Spectral Clustering

Part 3: The Normalized Laplacian

Ng Yen Kaow

More constraint for balance

- Further constraints can be added to the eigenvalue system
  - The next problem, Graph Partitioning, will use this strategy
  - However, the resultant eigenvalue system will no longer be standard

Graph Partitioning Problem

- Given edge weight matrix $W = (w_{ij})$ and vertex mass matrix M with diagonal elements (m_i), a 2-partitioning of an undirected graph $G = (V, E)$ is a partition of V into two groups S and $\bar{S}$ such that $\text{cut}(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} w_{ij}$ is minimized under the constraint that $\sum_{i \in S} m_i = \sum_{i \in \bar{S}} m_i$, or $\mathbf{1}^\top M x = 0$
  - Observe that if $m_i = 1$ for all i, then the condition $\sum_{i \in S} m_i = \sum_{i \in \bar{S}} m_i$ is the same as $|S| = |\bar{S}|$

Constrained optimization problem

- Minimize $x^\top Lx$ where $L = D' - W$

  subject to $x^\top M \in \{1, -1\}$ and $\mathbf{1}^\top Mx = 0$
  - $x_i \in \{1, -1\}$ and $\mathbf{1}^\top Mx = 0$ together enforce balance in the solution
  - However, problem is NP-hard
    - Recall that even the minimum bisection problem, where all edges and vertices have the same weight, is NP-hard

Relaxed Rayleigh quotient version

- Minimize $x^\top Lx$ where $L = D' - W$

  subject to $x^\top Mx = \sum_i m_i$ and $\mathbf{1}^\top Mx = 0$
  - $x_i \in \{1, -1\} \Rightarrow x^\top Mx = \sum_i m_i$ but not the other way around
  - Balance no longer enforced but that's the least of our worry for now because instead of the standard eigensystem

- Optimization must now be achieved through solving the generalized eigensystem

$$Lx = \lambda Mx$$

Relaxed Rayleigh quotient version

- Minimize $x^\top Lx$ where $L = D' - W$
  subject to $x^\top Mx = \sum_i m_i$ and $\mathbf{1}^\top Mx = 0$
- Optimize through $Lx = \lambda Mx$
- Since $\mathbf{1}$ fulfills condition for L and M, $\mu_k = \mathbf{1}$
  - However, eigenvectors in the solutions are not orthogonal but rather, M-orthogonal ($\mu_i M \mu_j = 0$ for $i \neq j$)
    - $\mathbf{1}^\top M \mu_{k-1} = 0$ is fulfilled
- Convert to a standard eigenvalue system $M^{-1/2} L M^{-1/2} x = \lambda x$ to compute

