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 $\mathrm{cut}(S, \bar{S}) = \sum_{i\in S, j\in \bar{S}} w_{ij}$ 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 $x^\top L x$ where $L = D' - W$ subject to $x^\top M x = \sum_i m_i$ and $\mathbf{1}^\top M x = 0$
  - $x_i \in \{1, -1\} \Rightarrow x^\top M x = \sum_i m_i$ 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

$$Lx = \lambda M x$$

Relaxed Rayleigh quotient version

- Minimize $x^\top L x$ where $L = D' - W$ subject to $x^\top M x = \sum_i m_i$ and $\mathbf{1}^\top M x = 0$
- Optimize through $Lx = \lambda M x$
- 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 $yM^{-1/2}LM^{-1/2}y$

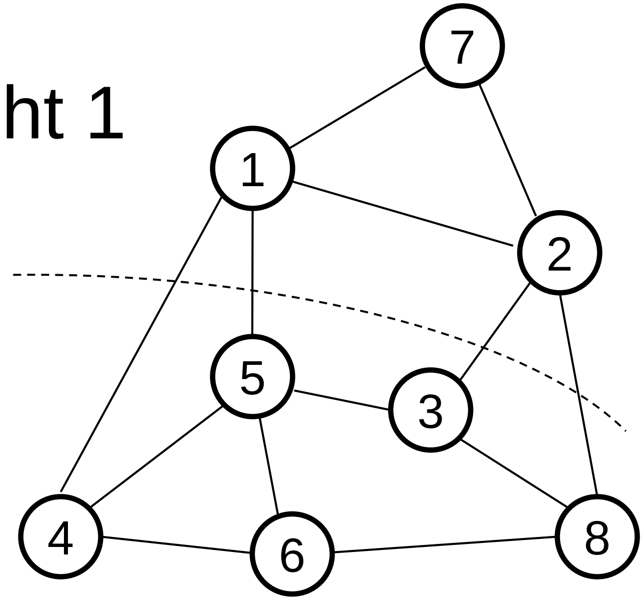
  subject to $y^\top y = \sum_i m_i$ and $\mathbf{1}^\top M^{1/2}y = 0$

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, $(M^{1/2}\mathbf{1})\mu_{k-1} = 0$
    $\Rightarrow \mathbf{1}^\top M^{1/2}y = 0$ 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

- Note that $M^{-1/2}LM^{-1/2}$ cannot be related to the incidence matrix as with the earlier graph Laplacian

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

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

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

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

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

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

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

Constrained optimization problem

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

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

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

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

Relaxed Rayleigh quotient version

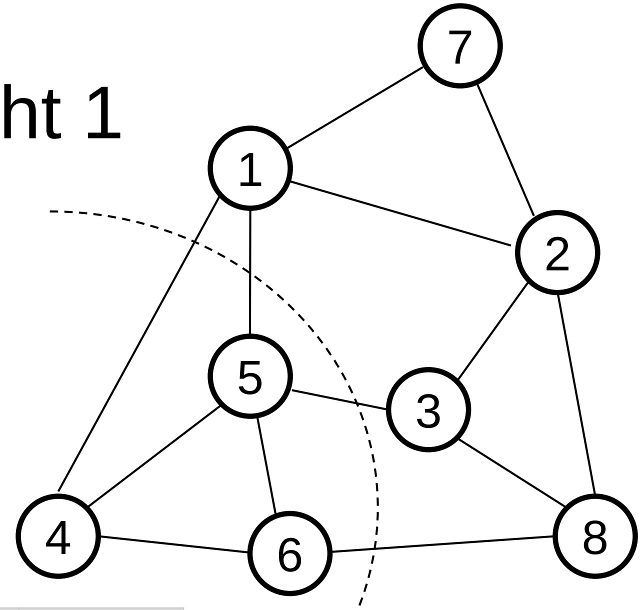
- Minimize $x^\top L x$
  subject to $x^\top D' x = 1$ and $\mathbf{1}^\top D' x = 0$

Through the same reasoning as in graph partitioning problem, equivalently solve the generalized eigensystem $Lx = \lambda D' x$

- 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 called the **normalized Laplacian** (due to its relation to $D^{-1}W$… later)

