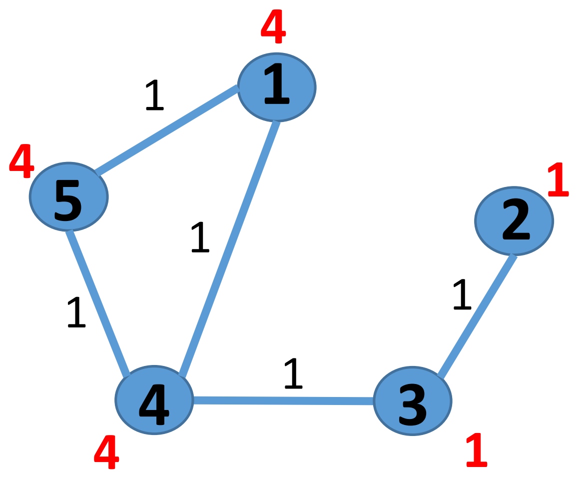
$$L = \begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$



$$1.\quad f^T L f = \frac{1}{2} \Sigma_{i,j=1\ldots n} w_{ij} (f_i - f_j)^2 \quad \text{for all } f \epsilon \mathbb{R}^n$$

$$f = \begin{pmatrix} 4 \\ 1 \\ 1 \\ 4 \\ 4 \end{pmatrix} \Rightarrow f^T L f = 9$$

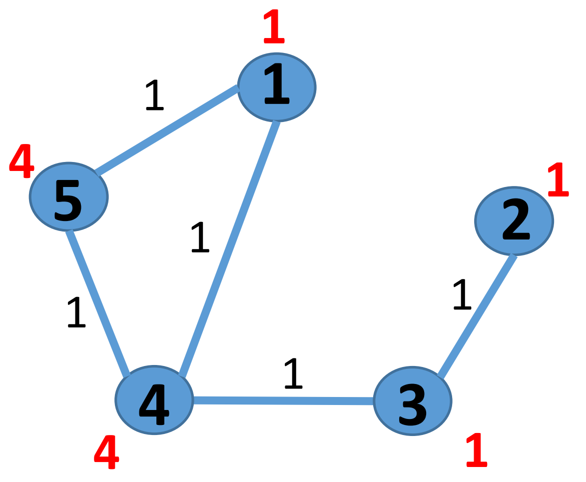
$$L = \begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$



1. $\boldsymbol{f^T L f = \frac{1}{2} \Sigma_{i,j=1\ldots n} w_{ij} (f_i - f_j)^2}$ for all $\boldsymbol{f} \epsilon \mathbb{R}^n$

$$\boldsymbol{f = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 4 \\ 4 \end{pmatrix} \Rightarrow f^T L f = 27}$$

$$L = \begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$



1. $\boldsymbol{f^T L f = \frac{1}{2} \Sigma_{i,j=1\ldots n} w_{ij} (f_i - f_j)^2}$ for all $\boldsymbol{f \epsilon \mathbb{R}^n}$

1. $\boldsymbol{f^T L f = \frac{1}{2} \Sigma_{i,j=1 \ldots n} w_{ij} (f_i - f_j)^2}$ for all $\boldsymbol{f} \boldsymbol{\epsilon} \mathbb{R}^{\boldsymbol{n}}$

$$(f_1 \ldots f_n)(D - W)\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}$$

1. $\boldsymbol{f^T L f = \frac{1}{2}\Sigma_{i,j=1\ldots n} w_{ij}(f_i - f_j)^2}$ for all $\boldsymbol{f} \boldsymbol{\epsilon} \mathbb{R}^{\boldsymbol{n}}$

$$(f_1 \ldots f_n)(D - W)\begin{pmatrix} f_1 \\ f_n \end{pmatrix}$$

$$= (f_1 \ldots f_n)\begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix}\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} - (f_1 \ldots f_n)\begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{pmatrix}\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}$$

1. $\boldsymbol{f^T L f = \frac{1}{2} \Sigma_{i,j=1\dots n} w_{ij}(f_i - f_j)^2}$ for all $\boldsymbol{f} \boldsymbol{\epsilon} \mathbb{R}^{\boldsymbol{n}}$

$$(f_1 \dots f_n)(D - W)\begin{pmatrix} f_1 \\ f_n \end{pmatrix}$$

$$(f_1 \dots f_n)\begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix}\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} - (f_1 \dots f_n)\begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{pmatrix}\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}$$

$$= \sum_{i=1}^{n} d_i \, {f_i}^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} f_i f_j$$

1. $\boldsymbol{f^T L f = \frac{1}{2} \Sigma_{i,j=1\ldots n} w_{ij} (f_i - f_j)^2}$ for all $\boldsymbol{f \epsilon} \mathbb{R}^{\boldsymbol{n}}$

$$(f_1 \ldots f_n)(D-W)\begin{pmatrix} f_1 \\ f_n \end{pmatrix}$$

$$(f_1 \ldots f_n)\begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix}\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} - (f_1 \ldots f_n)\begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{pmatrix}\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}$$

$$= \sum_{i=1}^{n} d_i\, {f_i}^2 - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n} d_i\, {f_i}^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n} d_j\, {f_j}^2\right)$$

1. $\boldsymbol{f^T L f = \frac{1}{2} \Sigma_{i,j=1\dots n} w_{ij}(f_i - f_j)^2}$ for all $\boldsymbol{f} \boldsymbol{\epsilon} \mathbb{R}^{\boldsymbol{n}}$

$$(f_1 \dots f_n)(D - W)\begin{pmatrix} f_1 \\ f_n \end{pmatrix}$$

$$(f_1 \dots f_n)\begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix}\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} - (f_1 \dots f_n)\begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{pmatrix}\begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix}$$

$$= \sum_{i=1}^{n} d_i\, f_i^{\,2} - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n} d_i\, f_i^{\,2} - 2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n} d_j\, f_j^{\,2}\right)$$

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

2. L is symmetric and positive semi-definite

2. L is symmetric and positive semi-definite

Recall the definition: $L = D - W$

D diagonal, W symmetric $\rightarrow$ L symmetric

$$L = \begin{pmatrix} 3 & -3 & 0 & 0 & 0 \\ -3 & 4 & -1 & 0 & 0 \\ 0 & -1 & 5 & -4 & 0 \\ 0 & 0 & -4 & 9 & -5 \\ 0 & 0 & 0 & -5 & 5 \end{pmatrix}$$

2. L is symmetric and positive semi-definite

Recall the definition: $L = D - W$

D diagonal, W symmetric $\rightarrow L$ symmetric

Recall property 1:

$$f^T L f = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} w_{ij} (f_i - f_j)^2 \geq 0 \quad \forall f \in \mathbb{R}^n$$

3. The smallest eigenvalue of L is 0 with corresponding eigenvector being the constant vector $\mathbb{1}$

3. The smallest eigenvalue of L is 0 with corresponding eigenvector being the constant vector $\mathbb{1}$

$$L = D - W = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix} - \begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{pmatrix}$$

3. The smallest eigenvalue of L is 0 with corresponding eigenvector being the constant vector $\mathbb{1}$

$$L = D - W = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix} - \begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{j=1}^{n} w_{1j} - w_{11} & \cdots & -w_{1n} \\ \vdots & \ddots & \vdots \\ -w_{n1} & \cdots & \sum_{j=1}^{n} w_{nj} - w_{nn} \end{pmatrix}$$

3. The smallest eigenvalue of L is 0 with corresponding eigenvector being the constant vector $\mathbb{1}$

$$L = D - W = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix} - \begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{j=1}^{n} w_{1j} - w_{11} & \cdots & -w_{1n} \\ \vdots & \ddots & \vdots \\ -w_{n1} & \cdots & \sum_{j=1}^{n} w_{nj} - w_{nn} \end{pmatrix}$$

$$\Rightarrow L\mathbf{1} = \begin{pmatrix} \sum_{j=1}^{n} w_{1j} - w_{11} & \cdots & -w_{1n} \\ \vdots & \ddots & \vdots \\ -w_{n1} & \cdots & \sum_{j=1}^{n} w_{nj} - w_{nn} \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

3. The smallest eigenvalue of L is 0 with corresponding eigenvector being the constant vector $\mathbb{1}$

$$L = D - W = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix} - \begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{j=1}^{n} w_{1j} - w_{11} & \cdots & -w_{1n} \\ \vdots & \ddots & \vdots \\ -w_{n1} & \cdots & \sum_{j=1}^{n} w_{nj} - w_{nn} \end{pmatrix}$$

$$\Rightarrow L\mathbf{1} = \begin{pmatrix} \sum_{j=1}^{n} w_{1j} - w_{11} & \cdots & -w_{1n} \\ \vdots & \ddots & \vdots \\ -w_{n1} & \cdots & \sum_{j=1}^{n} w_{nj} - w_{nn} \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{j=1}^{n} w_{1j} - w_{11} - \sum_{j=2}^{n} w_{1j} \\ \vdots \\ \sum_{j=1}^{n-1} (-w_{nj}) - \sum_{j=1}^{n} w_{nj} - w_{nn} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

3. The smallest eigenvalue of L is 0 with corresponding eigenvector being the constant vector $\mathbb{I}$

$$L = D - W = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix} - \begin{pmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{j=1}^{n} w_{1j} - w_{11} & \cdots & -w_{1n} \\ \vdots & \ddots & \vdots \\ -w_{n1} & \cdots & \sum_{j=1}^{n} w_{nj} - w_{nn} \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} \sum_{j=1}^{n} w_{1j} - w_{11} & \cdots & -w_{1n} \\ \vdots & \ddots & \vdots \\ -w_{n1} & \cdots & \sum_{j=1}^{n} w_{nj} - w_{nn} \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} =$$

$$\begin{pmatrix} \sum_{j=1}^{n} w_{1j} - w_{11} - \sum_{j=2}^{n} w_{1j} \\ \vdots \\ \sum_{j=1}^{n-1} (-w_{nj}) - \sum_{j=1}^{n} w_{nj} - w_{nn} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

$\Rightarrow$0 is an eigenvalue and its corresponding eigenvector is the constant vector $\mathbb{I}$

$\Rightarrow$Since each eigenvalue is ≥ 0 **(property 2),** this is the smallest eigenvalue

4. L has non-negative, real-valued eigenvalues $\mathbf{0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n}$

By the property 2, L is symmetric $\rightarrow L$ has n real eigenvalues

4. L has non-negative, real-valued eigenvalues

$$\mathbf{0} = \boldsymbol{\lambda_1} \leq \boldsymbol{\lambda_2} \leq \cdots \leq \boldsymbol{\lambda_n}$$

By the property 2, L is symmetric $\rightarrow L$ has n real eigenvalues

Again by property 2, L is positive-semidefinite

$$\rightarrow \mathbf{0} \leq \boldsymbol{\lambda_1} \leq \boldsymbol{\lambda_2} \leq \cdots \leq \boldsymbol{\lambda_n}$$

4. L has non-negative, real-valued eigenvalues

$$\mathbf{0} = \boldsymbol{\lambda}_{\mathbf{1}} \leq \boldsymbol{\lambda}_{\mathbf{2}} \leq \cdots \leq \boldsymbol{\lambda}_{\boldsymbol{n}}$$

By the property 2, L is symmetric $\rightarrow L$ has n real eigenvalues

Again by property 2, L is positive-semidefinite

$$\rightarrow \mathbf{0} \leq \boldsymbol{\lambda}_{\mathbf{1}} \leq \boldsymbol{\lambda}_{\mathbf{2}} \leq \cdots \leq \boldsymbol{\lambda}_{\boldsymbol{n}}$$

By property 3, $\boldsymbol{\lambda}_{\mathbf{1}} = \mathbf{0}$

4. L has non-negative, real-valued eigenvalues $\mathbf{0 = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n}$

By the property 2, L is symmetric $\rightarrow L$ has n real eigenvalues

Again by property 2, L is positive-semidefinite

$$\rightarrow \mathbf{0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n}$$

By property 3, $\boldsymbol{\lambda_1 = 0}$

Therefore, $\mathbf{0 = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n}$

Properties of Graph Laplacians: Summary

$$L = D - W$$

1. $f'Lf = \frac{1}{2}\Sigma_{i,j=1\ldots n} w_{ij}(f_i - f_j)^2$ for all $f \epsilon \mathbb{R}^n$
2. L is **symmetric** and **positive semi-definite**
3. The smallest eigenvalue of L is 0, and the corresponding eigenvector is the constant vector $\mathbb{1}$
4. L has n nonnegative, real eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

<u>Proposition</u>: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Suppose $k = 1$.