Convert to $M^{-1/2}LM^{-1/2}x = \lambda x$

- Minimize $x^\top Lx$ where $L = D' - W$

  subject to $x^\top Mx = \sum_i m_i$ and $\mathbf{1}^\top Mx = 0$

- Let $y = M^{1/2}x$, that is, $x = M^{-1/2}y$

$$x^\top Lx \Rightarrow y^\top M^{-1/2}LM^{-1/2}y$$

$$x^\top Mx = \sum_i m_i \Rightarrow y^\top y = \sum_i m_i$$

$$\mathbf{1}^\top Mx = 0 \Rightarrow \mathbf{1}^\top M^{1/2}y = 0$$

Hence equivalently

- Minimize $yM^{-1/2}LM^{-1/2}y$

  subject to $y^\top y = \sum_i m_i$ and $\mathbf{1}^\top M^{1/2}y = 0$

Convert to $M^{-1/2}LM^{-1/2}x = \lambda x$

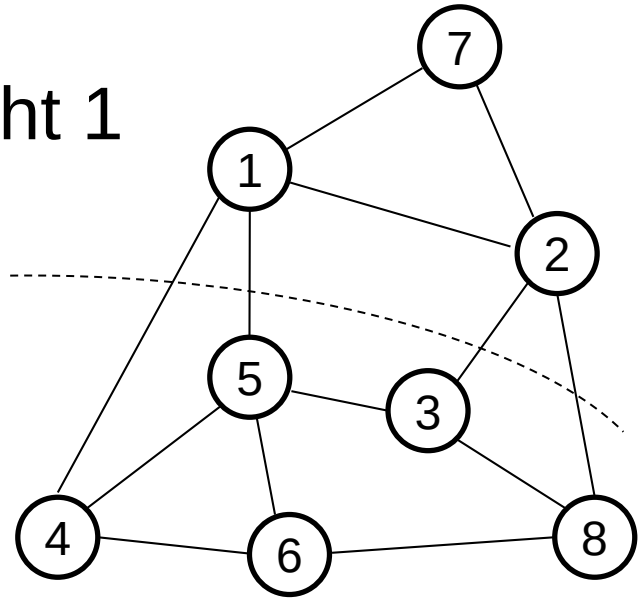
- Minimize $yM^{-1/2}LM^{-1/2}y$

  subject to $y^\top y = 1$ and $\mathbf{1}^\top M^{1/2}y = 0$
- As $\mathbf{1}$ is a eigenvector for $Lx = \lambda Mx$ with eigenvalue 0, $M^{1/2}\mathbf{1}$ is a eigenvector for this system with eigenvalue 0 (smallest)
  - Since eigenvectors of this system are orthogonal, $\left(M^{1/2}\mathbf{1}\right)\mu_{k-1} = 0$

    $\Rightarrow \mathbf{1}^\top M^{1/2}y = 0$ fulfilled
  - In fact the eigenvalues for this system are the same as those for $Lx = \lambda Mx$, even though the eigenvectors are different (related by $y = M^{1/2}x$)

Eigendecomposition

- Edges and vertices have weight 1



λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
5.9390	5.1420	4.6660	4.0	3.0500	1.8100	1.3940	0.0

μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_6	μ_6
0.5677	-0.1583	-0.4862	0.3536	0.2315	-0.2855	0.1766	0.3536
-0.4281	0.6222	-0.2059	0.3536	0.0622	0.2469	0.2690	0.3536
0.3517	0.1203	0.2984	-0.3536	0.5170	0.5007	-0.0694	0.3536
-0.0855	0.0612	0.6267	0.3536	0.1159	-0.4899	-0.3044	0.3536
-0.5514	-0.3549	-0.3566	-0.3536	0.3216	-0.1795	-0.2392	0.3536
0.2351	0.3822	-0.2014	-0.3536	-0.5589	-0.1183	-0.4263	0.3536
-0.0354	-0.1476	0.2596	-0.3536	-0.2798	-0.2029	0.7349	0.3536
-0.0540	-0.5251	0.0654	0.3536	-0.4096	0.5286	-0.1411	0.3536

Generalized eigenvalue system

- First use of generalized eigenvalue system for spectral clustering in

  Donath and Homan, "*Algorithms for partitioning of graphs and computer logic based on eigenvectors of connection matrices*", 1972, IBM Technical Disclosure Bulletin 15(3):938–944

- Also used in Normalized Cut
  - Which is currently almost synonymous with spectral clustering

Normalized Cut Problem

- Given weight matrix $W = (w_{ij})$ and weighted degree matrix $D' = (d_i)$, the normalized cut of an undirected graph $G = (V, E)$ is a partition of V into two groups S and $\bar{S}$ such that

$$\text{ncut}(S, \bar{S}) = \text{cut}(S, \bar{S}) \left(\frac{1}{\text{vol}(S)} + \frac{1}{\text{vol}(\bar{S})} \right)$$

is minimized, where $\text{vol}(S) = \sum_{i \in S} d_i$, that is, sum of all the weights of the edges adjacent to vertices in S, and $\text{cut}(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} w_{ij}$

Normalized Cut

- Represent a partition $S, \bar{S}$ of V with $x \in \mathbb{R}^n$, where

$$x_i = \begin{cases} \dfrac{1}{\mathrm{vol}(S)} & \text{if } i \in S \\ -\dfrac{1}{\mathrm{vol}(\bar{S})} & \text{if } i \in \bar{S} \end{cases}$$