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

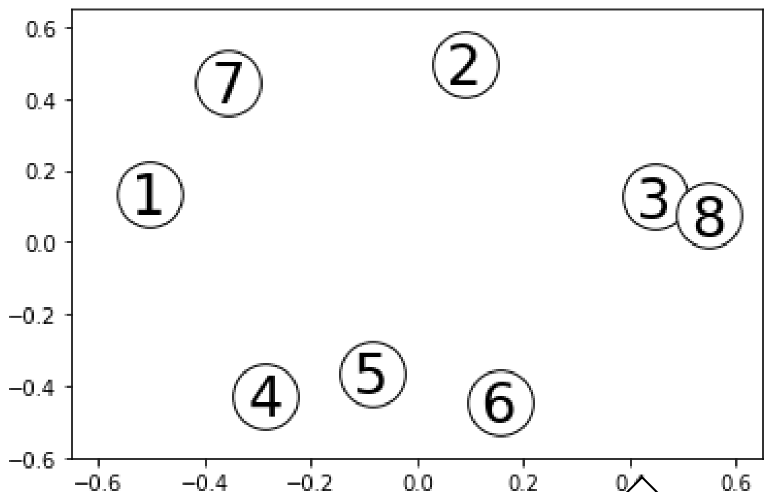
The limiting distribution of the normalized Laplacian is not $f(v) = \text{const}$

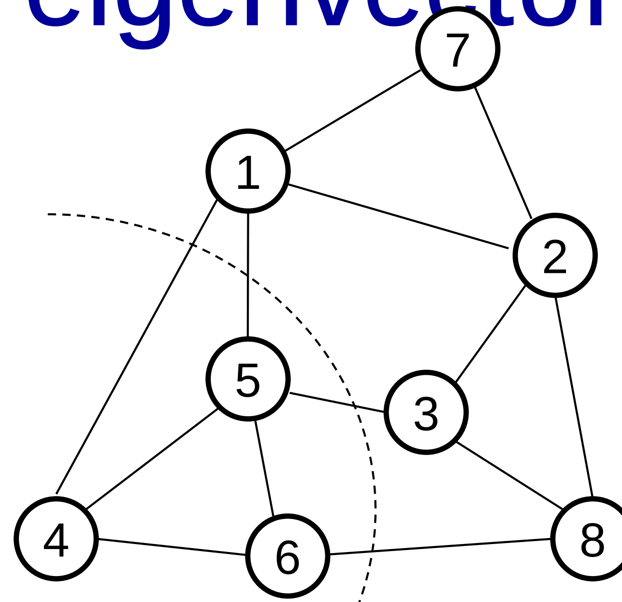
Shi and Malik (1997, 2000)

- Proposed the NP-hard ncut problem
- Related ncut 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