<u>Proposition</u>: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Suppose $k = 1$.

Suppose further that $\boldsymbol{f}$ is an eigenvector with eigenvalue 0, i.e., $L\boldsymbol{f} = \mathbf{0}$

<u>Proposition</u>: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Suppose $k = 1$.

Suppose further that $\boldsymbol{f}$ is an eigenvector with eigenvalue 0, i.e., $L\boldsymbol{f} = \boldsymbol{0}$

$$\rightarrow\ 0 = \boldsymbol{f}^T L \boldsymbol{f} = \frac{1}{2}\Sigma_{i,j=1\ldots n} W_{ij}(\boldsymbol{f}_i - \boldsymbol{f}_j)^2$$

Proposition: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Suppose $k = 1$.

Suppose further that $\boldsymbol{f}$ is an eigenvector with eigenvalue 0, i.e., $L\boldsymbol{f} = \boldsymbol{0}$

$$\rightarrow\ 0 = \boldsymbol{f}^T L \boldsymbol{f} = \frac{1}{2} \Sigma_{i,j=1\ldots n} W_{ij} (\boldsymbol{f}_i - \boldsymbol{f}_j)^2$$

Since all weights are positive,

$\rightarrow \boldsymbol{f}_i = \boldsymbol{f}_j$ for all connected vertices i, j

<u>Proposition</u>: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Suppose $k = 1$.

Suppose further that $\boldsymbol{f}$ is an eigenvector with eigenvalue 0, i.e., $L\boldsymbol{f} = \mathbf{0}$

$$\rightarrow \ 0 = \boldsymbol{f}^T L \boldsymbol{f} = \frac{1}{2} \Sigma_{i,j=1\ldots n} W_{ij} (\boldsymbol{f}_i - \boldsymbol{f}_j)^2$$

Since all weights are positive,

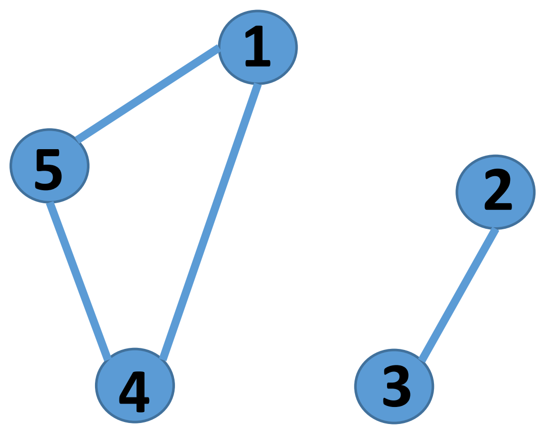
$\rightarrow \boldsymbol{f}_i = \boldsymbol{f}_j$ for all connected vertices i, j

Thus $\boldsymbol{f}_{\boldsymbol{v_1}} = \boldsymbol{f}_{\boldsymbol{v_2}} = \cdots = \boldsymbol{f}_{\boldsymbol{v_m}}$ for any path over vertices $v_1, v_2, \ldots, v_m$

Proposition: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Now suppose that $k > 1$.

WLOG, order the vertices according to the components they belong to

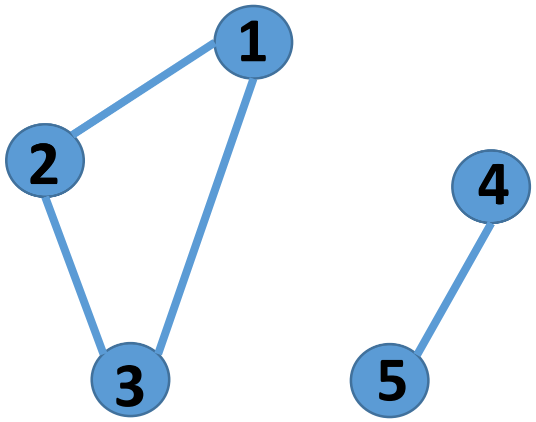


$$L = \begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$

Proposition: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Now suppose that $k > 1$.

WLOG, order the vertices according to the components they belong to



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

<u>Proposition</u>: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Now suppose that $k > 1$.

WLOG, order the vertices according to the components they belong to

$\rightarrow L$ has a block diagonal form

$$L = \begin{pmatrix} L_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & L_k \end{pmatrix}$$

<u>Proposition</u>: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Now suppose that $k > 1$.

WLOG, order the vertices according to the components they belong to

$\rightarrow L$ has a block diagonal form

The spectrum of L is the union of the spectra of $L_1, \dots, L_k$

$$L = \begin{pmatrix} L_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & L_k \end{pmatrix}$$

Proposition: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Now suppose that $k > 1$.

WLOG, order the vertices according to the components they belong to

$\rightarrow L$ has a block diagonal form

The spectrum of L is the union of the spectra of $L_1, \dots, L_k$

Since each L_i represents a connected component, the eigenvectors corresponding to $\lambda_1 = 0$ of L are constant for the block L_i and 0 elsewhere

$$\begin{pmatrix} L_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & L_k \end{pmatrix} \begin{pmatrix} \mathbf{1} \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

<u>Proposition</u>: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Now suppose that $k > 1$.

WLOG, order the vertices according to the components they belong to

$\rightarrow L$ has a block diagonal form

The spectrum of L is the union of the spectra of $L_1, \ldots, L_k$

Since each L_i represents a connected component, the eigenvectors corresponding to $\lambda_1 = 0$ of L are constant for the block L_i and 0 elsewhere

$$\begin{pmatrix} L_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & L_k \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ \mathbf{1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

<u>Proposition</u>: The number of connected components of a graph is equal to the multiplicity k of the first eigenvalue of the graph Laplacian matrix

Now suppose that $k > 1$.

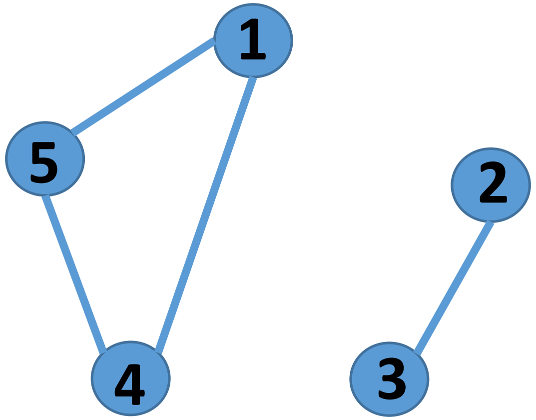
WLOG, order the vertices according to the components they belong to

$\rightarrow L$ has a block diagonal form

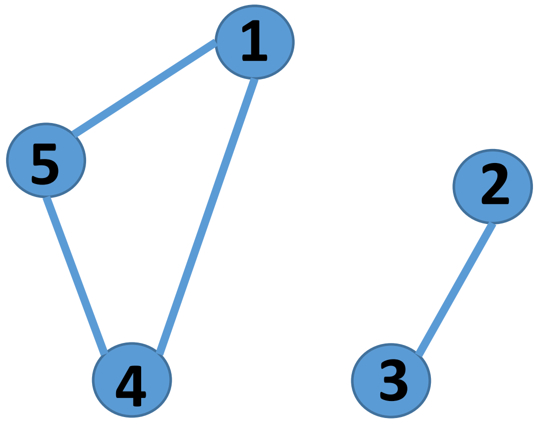
The spectrum of L is the union of the spectra of $L_1, \dots, L_k$

Since each L_i represents a connected component, the eigenvectors corresponding to $\lambda_1 = 0$ of L are constant for the block L_i and 0 elsewhere

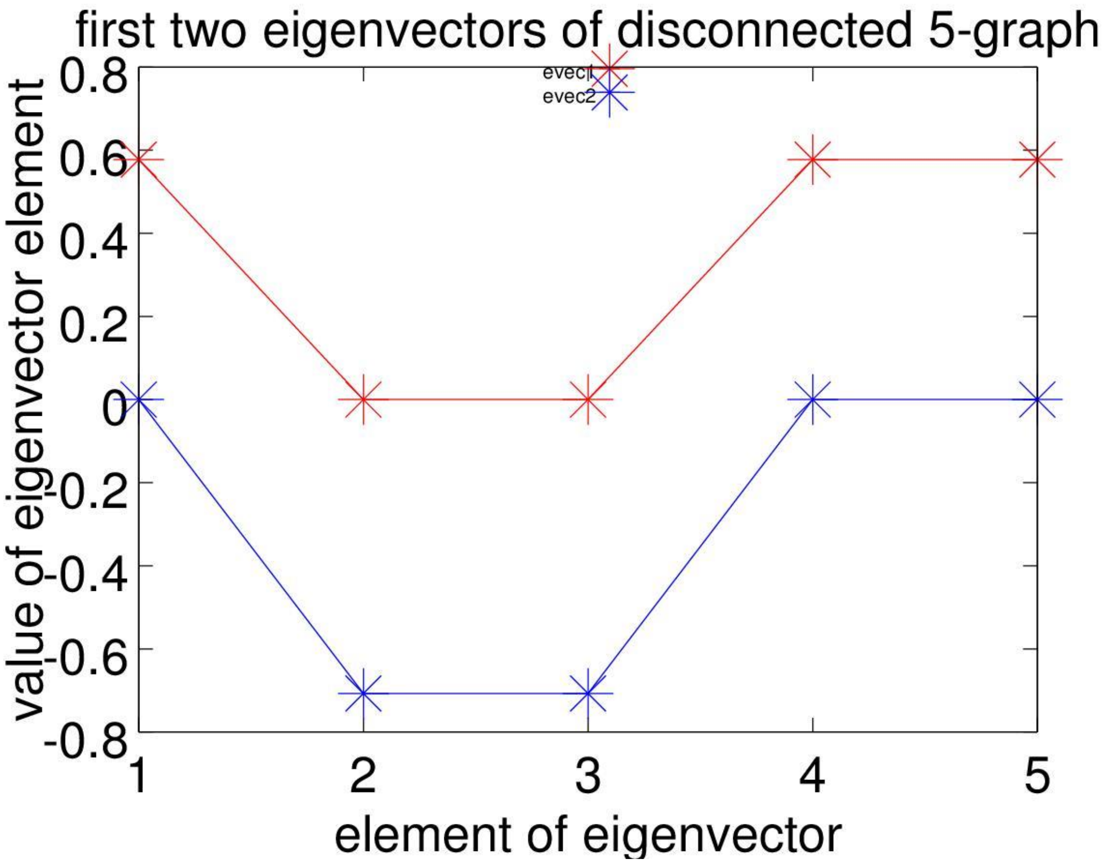
Thus, L has as many 0 eigenvalues as there are connected components