As in Ratio Cut, $|x_i|$ **changes according to the solution**

1. $x^\top L x = \sum_{ij} w_{ij}\left(x_i - x_j\right)^2 = \left(\dfrac{1}{\mathrm{vol}(S)} + \dfrac{1}{\mathrm{vol}(\bar{S})}\right)^2 \sum_{ij} w_{ij}$

$$= \left(\frac{1}{\mathrm{vol}(S)} + \frac{1}{\mathrm{vol}(\bar{S})}\right)^2 \mathrm{cut}\left(S, \bar{S}\right)$$

2. $x^\top D' x = \sum_i d_i (x_i)^2 = \sum_{i \in S} \dfrac{d_i}{\mathrm{vol}(S)^2} + \sum_{i \in \bar{S}} \dfrac{d_i}{\mathrm{vol}(\bar{S})^2} = \dfrac{1}{\mathrm{vol}(S)} + \dfrac{1}{\mathrm{vol}(\bar{S})}$

$$1 + 2 \Rightarrow \frac{x^\top L x}{x^\top D' x} = \mathrm{cut}(S, \bar{S}) \left(\frac{1}{\mathrm{vol}(S)} + \frac{1}{\mathrm{vol}(\bar{S})}\right) = \mathrm{ncut}(S, \bar{S})$$

Constrained optimization problem

- Minimize $x^\top Lx$ where $L = D' - W$

  subject to $x_i \in \left\{\frac{1}{\text{vol}(S)}, -\frac{1}{\text{vol}(\bar{S})}\right\}$,

  $x^\top D'x = 1$, and

  $\mathbf{1}^\top D'x = 0$

- Problem is NP-hard
- Note:
  - $\mathbf{1}^\top D'x = \sum_{i\in S}\frac{d_i}{\text{vol}(S)} - \sum_{i\in \bar{S}}\frac{d_i}{\text{vol}(\bar{S})} = 1 - 1 = 0$
  - $\frac{1}{\text{vol}(S)}, -\frac{1}{\text{vol}(\bar{S})}$ are not the only possible choices
    - See https://arxiv.org/abs/1311.2492

Relaxed Rayleigh quotient version

- Minimize $x^\top Lx$

  subject to $x^\top D'x = 1$ and $\mathbf{1}^\top D'x = 0$

Through the same reasoning as in graph partitioning problem, equivalently solve

- Minimize $y(D')^{-1/2}L(D')^{-1/2}y$

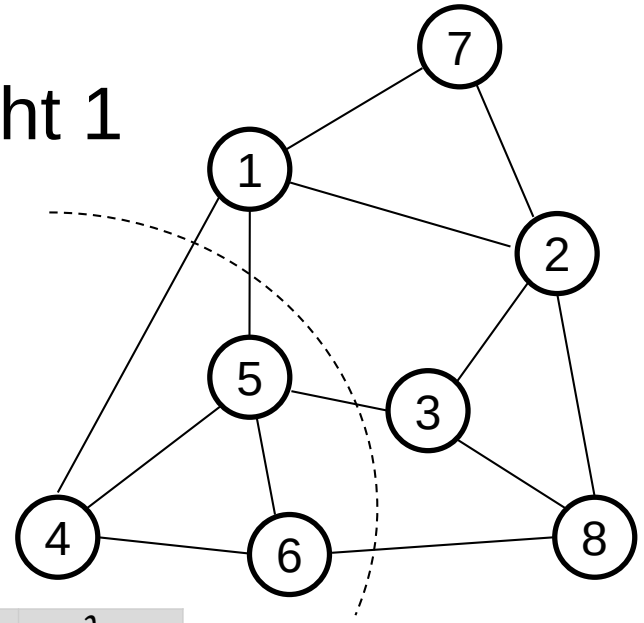
  subject to $y^\top y = 1$ and $\mathbf{1}^\top (D')^{1/2}y = 0$

  where $y = (D')^{1/2}x$

  - $(D')^{-1/2}L(D')^{-1/2}$ is now commonly known as the **normalized Laplacian**