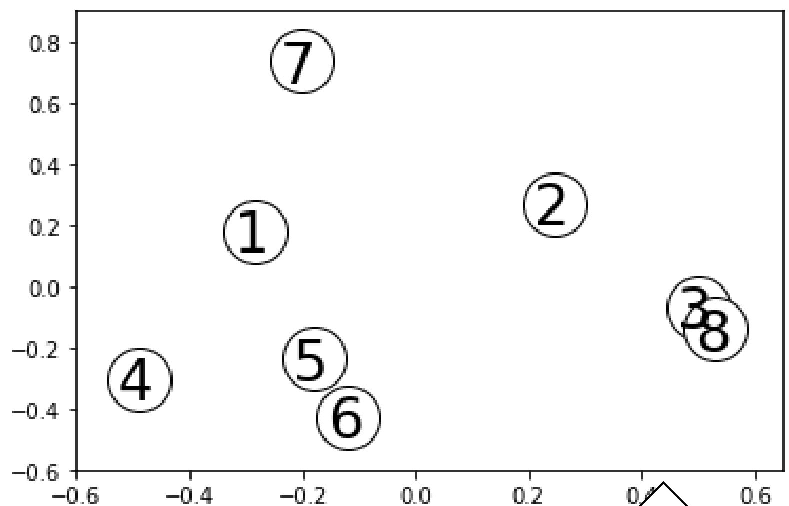
Edges and vertices have weight 1

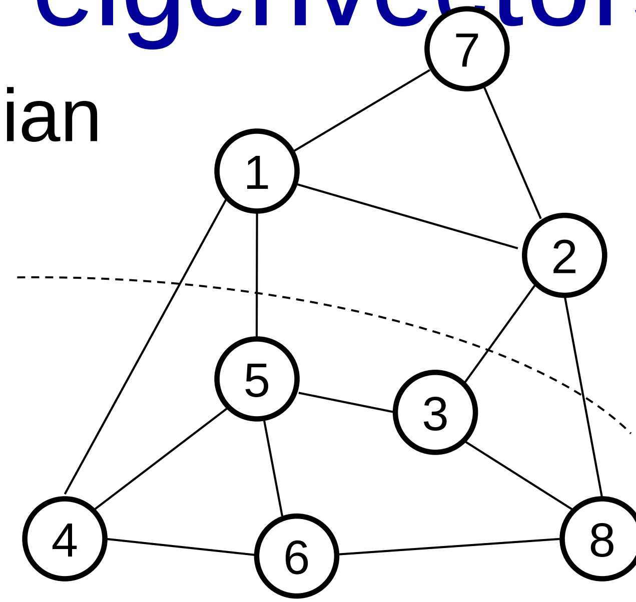
μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8
0.3485	0.0034	0.6240	-0.2451	-0.0704	-0.5023	0.1342	0.3922
-0.0304	0.6546	-0.3393	-0.2014	0.0768	0.0885	0.4973	0.3922
0.4129	-0.3896	-0.1906	-0.0484	-0.5545	0.4474	0.1265	0.3397
-0.2148	-0.2574	-0.4363	-0.5537	0.0989	-0.2859	-0.4286	0.3397
-0.4292	0.2801	0.1122	0.4236	-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

Single/multiple eigenvectors use

- Historical use based on 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
- After Shi and Malik, multiple eigenvectors
  - Simultaneous k-way (Shi and Malik 2000)
  - k-means (Ng, Jordan and Weiss 2001)

Theoretical justification

- How should we view the normalized Laplacian
  - Since normalized Laplacian cannot be related to the incidence matrix, it requires a new characterization

  ⇒ Random walk characterization (Meilă and Shi 2000)
- Arguments based on minimizing divergence and objective functions justify only the use of only one eigenvector (not multiple eigenvectors)
  - Furthermore, the argument from minimizing divergence is no longer valid for the normalized Laplacian

  ⇒ (Weiss 1999), (Meilă and Shi 2000), (Ng, Jordan and Weiss 2001) successively gives justification for the use of the eigenvectors

Random walk characterization

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

    Proof.

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

$$Px = (I - \lambda)x \Rightarrow D^{-1}Wx = (I - \lambda)x$$
$$(I - D^{-1}W)x = \lambda x$$
$$(D - W)x = D\lambda x$$
$$Lx = D\lambda x$$

Random walk characterization

- Let $P = D^{-1}W$ (where $L = D - W$)
  - A solution x for $Px = \lambda x$ is a solution for the generalized eigensystem $Lx = \lambda Dx$ (with eigenvalues $1 - \lambda$), and vice versa
    - **The normalized Laplacian $D^{-1/2}LD^{-1/2}$ computes the solutions to $Px = \lambda x$ for the normalized matrix P**
  - However, P is not symmetric
    - Doesn't decompose to orthogonal eigenbasis
  - On the other hand $D^{-1/2}LD^{-1/2}$ is symmetric
    - Chosen over P for spectral clustering

Random walk characterization

- Each row in P sums to 1 (normalized)
  - P **is a Markovian transition matrix**
- To start a walk from v_1, let $x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$, then $P^l x$ is the probability distribution after l steps from v_1
- x_i for neighboring vertices will become more similar $\Rightarrow$ gradients decrease
- Parts of the graph will even out more quickly

Random walk characterization

- Each row in P sums to 1 (normalized)
  - P **is a Markovian transition matrix**
- A limiting/stable/stationary state for a random walk P is a distribution x^* where $Px^* = x^*$
  - By definition x^* is a eigenvector of P with $\lambda = 1$

Furthermore, x^* is everywhere constant if P is

- A transition matrix for a regular graph
  By symmetry of the graph, a random walk from any vertex is equally likely to be at any other vertex in the limit
- A Laplacian $L = MM^\top$ for incidence matrix M
  First note that x^* minimizes $x^\top Lx$. On the other hand we know that $x^\top Lx = \sum_v f(v)\Delta f(v)$. Since $\Delta f(v) = 0$ for the everywhere constant x', we have $x'^\top Lx' = 0$, its minimum. Hence $x^* = x'$