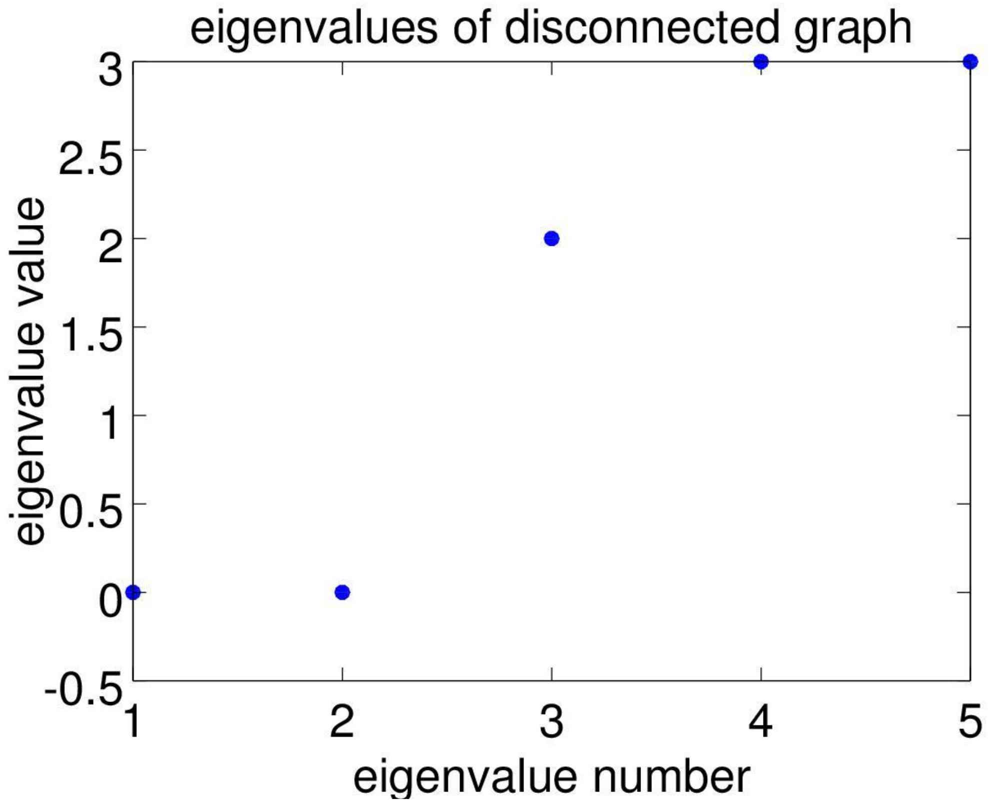


$$L = \begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$



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





Spectral Clustering Algorithm

Given: A graph with n vertices and edge weights W_{ij} , number of desired clusters k

1. Construct (normalized) graph Laplacian $L\big(G(V,E)\big) = D - W$
2. Find the k eigenvectors corresponding to the k smallest eigenvalues of L
3. Let U be the $\mathrm{n} \times k$ matrix of eigenvectors
4. Use k-means to find k clusters C' letting x_i' be the rows of U
5. Assign data point x_i to the jth cluster if x_i' was assigned to cluster j

Graph Cut Point of View

The goal of spectral clustering is to minimize the sum of the weights of edges between clusters

Graph Cut Point of View

The goal of spectral clustering is to minimize the sum of the weights of edges between clusters

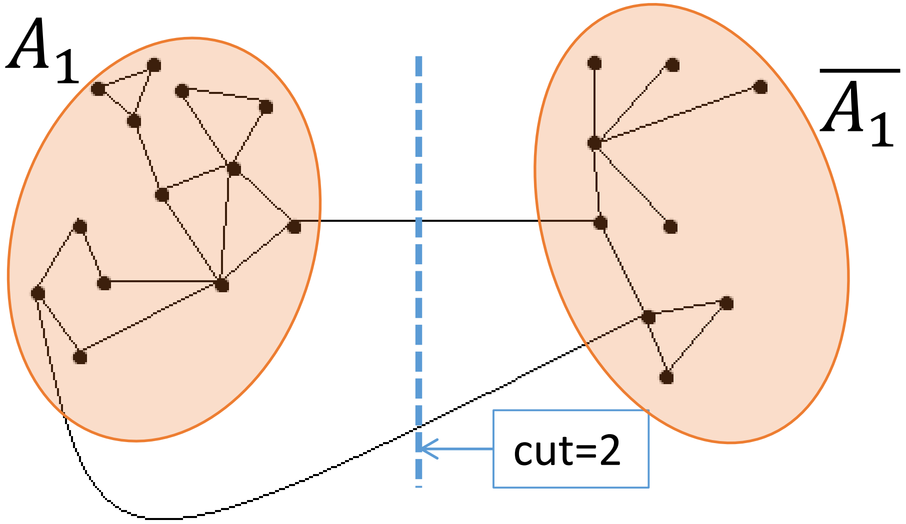
Define: $\text{cut}(A_1, \ldots, A_k) = \frac{1}{2} \Sigma_{i=1\ldots k} W(A_i, \bar{A}_i)$ where $W(A_i, \bar{A}_i) = \sum_{j \in A_i, l \in \overline{A_i}} w_{jl}$

Graph Cut Point of View

The goal of spectral clustering is to minimize the sum of the weights of edges between clusters

Define: cut$(A_1, \ldots, A_k) = \frac{1}{2}\Sigma_{i=1\ldots k} W(A_i, \bar{A}_i)$ where $W(A_i, \bar{A}_i) = \sum_{j \in A_i, l \in \overline{A_i}} w_{jl}$

Ex: $\boldsymbol{k} = 2$

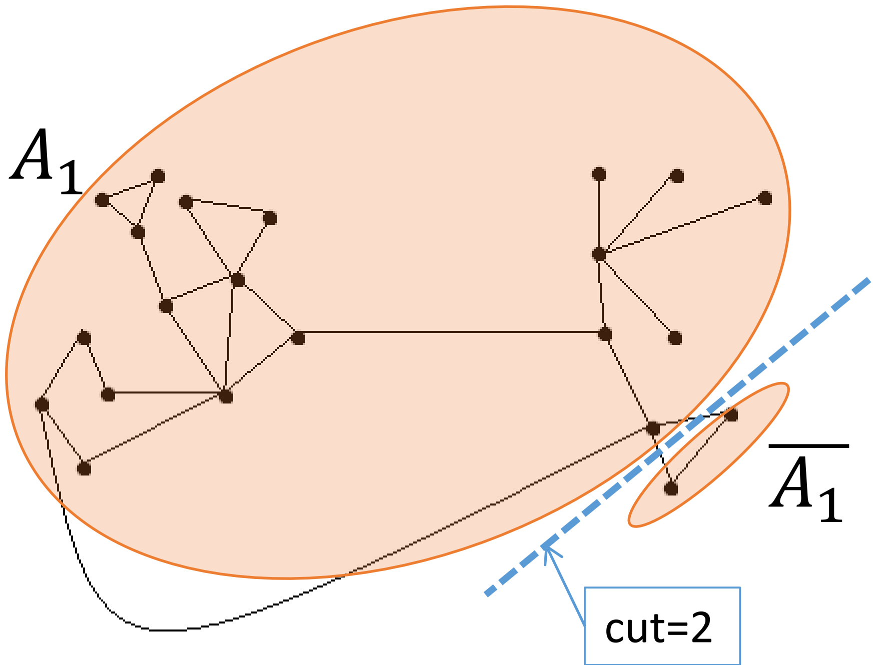


Graph Cut Point of View

The goal of spectral clustering is to minimize the sum of the weights of edges between clusters

Define: $\mathrm{cut}(A_1, \ldots, A_k) = \frac{1}{2}\Sigma_{i=1\ldots k} W(A_i, \bar{A}_i)$ where $W(A_i, \bar{A}_i) = \sum_{j \in A_i, l \in \overline{A_i}} w_{jl}$

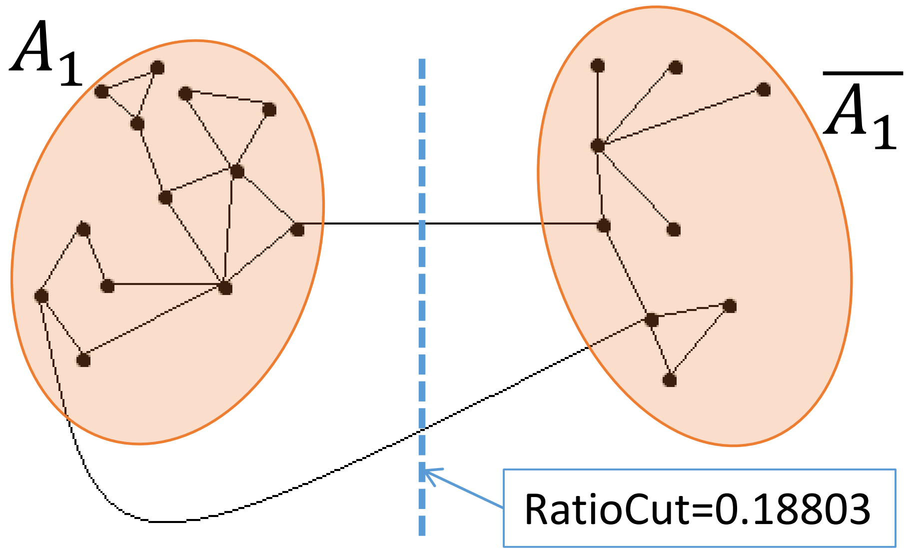
Ex: $k = 2$



Graph Cut Point of View

To create balanced clusters, minimize the *RatioCut* instead:

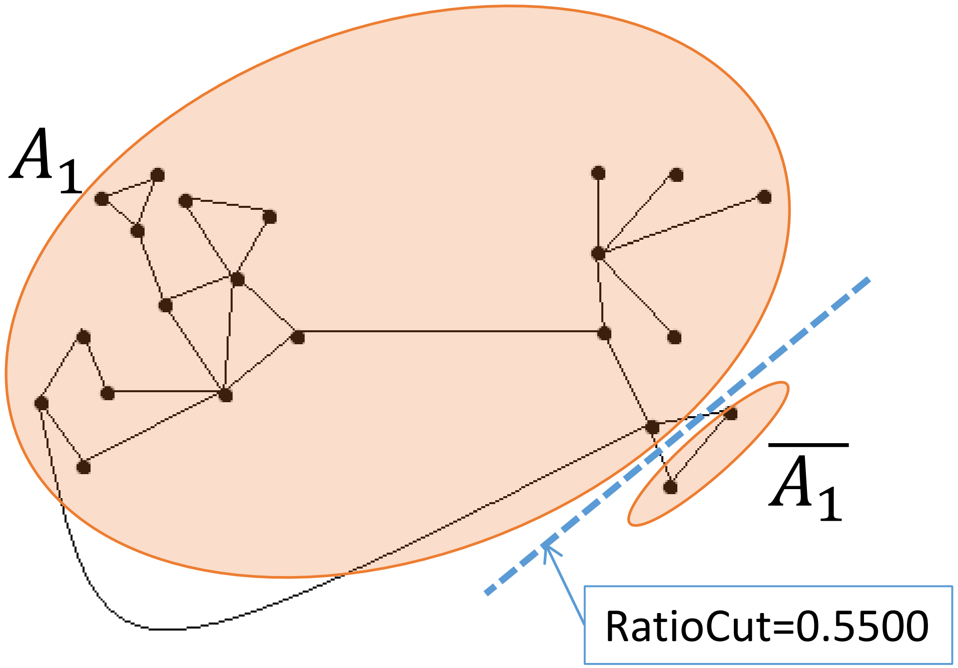
$$\text{RatioCut}(A_1, \ldots, A_k) = \frac{1}{2}\Sigma_{i=1\ldots k}\frac{W(A_i,\bar{A}_i)}{|A_i|} = \Sigma_{i=1\ldots k}\frac{cut(A_i,\bar{A}_i)}{|A_i|}$$



Graph Cut Point of View

To create balanced clusters, minimize the *RatioCut* instead:

$$\text{RatioCut}(A_1, \ldots, A_k) = \frac{1}{2}\Sigma_{i=1\ldots k}\frac{W(A_i,\bar{A}_i)}{|A_i|} = \Sigma_{i=1\ldots k}\frac{cut(A_i,\bar{A}_i)}{|A_i|}$$



…but minimizing RatioCut is NP-hard!! ☹

$$\min_{A_1, A_2, \ldots, A_k} \frac{1}{2} \Sigma_{i=1\ldots k} \frac{W(A_i, \bar{A}_i)}{|A_i|}$$

When the going gets tough…

We relax constraints on the problem ☺

Minimizing Ratio Cut for k=2

Only two clusters, A and $\bar{A}$ (complement of A)

Minimizing Ratio Cut for k=2

Only two clusters, A and $\bar{A}$ (complement of A)

Want to minimize: $\min_{A,\bar{A}} \frac{1}{2}\left(\frac{W(A,\bar{A})}{|A|} + \frac{W(\bar{A},A)}{|\bar{A}|}\right)$

Minimizing Ratio Cut for k=2

Only two clusters, A and $\bar{A}$ (complement of A)

Want to minimize: $\min_{A,\bar{A}} \frac{1}{2}\left(\frac{W(A,\bar{A})}{|A|} + \frac{W(\bar{A},A)}{|\bar{A}|}\right)$

Trick: find a vector f so that minimizing RatioCut$(A, \bar{A})$ is the same as minimizing $f^T L f$ subject to certain constraints.

Minimizing Ratio Cut for k=2

Goal: minimize: $\min_{A,\bar{A}} \frac{1}{2}\left(\frac{W(A,\bar{A})}{|A|} + \frac{W(\bar{A},A)}{|\bar{A}|}\right)$

Let: $$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} \ if\ v_i \in A \\ -\sqrt{|A|/|\bar{A}|} \ if\ v_i \in \ \bar{A} \end{cases}$$

Minimizing Ratio Cut for k=2

Goal: minimize: $\min_{A,\bar{A}} \frac{1}{2}\left(\frac{W(A,\bar{A})}{|A|} + \frac{W(\bar{A},A)}{|\bar{A}|}\right)$

Let: $$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} \ if\ v_i \in A \\ -\sqrt{|A|/|\bar{A}|} \ if\ v_i \in \ \bar{A} \end{cases}$$

Then, we can show that $f^T L f = |V|\text{RatioCut}(A, \bar{A})$

Minimizing Ratio Cut for k=2

Goal: minimize: $\min_{A,\bar{A}} \frac{1}{2}\left(\frac{W(A,\bar{A})}{|A|} + \frac{W(\bar{A},A)}{|\bar{A}|}\right)$

Let: $$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} \ \ if\ v_i \in A \\ -\sqrt{|A|/|\bar{A}|} \ \ if\ v_i \in \ \bar{A} \end{cases}$$

Then, we can show that $f^T L f = |V|\text{RatioCut}(A, \bar{A})$

Thus minimizing $\text{RatioCut}(A, \bar{A})$ is the same as minimizing $f^T L f$ with constraints:

Minimizing Ratio Cut for k=2

Goal: minimize: $\min_{A,\bar{A}} \frac{1}{2}\left(\frac{W(A,\bar{A})}{|A|} + \frac{W(\bar{A},A)}{|\bar{A}|}\right)$

Let: $$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} \ \ if\ v_i \in A \\ -\sqrt{|A|/|\bar{A}|} \ \ if\ v_i \in \ \bar{A} \end{cases}$$

Then, we can show that $f^T L f = |V|\text{RatioCut}(A, \bar{A})$

Thus minimizing $\text{RatioCut}(A, \bar{A})$ is the same as minimizing $f^T L f$ with constraints:

$\sum_{i=1}^{n} f_i = 0 \Rightarrow f \perp \mathbb{1}$

and $\|f\|_2{}^2 = n$, the number of vertices

and f_i taking on discrete values as defined above

Now relax the constraints:

Minimize $f^T L f$ subject to: $f \perp \mathbb{1}$, $||f||_2{}^2 = n$

Now relax the constraints:

Minimize $f^T L f$ subject to: $f \perp \mathbb{1}$, $||f||_2{}^2 = n$

The Rayleigh-Ritz theorem tells us that the solution to this optimization problem is given by e_2, the eigenvector corresponding to λ_2

Now relax the constraints:

Minimize $f^T L f$ subject to: $f \perp \mathbb{1}, \|f\|_2{}^2 = n$

The Rayleigh-Ritz theorem tells us that the solution to this optimization problem is given by e_2, the eigenvector corresponding to λ_2

So we can approximate RatioCut by using the second eigenvector (corresponding to the second smallest eigenvalue) of L!!!! (this translates to solving for eigenvectors in the Spectral Clustering algorithm)

How to transform approximate solution back to discrete setting?

Recall that we eliminated the constraint which assigned vertices to partitions:

$$f_i = \begin{cases} \sqrt{|\bar{A}|/|A|} \ if \ v_i \in A \\ -\sqrt{|A|/|\bar{A}|} \ if \ v_i \in \bar{A} \end{cases}$$

We can choose a threshold t, assigning all i for which $f_i > t$ to one partition A and all i for which $f_i \leq t$ to the other partition $\bar{A}$

Approximating RatioCut for $k > 2$

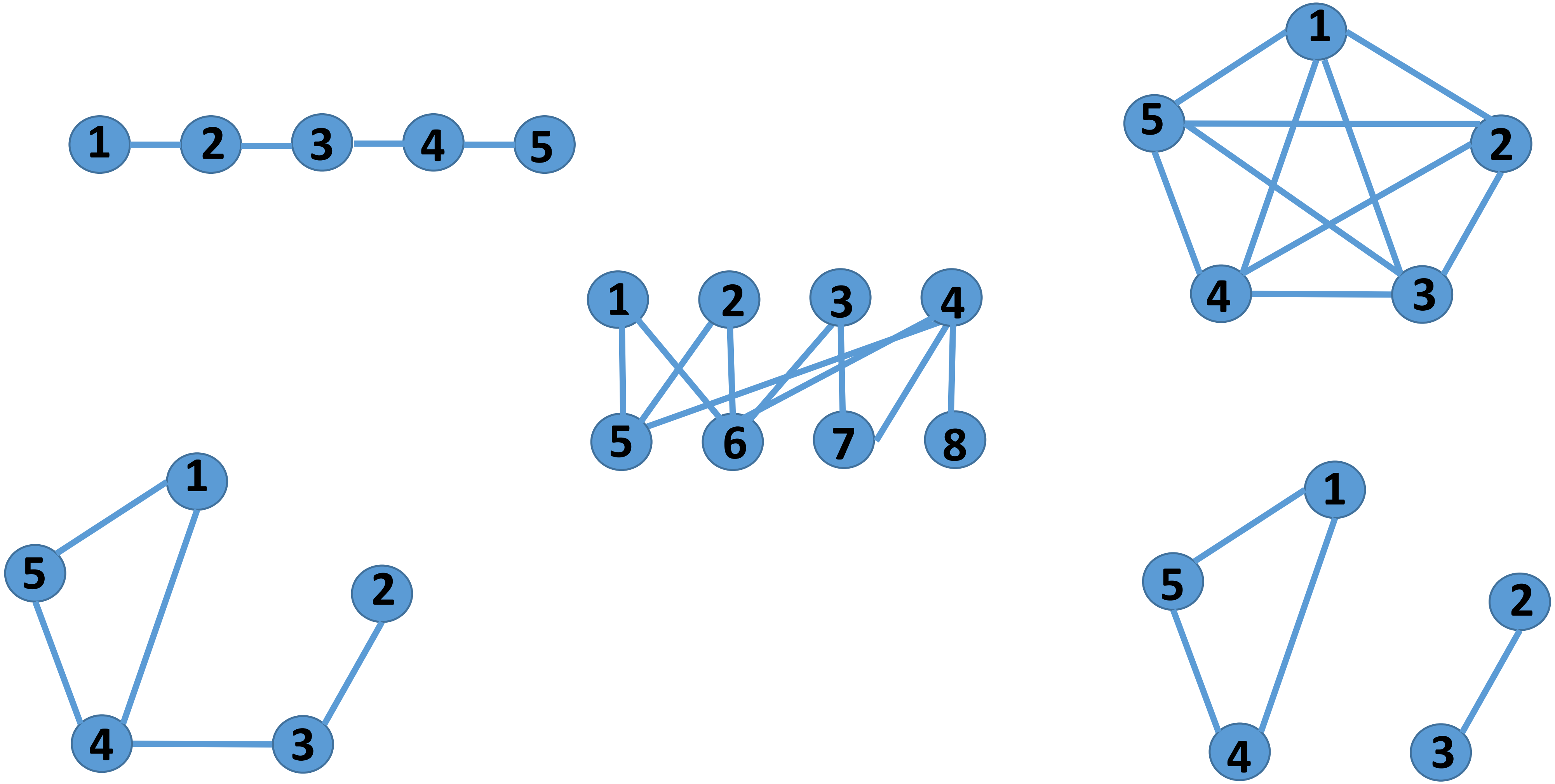
We can use similar arguments to show that solving minRatioCut for $k > 2$ corresponds to finding the first k eigenvectors of L

Spectral Clustering Algorithm

Given: A graph with n vertices and edge weights W_{ij} , number of desired clusters k

