MAT 280: Harmonic Analysis on Graphs & Networks
Lecture 8: Graph Laplacian Eigenfunctions II: Spectral Clustering

Naoki Saito

Department of Mathematics
University of California, Davis

October 17, 2019

Outline

1. Spectral Clustering

2. Spectral Clustering via Graph Cut Viewpoint

Outline

GL Eigenfunctions for L_{rw} and L_{sym}

Recall that we have three different versions of graph Laplacians:

$$L(G) := D - A \qquad \textit{Unnormalized}$$

$$L_{\mathrm{rw}}(G) := I_n - D^{-1}A = I_n - P = D^{-1}L \qquad \textit{Normalized}$$

$$L_{\mathrm{sym}}(G) := I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} \qquad \textit{Symmetrically-Normalized}$$

Proposition (Properties of L_{rw} and L_{sym})

(a) $(\lambda, \boldsymbol{\phi})$ *is an eigenpair of* L_{rw} *iff* $(\lambda, D^{1/2}\boldsymbol{\phi})$ *is an eigenpair of* L_{sym}. *In particular,* $(0, \mathbf{1}_n)$ *for* $L_{\mathrm{rw}} \Longleftrightarrow (0, D^{1/2}\mathbf{1}_n)$ *of* L_{sym}.

(b) $(\lambda, \boldsymbol{\phi})$ *is an eigenpair of* L_{rw} *iff* $(\lambda, \boldsymbol{\phi})$ *solves the generalized eigenproblem:* $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.

(c) *Both* L_{rw} *and* L_{sym} *are positive semi-definite and* n *nonnegative real-valued eigenvalues.*

GL Eigenfunctions for L_{rw} and L_{sym}

Recall that we have three different versions of graph Laplacians:

$$L(G) := D - A \qquad \textit{Unnormalized}$$

$$L_{\mathrm{rw}}(G) := I_n - D^{-1}A = I_n - P = D^{-1}L \qquad \textit{Normalized}$$

$$L_{\mathrm{sym}}(G) := I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} \qquad \textit{Symmetrically-Normalized}$$

Proposition (Properties of L_{rw} and L_{sym})

(a) $(\lambda, \boldsymbol{\phi})$ *is an eigenpair of* L_{rw} *iff* $(\lambda, D^{1/2}\boldsymbol{\phi})$ *is an eigenpair of* L_{sym}. *In particular,* $(0, \mathbf{1}_n)$ *for* $L_{\mathrm{rw}} \iff (0, D^{1/2}\mathbf{1}_n)$ *of* L_{sym}.

(b) $(\lambda, \boldsymbol{\phi})$ *is an eigenpair of* L_{rw} *iff* $(\lambda, \boldsymbol{\phi})$ *solves the generalized eigenproblem:* $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.

(c) *Both* L_{rw} *and* L_{sym} *are positive semi-definite and* n *nonnegative real-valued eigenvalues.*

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\Phi := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each *row* of Φ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth *row* vector of Φ.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the *k-means* algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\boldsymbol{\Phi} := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each *row* of $\boldsymbol{\Phi}$ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth *row* vector of $\boldsymbol{\Phi}$.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the *k-means* algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\Phi := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each *row* of Φ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth *row* vector of Φ.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the *k-means* algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D \boldsymbol{\phi}$.)
4. Let $\boldsymbol{\Phi} := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each *row* of $\boldsymbol{\Phi}$ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth *row* vector of $\boldsymbol{\Phi}$.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the *k-means* algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\boldsymbol{\Phi} := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each *row* of $\boldsymbol{\Phi}$ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth *row* vector of $\boldsymbol{\Phi}$.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the *k-means* algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\boldsymbol{\Phi} := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each *row* of $\boldsymbol{\Phi}$ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth *row* vector of $\boldsymbol{\Phi}$.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the *k-means* algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Spectral Clustering Algorithm for a Weighted Graph G

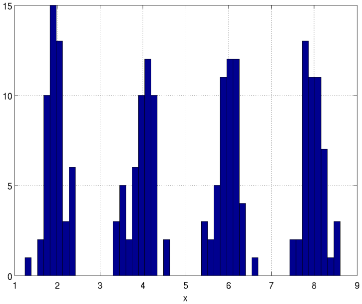
1. Construct a weighted adjacency matrix A.
2. Choose a graph Laplacian to use: L, L_{rw}, or L_{sym}.
3. Compute the first k eigenvectors $\boldsymbol{\phi}_0, \ldots, \boldsymbol{\phi}_{k-1}$. (Note in the case of L_{rw}, one needs to solve the generalized eigenproblem $L\boldsymbol{\phi} = \lambda D\boldsymbol{\phi}$.)
4. Let $\boldsymbol{\Phi} := [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}] \in \mathbb{R}^{n \times k}$. (Note in the case of L_{sym}, each *row* of $\boldsymbol{\Phi}$ is further normalized to have norm 1.)
5. Let $\boldsymbol{y}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times k}$ be the jth *row* vector of $\boldsymbol{\Phi}$.
6. Cluster these n vectors $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\} \subset \mathbb{R}^k$ representing $V(G)$ with the *k-means* algorithm into clusters $C_1, \ldots, C_k$.
7. Label each vertex with its cluster number.

Simple Examples for Spectral Clustering

- The following example was taken from Von Luxburg's tutorial paper with some modification.
- The dataset consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j,\ j = 1,2,3,4$, and $\sigma = 0.25$.
- These 200 points in $\mathbb{R}$ are the vertices.