Eigendecomposition

- Edges and vertices have weight 1



λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
1.6760	1.5100	1.42700	1.3100	0.9900	0.5880	0.4990	0.0

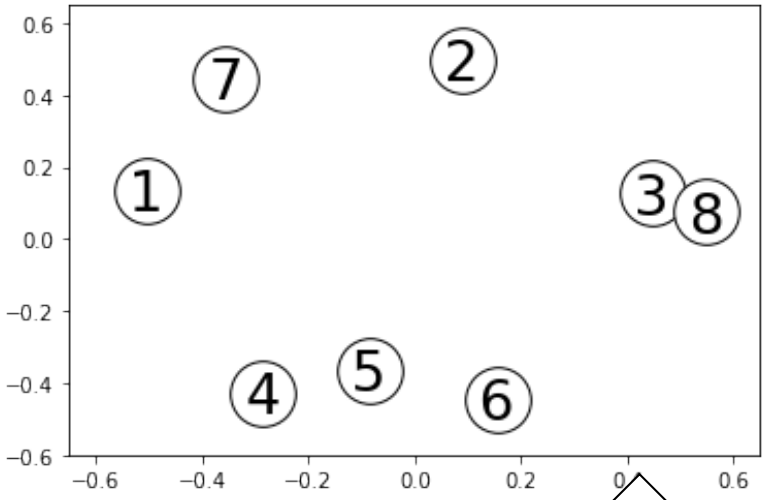
μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8
0.3485	0.0034	0.6240	-0.2451	-0.0704	-0.5023	0.1342	0.3922
-0.0304	0.6546	-0.3393	-0.2014	0.0768	0.0885	0.4973	0.3922
0.4129	-0.3896	-0.1906	-0.0484	-0.5545	0.4474	0.1265	0.3397
-0.2148	-0.2574	-0.4363	-0.5537	0.0989	-0.2859	-0.4286	0.3397
-0.4292	0.2801	0.1122	0.4236	-0.5021	-0.0836	-0.3638	0.3922
0.5058	0.1486	-0.0793	0.3598	0.4989	0.1541	-0.4454	0.3397
-0.1662	-0.4557	-0.2360	0.5096	0.2180	-0.3552	0.4457	0.2774
-0.4397	-0.2128	0.4406	-0.1475	0.3513	0.5487	0.0744	0.3397

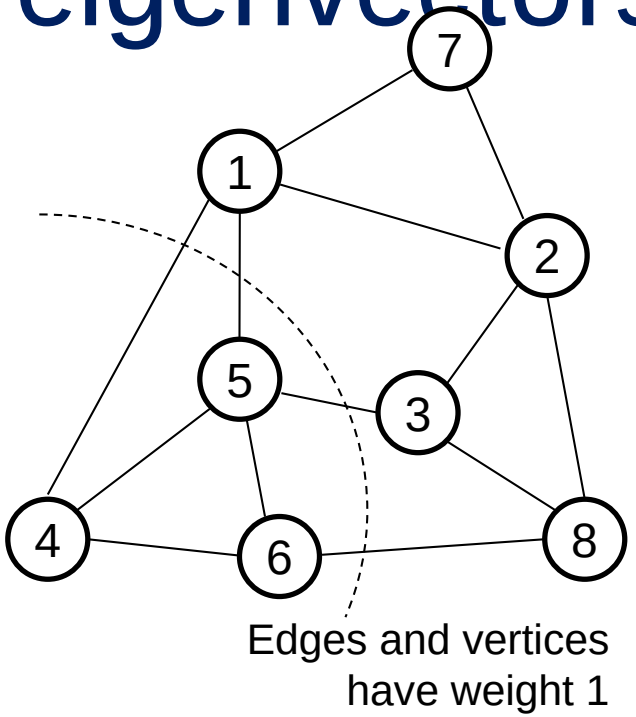
Shi and Malik (1997, 2000)