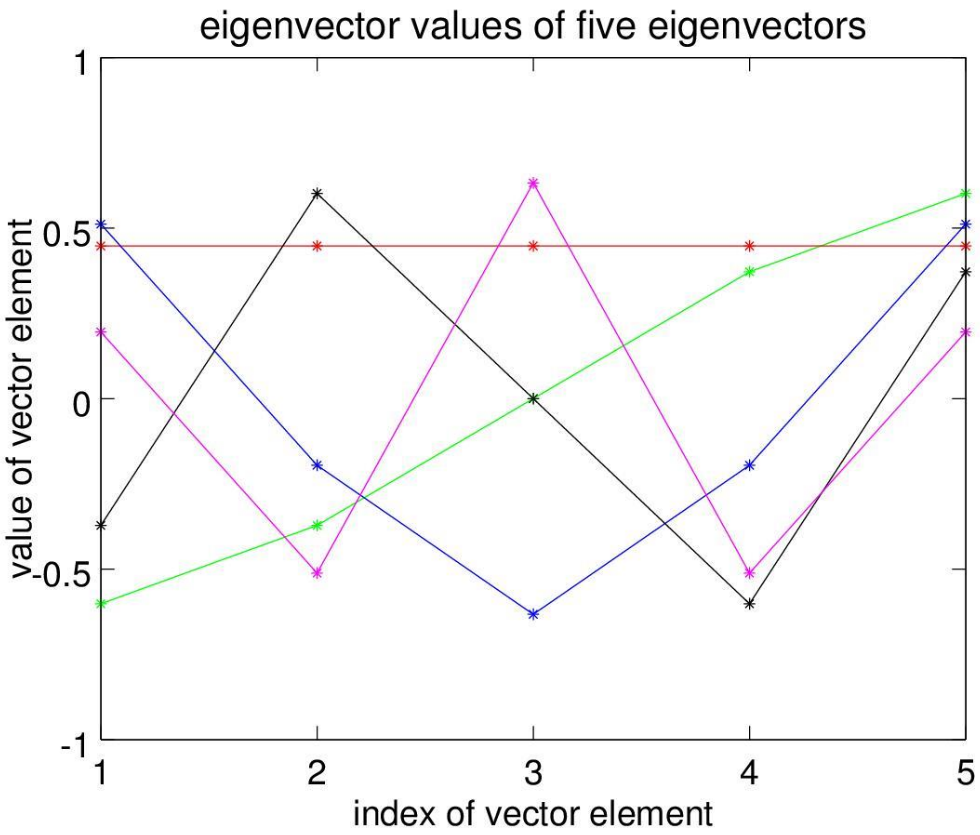
1. Construct (normalized) graph Laplacian $L\big(G(V,E)\big) = D - W$
2. Find the k eigenvectors corresponding to the k smallest eigenvalues of L
3. Let U be the $\mathrm{n} \times k$ matrix of eigenvectors
4. Use k-means to find k clusters C' letting x_i' be the rows of U
5. Assign data point x_i to the jth cluster if x_i' was assigned to cluster j

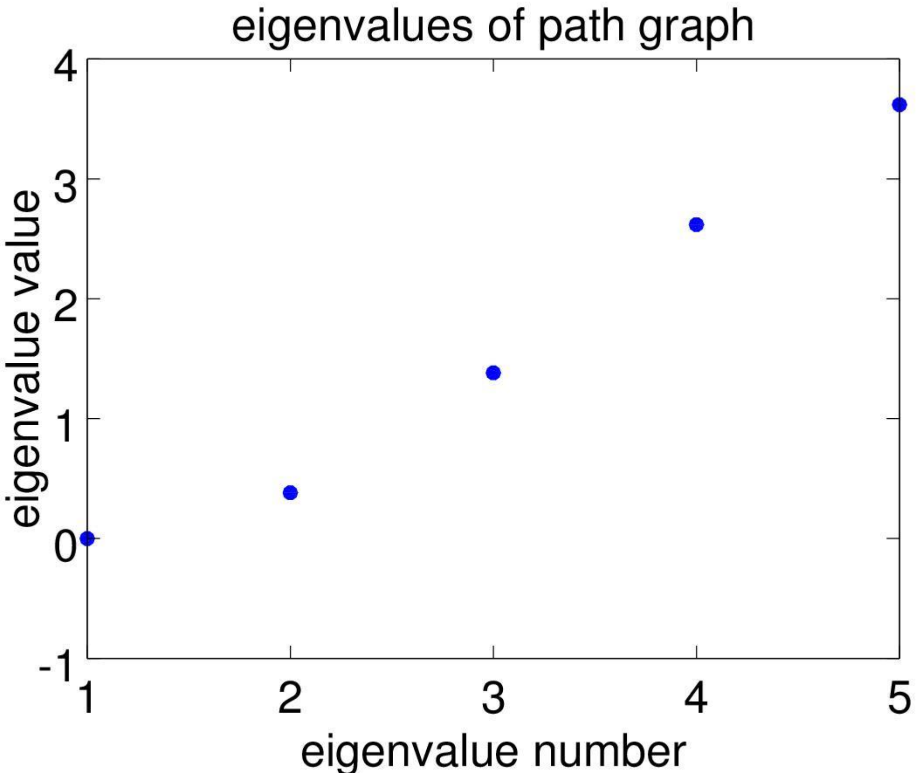
Examples:



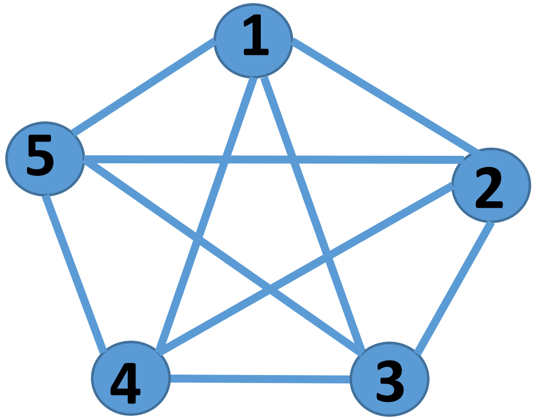
Path Graph

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

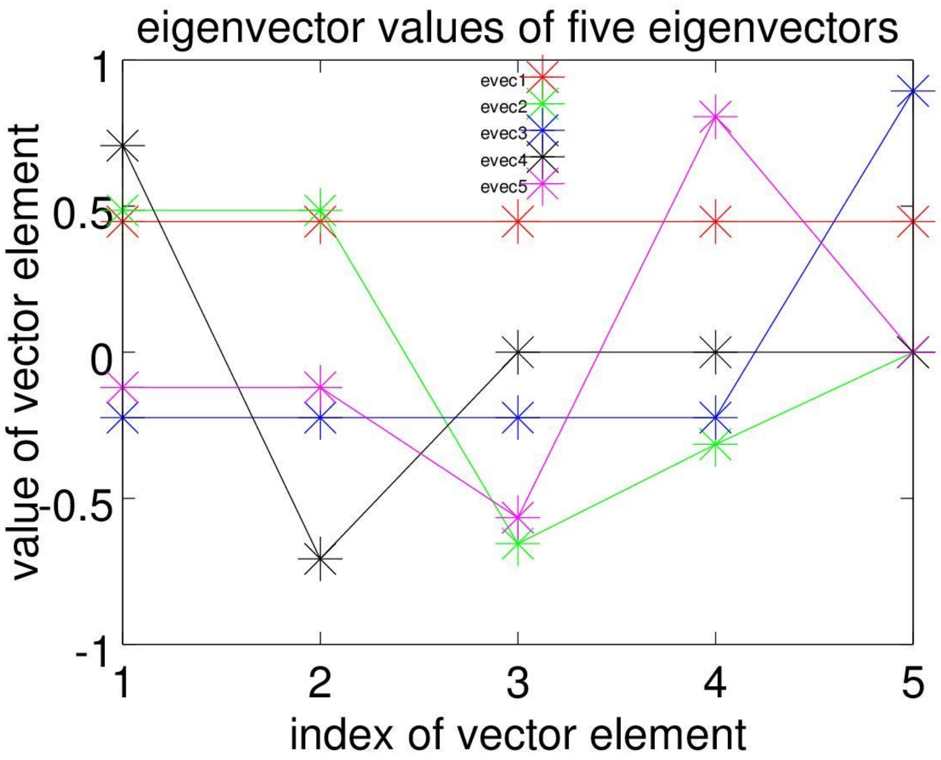


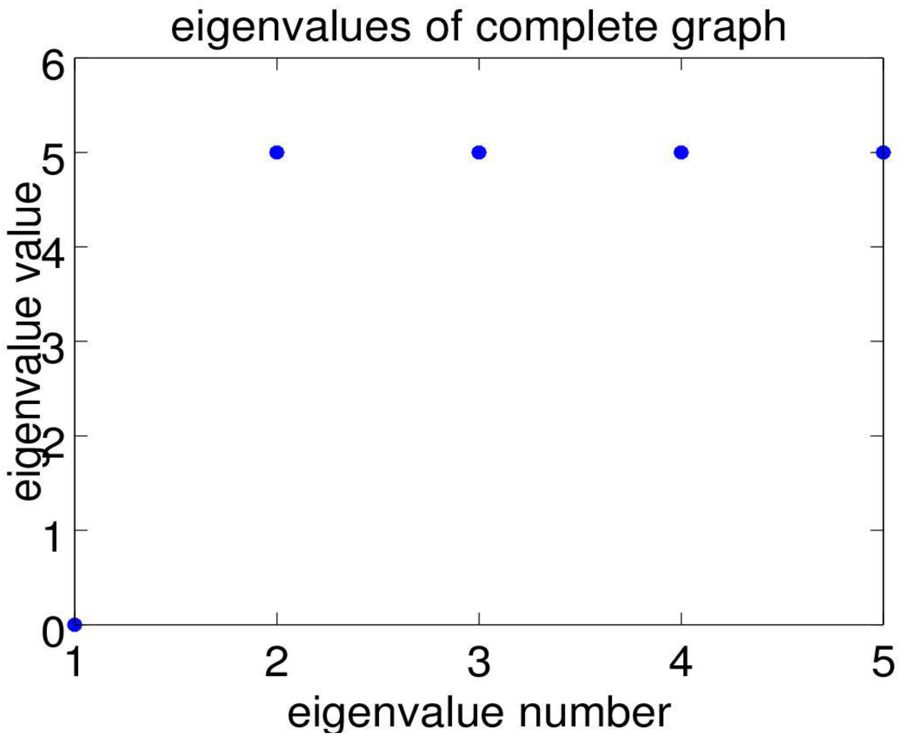


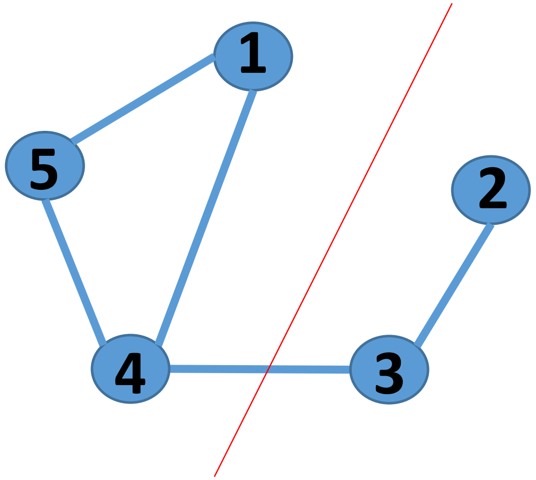
Complete Graph



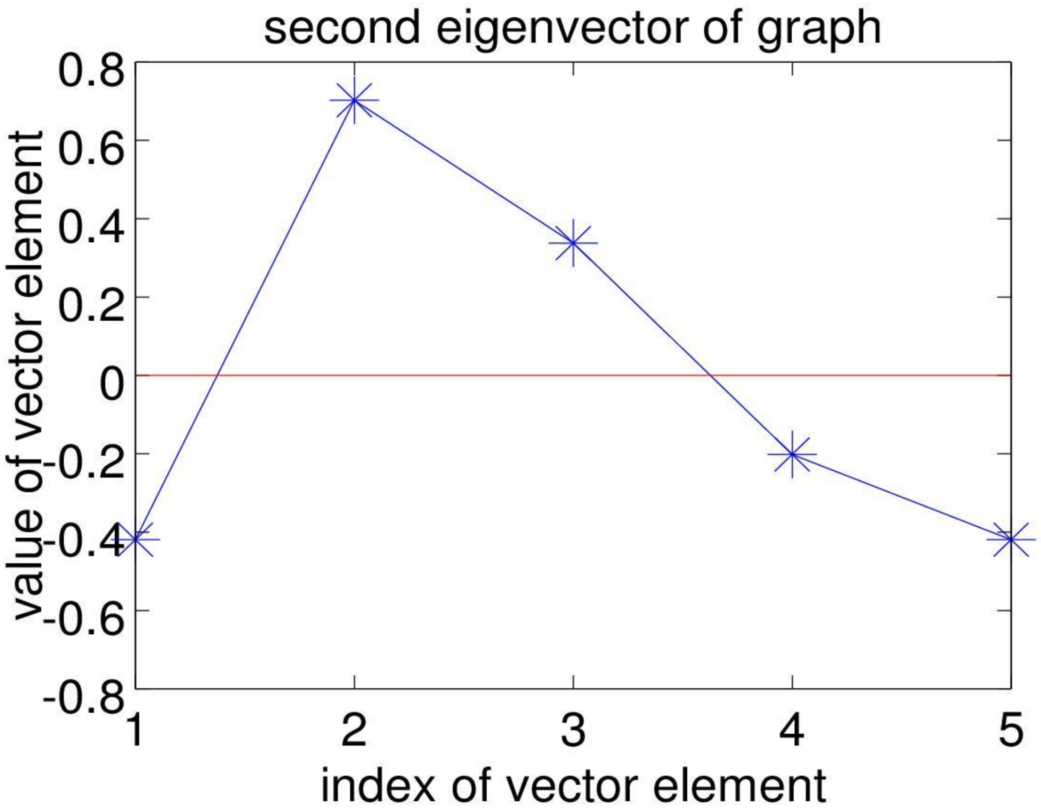
$$L = \begin{pmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & -1 & 4 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix}$$

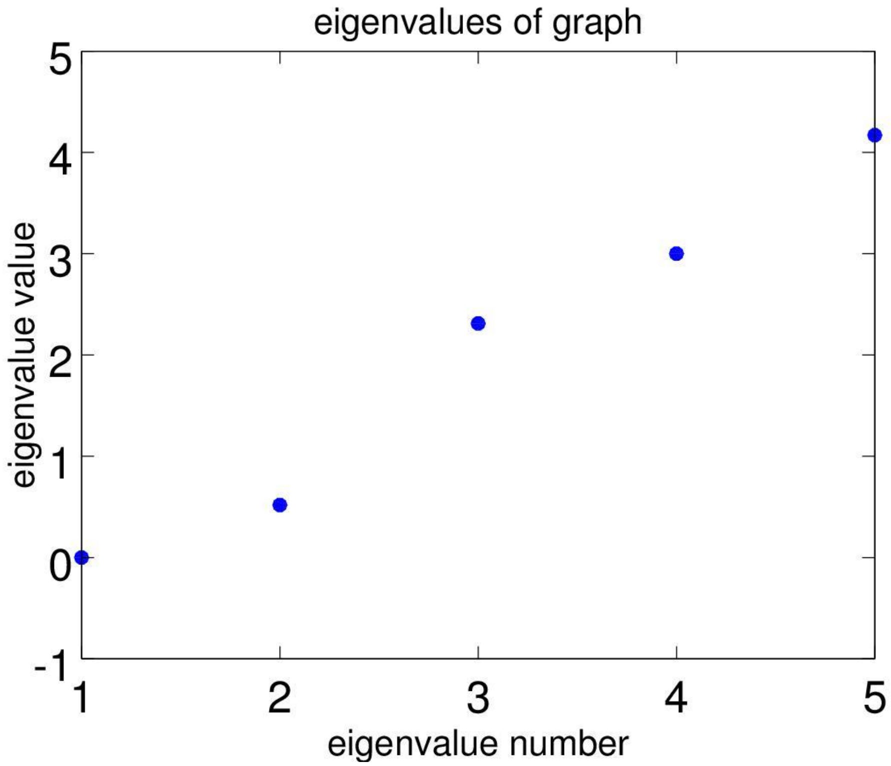


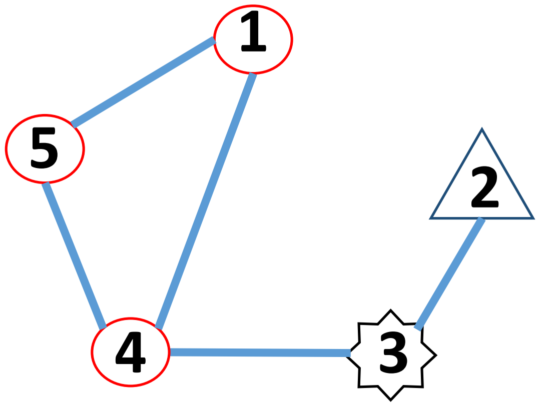




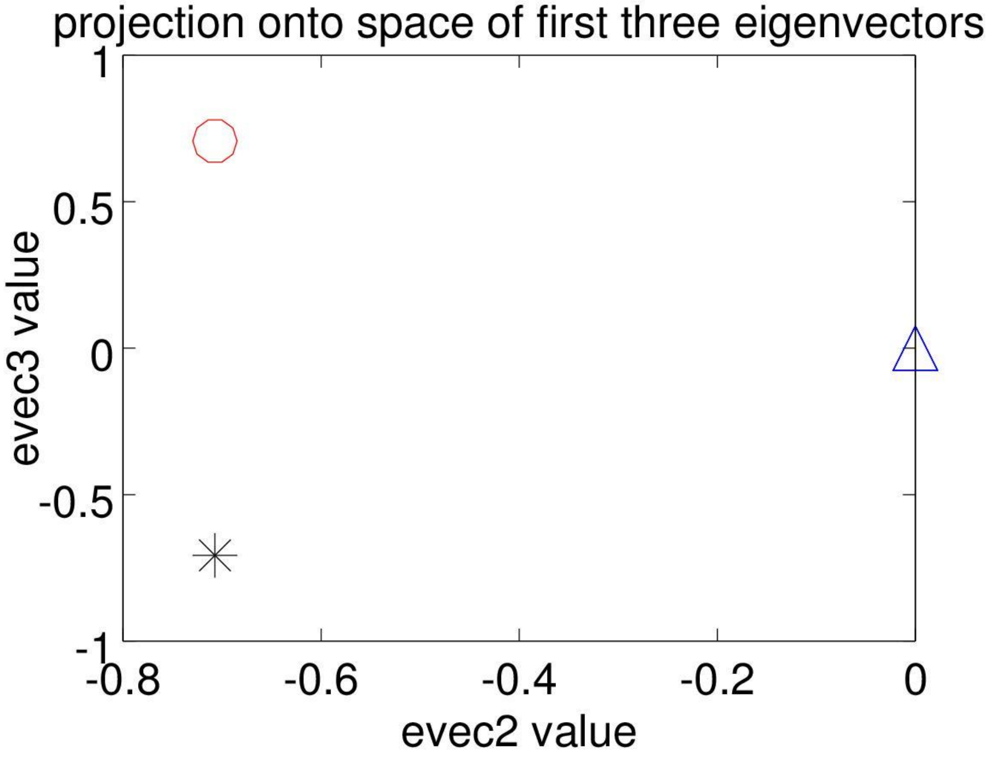
$$L = \begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$