Why use multiple eigenvectors

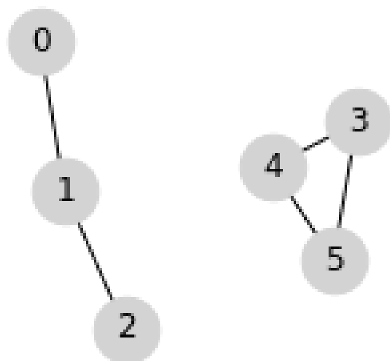
- For convenience use $L' = {D'}^{-1/2}(W){D'}^{-1/2}$ instead of the normalized Laplacian for analysis
  - $L' = I - L$ ($L =$ normalized Laplacian)

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

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

    $= I - {D'}^{-1/2}(W){D'}^{-1/2} = I - L'$
  - Results in the same eigenvectors but eigenvalues become $1 - \lambda_1, \dots, 1 - \lambda_k$
    - Since eigenvalues of L has range in [0,2], eigenvalues of L' has range in [-1,1]

Why use 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	$\lambda_1^u = 1$ $\lambda_2^u = 0$ $\lambda_3^u = -1$	$v_1^u = [.5 \quad .7 \quad .5]$ $v_2^u = [.7 \quad 0 \quad -.7]$ $v_3^u = [.5 \quad -.7 \quad .5]$
L'_l	$\lambda_1^l = 1$ $\lambda_2^l = -.5$ $\lambda_3^l = -.5$	$v_1^l = [.6 \quad .6 \quad .6]$ $v_2^l = [0 \quad -.7 \quad -.7]$ $v_3^l = [-.8 \quad .4 \quad .4]$
L'	$\lambda_1 = 1$ $\lambda_2 = 1$ $\lambda_3 = 0$ $\lambda_4 = -.5$ $\lambda_5 = -.5$ $\lambda_6 = -1$	$v_1 = [0 \quad 0 \quad 0 \quad .6 \quad .6 \quad .6]$ $v_2 = [.5 \quad .7 \quad .5 \quad 0 \quad 0 \quad 0]$ $v_3 = [.7 \quad 0 \quad -.7 \quad 0 \quad 0 \quad 0]$ $v_4 = [0 \quad 0 \quad 0 \quad 0 \quad -.7 \quad .7]$ $v_5 = [0 \quad 0 \quad 0 \quad -.8 \quad .4 \quad .4]$ $v_6 = [.5 \quad -.7 \quad .5 \quad 0 \quad 0 \quad 0]$

- The eigenvalues/vectors of L' compose of the eigenvalues/vectors of the submatrices L'_u and L'_l, with unconnected vertices set to 0
- The largest eigenvalue of L'_u and L'_l are both 1 for the ideal case

Why use multiple eigenvectors

- The largest eigenvalue of L'_u and L'_l is 1 for the ideal (disconnected) case
$$\lambda_1 = \lambda_2 = 1 \Rightarrow |\lambda_1 - \lambda_2| = 0$$
  - In non-ideal case, $\lambda_2 < \lambda_1$
  - The larger the eigenvalue (for L'), the more cohesive the cluster (this is opposite for L)
- $|\lambda_k - \lambda_{k+1}|$ is called **eigengap** or **spectral gap**
  - Large $|\lambda_k - \lambda_{k+1}|$ implies higher cohesion in the clusters given by μ_k than those by μ_{k+1}
  - Evaluate whether to use a eigenvector in clustering by its eigengap from the previous

Reconciliation with divergence

- No direct relation between the normalized L' (or L) with divergence

  $\Rightarrow$ Cannot assume that values in the eigenvector of largest eigenvalue μ_1 (for L') is constant

- However, from Fourier analysis, it remains the case that values in the eigenvectors of smaller eigenvalues will vary more rapidly across the graph (Shuman *et al.* 2000)

Reconciliation with divergence

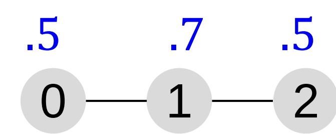
- Values in eigenvectors of smaller eigenvalues vary more rapidly across the graph

Example:

- At the largest eigenvalue (for L')
  - Not exactly but still, almost constant everywhere
  - Coincides with the lowest divergence case
- At larger eigenvalues (for L')
  - Smaller variation across connected vertices
  - Coincides with lower divergence case
- At small eigenvalues (for L')
  - Large variation across connected vertices
  - Coincides with higher divergence case

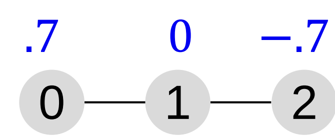
L'_u from earlier example

$\lambda_1^u = 1$



not constant!

$\lambda_2^u = 0$



$\lambda_3^u = -1$