- Proposed the NP-hard ncut problem
  - Derived Laplacian for ncut
- Related ncut Laplacian to generalized eigenvalue system, resulting in the now ubiquitous **normalized Laplacian**
  - However, the first use of the generalized eigenvalue system for spectral clustering was in 1972
- Use Gaussian function $e^{-d^2/2\sigma^2}$ for weights
  - Previously used for min-cut (Wu and Leahy 1993)
  - Used for RatioCut later (Wang and Siskin 2003)
- Clustering with multiple eigenvectors (Shi and Malik 2000)

Clustering w/ multiple eigenvectors

- With normalized Laplacian





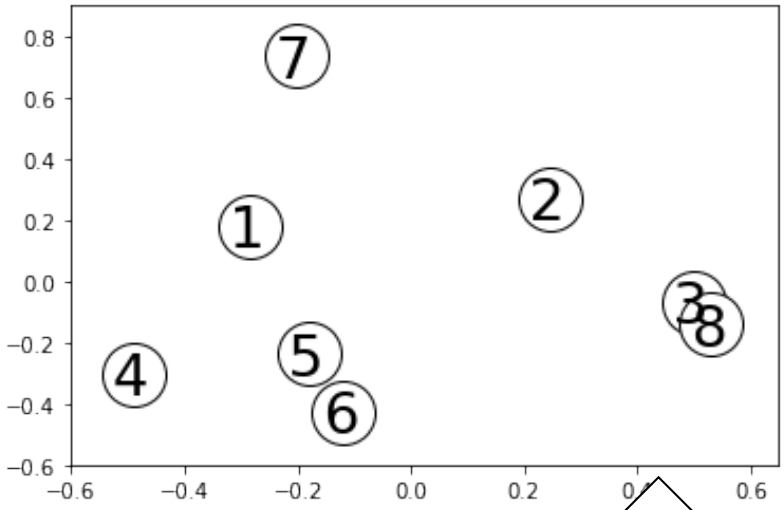
Edges and vertices have weight 1

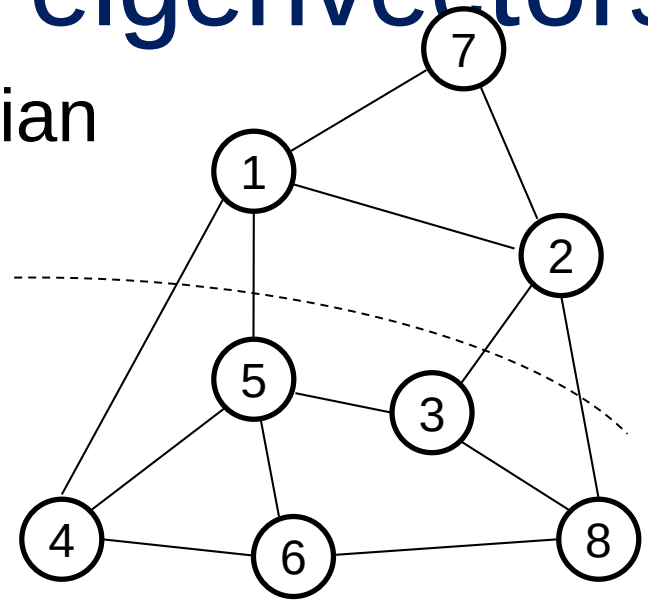
μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8
0.3485	0.0034	0.6240	-0.2451	-0.0704	-0.5023	0.1342	0.3922
-0.0304	0.6546	-0.3393	-0.2014	0.0768	0.0885	0.4973	0.3922
0.4129	-0.3896	-0.1906	-0.0484	-0.5545	0.4474	0.1265	0.3397
-0.2148	-0.2574	-0.4363	-0.5537	0.0989	-0.2859	-0.4286	0.3397
-0.4292	0.2801	0.1122	0.4236	-0.5021	-0.0836	-0.3638	0.3922
0.5058	0.1486	-0.0793	0.3598	0.4989	0.1541	-0.4454	0.3397
-0.1662	-0.4557	-0.2360	0.5096	0.2180	-0.3552	0.4457	0.2774
-0.4397	-0.2128	0.4406	-0.1475	0.3513	0.5487	0.0744	0.3397