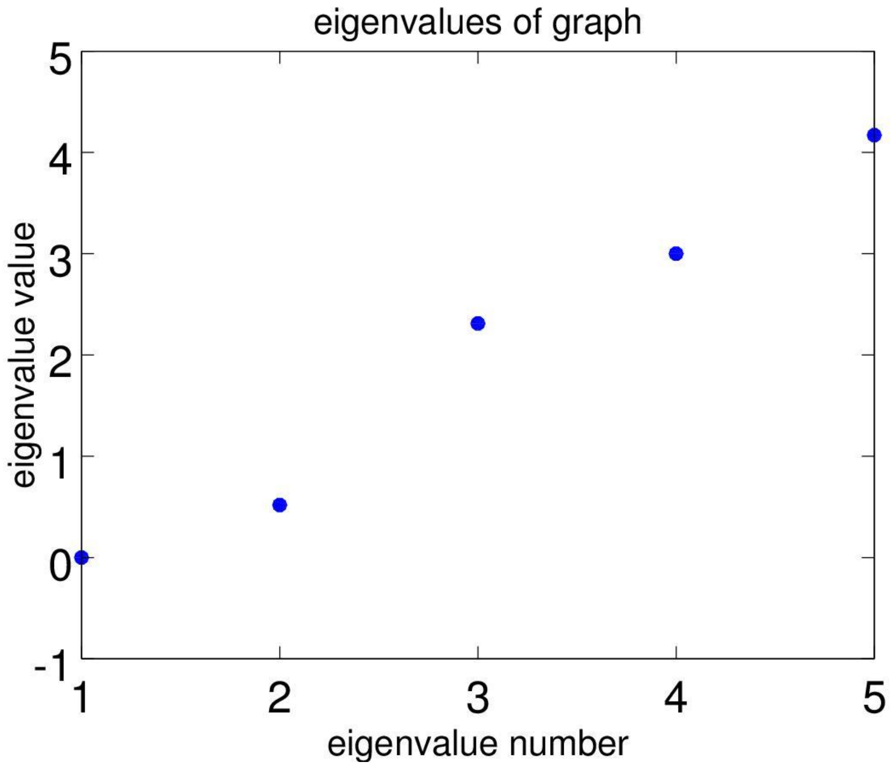




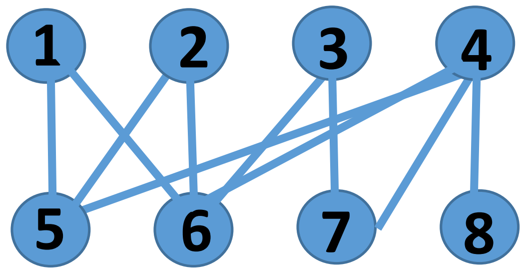


$$L = \begin{pmatrix} 2 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$

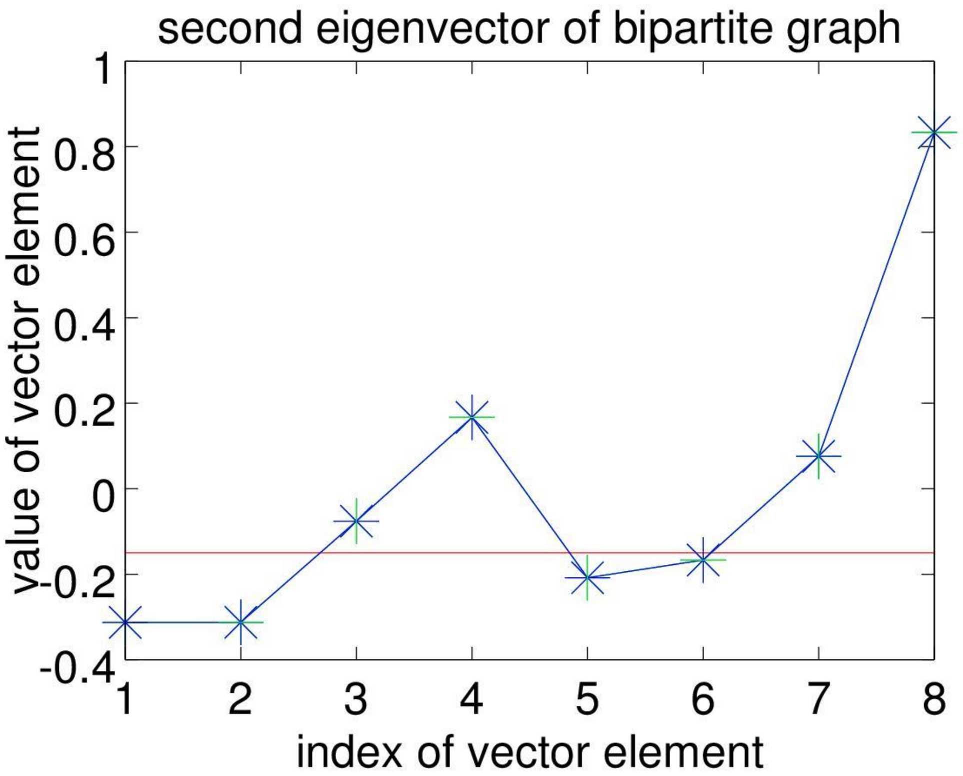


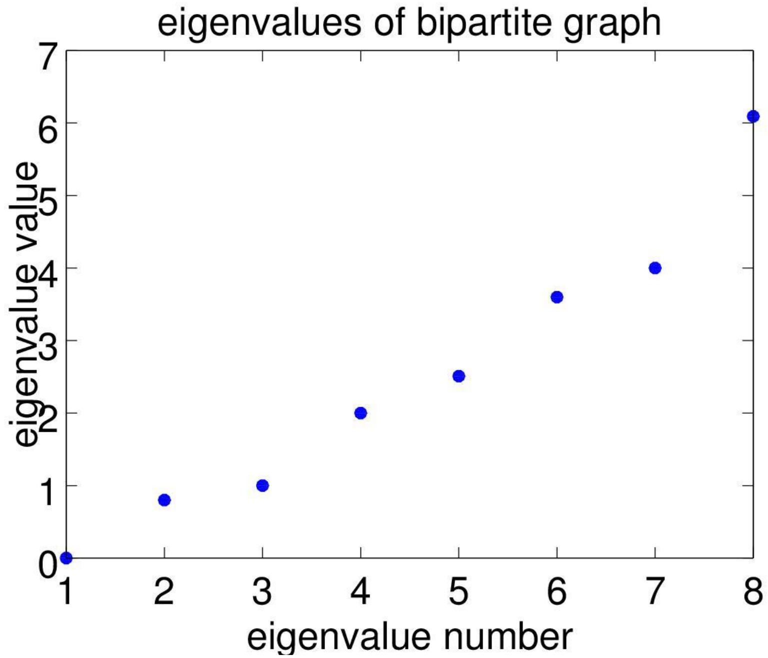


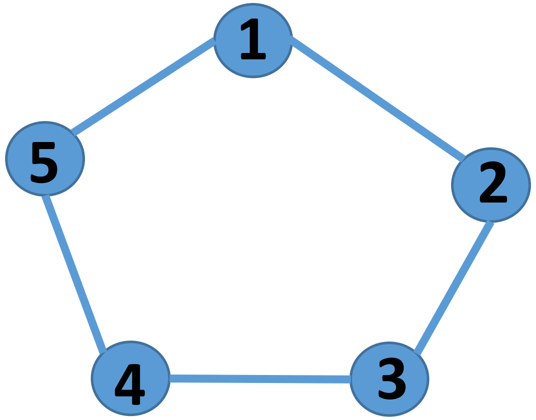
Bipartite Graph



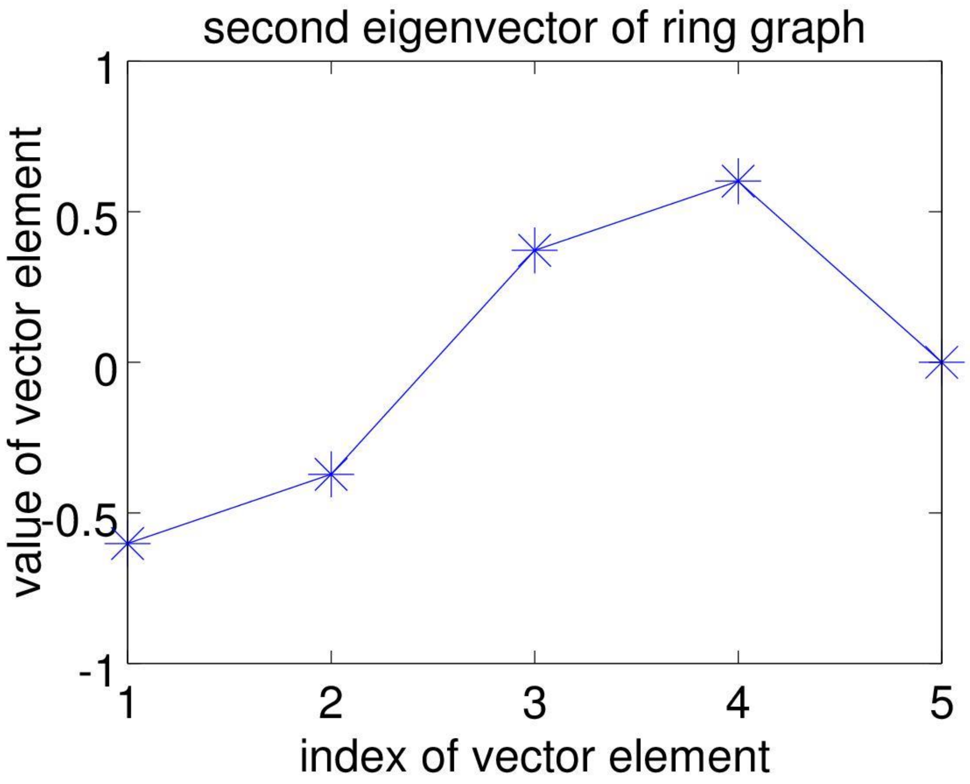
$$L = \begin{pmatrix} 2 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 2 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 4 & -1 & -1 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 0 & 4 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