Use the values from the top few eigenvectors for clustering (with, for example, *k*-means)

Clustering w/ multiple eigenvectors

- With graph partitioning Laplacian





Edges and vertices have weight 1

μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_6	μ_6
0.5677	-0.1583	-0.4862	0.3536	0.2315	-0.2855	0.1766	0.3536
-0.4281	0.6222	-0.2059	0.3536	0.0622	0.2469	0.2690	0.3536
0.3517	0.1203	0.2984	-0.3536	0.5170	0.5007	-0.0694	0.3536
-0.0855	0.0612	0.6267	0.3536	0.1159	-0.4899	-0.3044	0.3536
-0.5514	-0.3549	-0.3566	-0.3536	0.3216	-0.1795	-0.2392	0.3536
0.2351	0.3822	-0.2014	-0.3536	-0.5589	-0.1183	-0.4263	0.3536
-0.0354	-0.1476	0.2596	-0.3536	-0.2798	-0.2029	0.7349	0.3536
-0.0540	-0.5251	0.0654	0.3536	-0.4096	0.5286	-0.1411	0.3536

The resultant eigenvectors are less suitable for clustering

Theoretical justification

- Why does normalized Laplacian work
  - Probabilistic (random walk) justification by Meila and Shi (2000)

- Why does k-means on multiple eigenvectors work for the normalized Laplacian
  - Ng *et al.* (2001) show conditions for method to work

[Meila & Shi 2000] Random walk

- Let $x = x_1 x_2 \ldots x_n$ be a eigenvector used for clustering, then

$$x_i = \begin{cases} \frac{1}{\mathrm{vol}(S)} & \text{if } i \in S \\ -\frac{1}{\mathrm{vol}(\bar{S})} & \text{if } i \in \bar{S} \end{cases}$$

  - What guarantees x to be piecewise constant (constant for each of the cluster)?

- We first show that the x is related to a random walk of the graph

[Meila & Shi 2000] Random walk

- Let $P = D^{-1}L$.
  - A solution x for $Px = \lambda x$ is a solution for $Lx = \lambda Dx$ (with eigenvalues $1 - \lambda$), and vice versa

    Proof. $Lx = \lambda Dx \Rightarrow D^{-1}Lx = \lambda Ix \Rightarrow Px = \lambda x$

    $$Px = \lambda x \Rightarrow D(D^{-1}L)x = D\lambda x \Rightarrow Lx = \lambda Dx$$

    - However, P is not symmetric
      ⇒ doesn't decompose to orthogonal eigenbasis
    - On the other hand $D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$ is symmetric
      ⇒ chosen over P for spectral clustering
  - Each row in P sums to 1 (normalized)
    - P is a Markovian transition matrix
    - Solution of $Px = \lambda x$ is a random walk

[Meila & Shi 2000] Random walk

- $Px = \lambda x$ is a random walk
  - Consider $x' = Px$
    - Each x'_i is an average of the entries in x that are "adjacent to x_i" (compare to Graph Neural Networks)
    - $x' = x$ ($\lambda = 1$) $\Rightarrow$ system reached equilibrium (stationery point)
  - x being piecewise constant implies that there exist parts of the graph where the values propagate more easily within each part

[Ng 2001] Multiple eigenvectors

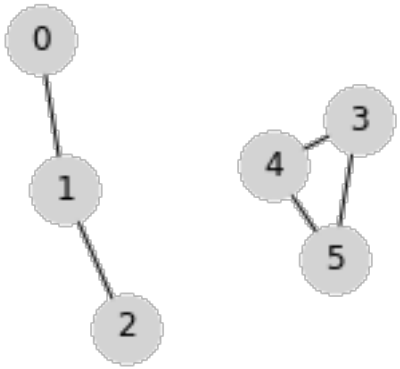
- For convenience use $L' = D'^{-\frac{1}{2}}(W)D'^{-\frac{1}{2}}$ instead of the normalized Laplacian for analysis
  - $L' = I - L$ ($L =$ normalized Laplacian)

    Proof. $L = D'^{-1/2}(D' - W)D'^{-1/2}$

    $= D'^{-1/2}(D')D'^{-\frac{1}{2}} - D'^{-\frac{1}{2}}(W)D'^{-\frac{1}{2}}$

    $= I - D'^{-\frac{1}{2}}(W)D'^{-\frac{1}{2}} = I - L'$
  - Results in the same eigenvectors but eigenvalues become $1 - \lambda_1, \ldots, 1 - \lambda_k$

[Ng 2001] Multiple eigenvectors



L^u

$$L' = \begin{bmatrix} 0 & .7 & 0 & 0 & 0 & 0 \\ .7 & 0 & .7 & 0 & \text{zeroes!} & 0 \\ 0 & .7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & .5 & .5 \\ 0 & \text{zeroes!} & 0 & .5 & 0 & .5 \\ 0 & 0 & 0 & .5 & .5 & 0 \end{bmatrix}$$

L^l

Matrix	Eigenvalues/vectors (decreasing order)
L^u	$\lambda_1^u = 1$ $\quad v_1^u = [.5 \quad .7 \quad .5]$ $\lambda_2^u = 0$ $\quad v_2^u = [.7 \quad 0 \quad -.7]$ $\lambda_3^u = -1$ $\quad v_3^u = [5 \quad -.7 \quad .5]$
L^l	$\lambda_1^l = 1$ $\quad v_1^l = [.6 \quad .6 \quad .6]$ $\lambda_2^l = -.5$ $\quad v_2^l = [0 \quad -.7 \quad -.7]$ $\lambda_3^l = -.5$ $\quad v_3^l = [-.8 \quad .4 \quad .4]$
L'	$\lambda_1 = 1$ $\quad v_1 = [0 \quad 0 \quad 0 \quad .6 \quad .6 \quad .6]$ $\lambda_2 = 1$ $\quad v_2 = [.5 \quad .7 \quad .5 \quad 0 \quad 0 \quad 0]$ $\lambda_3 = 0$ $\quad v_3 = [.7 \quad 0 \quad -.7 \quad 0 \quad 0 \quad 0]$ $\lambda_4 = -.5$ $\quad v_4 = [0 \quad 0 \quad 0 \quad 0 \quad -.7 \quad .7]$ $\lambda_5 = -.5$ $\quad v_5 = [0 \quad 0 \quad 0 \quad -.8 \quad .4 \quad .4]$ $\lambda_6 = -1$ $\quad v_6 = [.5 \quad -.7 \quad .5 \quad 0 \quad 0 \quad 0]$

- The eigenvalues/vectors of L' compose of the eigenvalues/vectors of the submatrices L^u and L^l
- The largest eigenvalue of L^u and L^l are both 1 for the ideal case

[Ng 2001] Multiple eigenvectors

- The largest eigenvalue of L^u and L^l is 1 for the ideal case

$$\lambda_1 = \lambda_2 = 1 \Rightarrow |\lambda_1 - \lambda_2| = 0$$

  - In non-ideal case, $\lambda_2 < \lambda_1$
  - The larger the eigenvalue, the more cohesive the cluster

- $|\lambda_2 - \lambda_3|$ is called a **eigengap**, or **spectral gap**
  - In all cases (ideal or non-ideal), $|\lambda_2 - \lambda_3| > 0$
  - The eigengap indicates degree of cluster separation

More clustering methods

- Based on the Fiedler vector
  - Sign cut or zero threshold cut
  - Median cut (ensures balance)
  - Sweep/criterion cut
    - Sort vertices by Fiedler vector values and cut at the lowest value of some cost function
  - Jump/gap cut
    - Sort vertices by Fiedler vector values and cut at the point of largest gap
- Based on multiple eigenvectors
  - Simultaneous k-way (Shi and Malik 2000)
  - k-means (Ng *et al.* 2001)