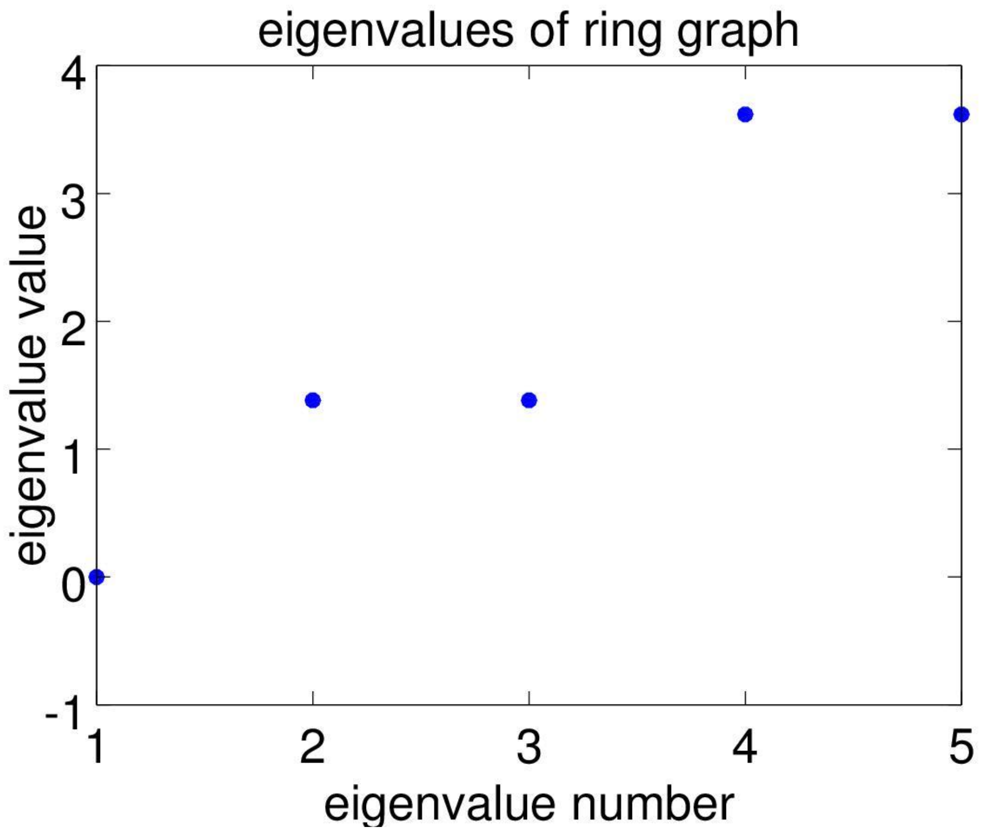




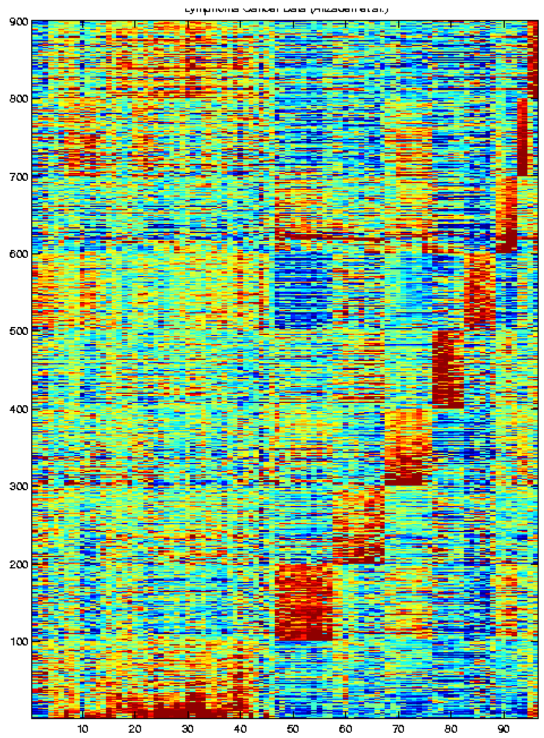


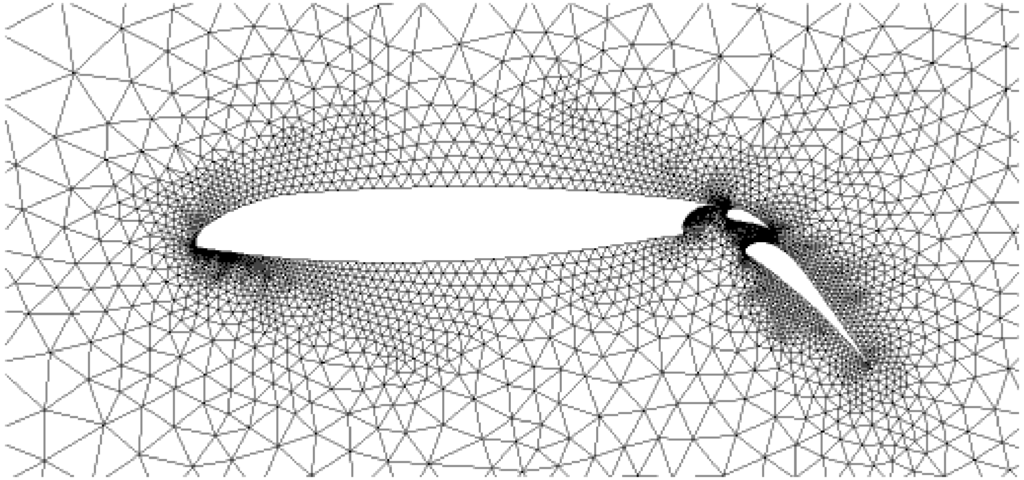
$$L = \begin{pmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$





Real-World Examples





Issues in Spectral Clustering

- similarity function
  - a reasonable choice is $s_{ij} = e^{-\|x_i - x_j\|/2\sigma^2}$ when the data points live in a Euclidean space, but it always depends on the domain

Issues in Spectral Clustering

- similarity function
  - a reasonable choice is $s_{ij} = e^{-\|x_i - x_j\|/2\sigma^2}$ when the data points live in a Euclidean space, but it always depends on the domain
- type of similarity graph
  - ε-neighborhood: difficulties arise when data is on "different scales" – which ε should we choose?
  - **$\boldsymbol{k}$-nearest neighbors:** points in low densities can be grouped with points in high densities
  - mutual k-nearest neighbors: tends to connect points with constant densities, but not points in densities that are different from each other

More issues

- fully connected graph
  - usually used with gaussian similarity function
  - need to pick σ wisely

More issues

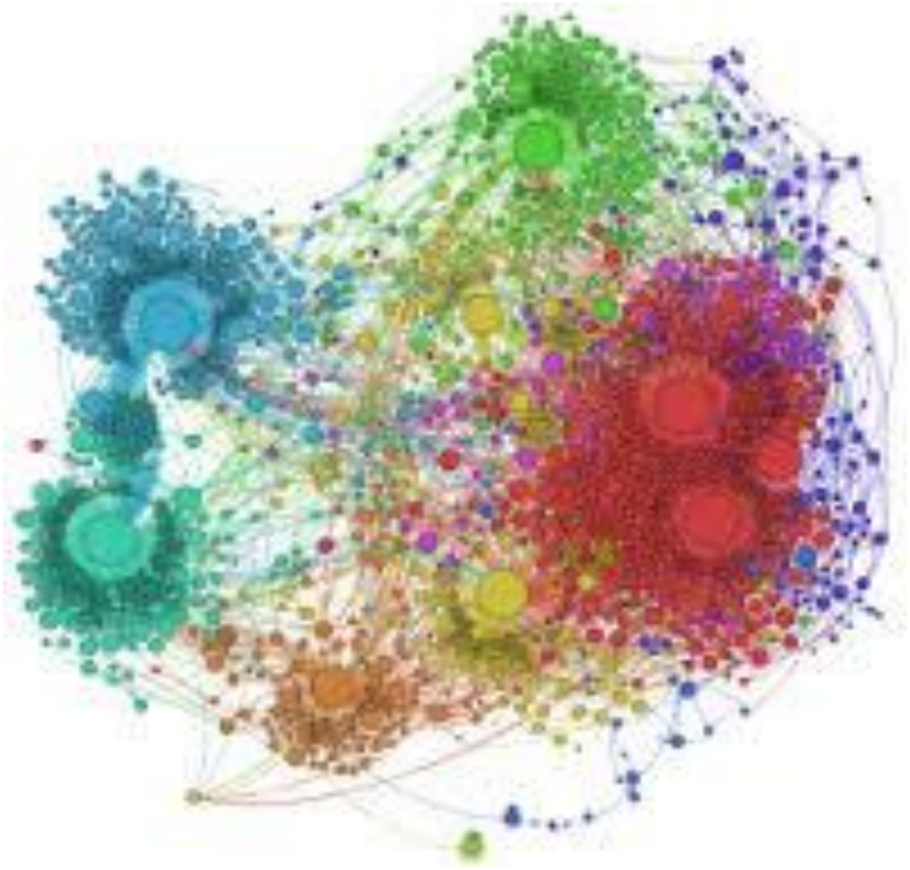
- fully connected graph
  - usually used with gaussian similarity function
  - need to pick σ wisely
- finding the eigenvectors
  - If we use the k-nearest neighbor graph or the ε-neighborhood graph, the Laplace matrix will be sparse, and we have efficient methods to compute the first k eigenvectors of sparse matrices
  - However, the speed of convergence of popular methods depends on the size of the eigengap

More issues

- fully connected graph
  - usually used with gaussian similarity function
  - need to pick σ wisely
- finding the eigenvectors
  - If we use the k-nearest neighbor graph or the ε-neighborhood graph, the Laplace matrix will be sparse, and we have efficient methods to compute the first k eigenvectors of sparse matrices
  - However, the speed of convergence of popular methods depends on the size of the eigengap
- choosing the number of clusters
  - “A variety of more or less successful methods have been devised for this problem”

Large-Scale Spectral Clustering

Main Challenge: Finding the first k eigenvectors can be prohibitively expensive



Large-Scale Spectral Clustering

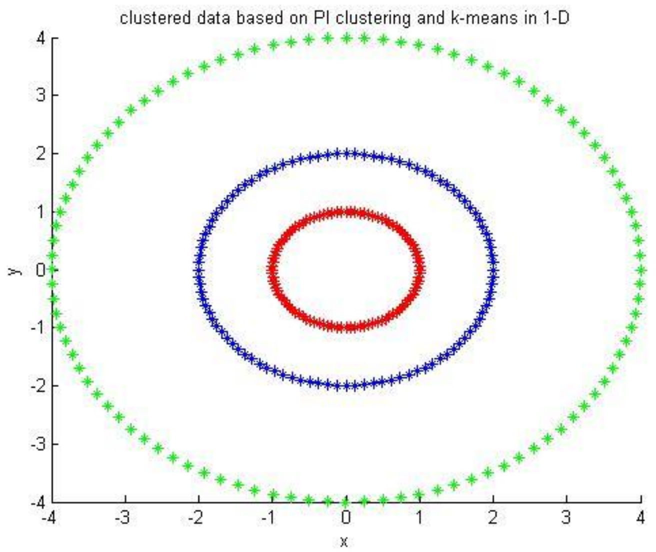
Main Challenge: Finding the first k eigenvectors can be prohibitively expensive

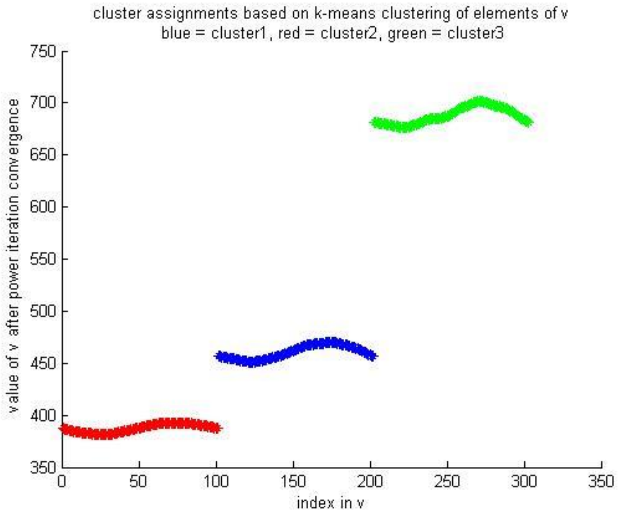
2010: *Power Iteration Clustering:* Key idea: use power iteration to find only ONE vector that will give partitions similar to those found by looking at first k eigenvectors

Power Iteration Clustering

Large-scale extension to Spectral Clustering

Uses power iteration on $\mathrm{I} - D^{-1}\mathrm{L} = D^{-1}W$ until convergence to a linear combination of the k smallest eigenvectors





Power Iteration Clustering

Large-scale extension to Spectral Clustering

Uses power iteration on $\mathrm{I} - D^{-1}\mathrm{L} = D^{-1}W$ until convergence to a linear combination of the k smallest eigenvectors

Applicable to large-scale text-document classification, works well for small values of k

- If $FF^T = W$ then $(D^{-1}W)v = D^{-1}FF^Tv$

Conclusion

Spectral Clustering has been a successful heuristic algorithm for partitioning the vertices of a graph

Relates linear algebraic theory to a discrete optimization problem

Has been extended to many application domains and to a large-scale setting

More Details

For an arbitrary k

We look for a solution of:

$$\min_{A_1 \dots A_k} Tr(H'LH) = \min_{A_1 \dots A_k} RatioCut(A_1, \dots, A_k)$$

With the constraints:

$$H'H = I, h_{ij} = \begin{cases} \dfrac{1}{\sqrt{|A_j|}} & if\, v_i \in A_j \\ 0\ otherwise \end{cases}$$

We relax the problem, then come back to unrelaxed version

The relaxed problem becomes:

$\min_{H \in \mathbb{R}^{n \times k}} Tr(H'LH)$ where $H'H = I$

Rayleigh-Ritz Theorem tells us that the solution to this problem is the H which contains the first k eigenvectors of L as columns

And we get back to discrete partitions of the graph by using k-means clustering on the rows of $H = U$

Note: In general it is known that efficient algorithms to approximate balanced graph cuts up to a constant factor do not exist

Normalized Graph Laplacians

- $L_{sym} = D^{-\frac{1}{2}} L D^{\frac{1}{2}}$
- $L_{rw} = D^{-1} L$
- Their eigenvalues and vectors are closely related to eachother and to the unnormalized graph Laplacian L

Normalized Laplacians - Properties

- The multiplicity of the eigenvalue 0 of both L_{sym} and L_{rw} is the number of connected components in the graph
- Graph is undirected with nonnegative weights
- The eigenspace of 0 for L_{rw} is spanned by the indicator vectors $\mathbb{I}_{A_i}$ for those components
- The eigenspace of 0 for L_{sym} is spanned by $D^{-1/2}\mathbb{I}_{A_i}$