Simple Examples for Spectral Clustering

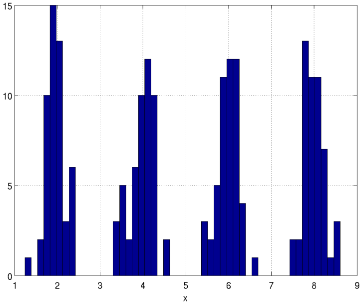
- The following example was taken from Von Luxburg's tutorial paper with some modification.
- The dataset consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j$, $j = 1, 2, 3, 4$, and $\sigma = 0.25$.



- These 200 points in $\mathbb{R}$ are the vertices.

Simple Examples for Spectral Clustering

- The following example was taken from Von Luxburg's tutorial paper with some modification.
- The dataset consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j,\ j = 1, 2, 3, 4$, and $\sigma = 0.25$.



- These 200 points in $\mathbb{R}$ are the vertices.

Simple Examples for Spectral Clustering ...

- A *complete* graph K_{200} was generated with the edge weight by $a_{ij} = \exp(-|x_i - x_j|^2/2\epsilon^2)$ where $\epsilon = 1$ was used throughout the experiments.
- Applied the spectral clustering algorithms.
- Note that we will discuss more about *how to construct a graph from given datasets* in the future lectures. The above strategy is used for simplicity.

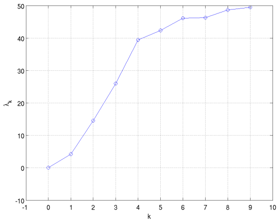
Simple Examples for Spectral Clustering ...

- A *complete* graph K_{200} was generated with the edge weight by $a_{ij} = \exp(-|x_i - x_j|^2/2\epsilon^2)$ where $\epsilon = 1$ was used throughout the experiments.
- Applied the spectral clustering algorithms.
- Note that we will discuss more about *how to construct a graph from given datasets* in the future lectures. The above strategy is used for simplicity.

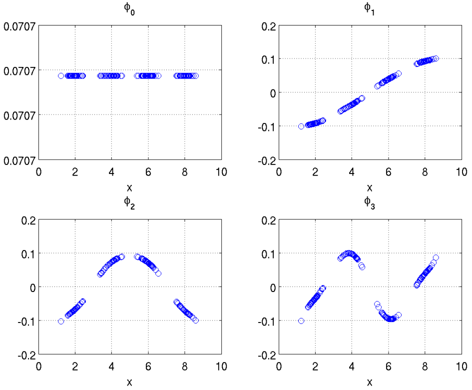
Simple Examples for Spectral Clustering . . .

- A *complete* graph K_{200} was generated with the edge weight by $a_{ij} = \exp(-|x_i - x_j|^2/2\epsilon^2)$ where $\epsilon = 1$ was used throughout the experiments.
- Applied the spectral clustering algorithms.
- Note that we will discuss more about *how to construct a graph from given datasets* in the future lectures. The above strategy is used for simplicity.

Using L

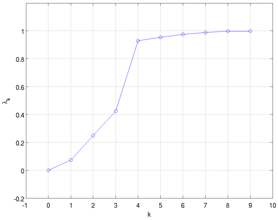


(a) λ_k

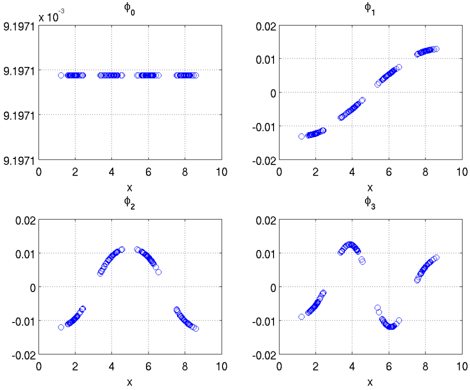


(b) $\boldsymbol{\phi}_k$

Using L_{rw}

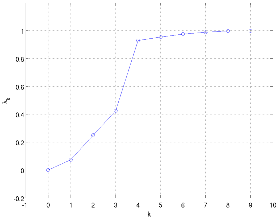


(a) λ_k

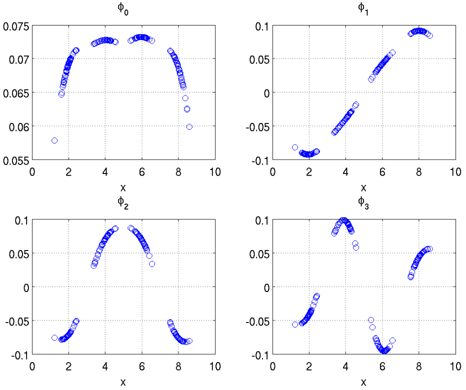


(b) $\boldsymbol{\phi}_k$

Using L_{sym}



(a) λ_k



(b) ϕ_k

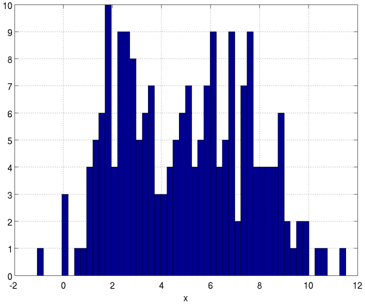
Simple Examples for Spectral Clustering . . .

- Now, let's consider a less clear cut case. This time, the dataset still consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j,\ j = 1, 2, 3, 4$. But now I set the larger standard deviation, i.e., $\sigma = 1$ instead of $\sigma = 0.25$.

- Then let's repeat the same experiments and see how the situation changes.

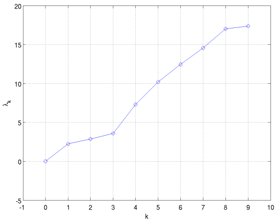
Simple Examples for Spectral Clustering . . .

- Now, let's consider a less clear cut case. This time, the dataset still consists of 200 random samples from four normal distributions $\mathcal{N}(\mu_j, \sigma^2)$ where $\mu_j = 2j$, $j = 1, 2, 3, 4$. But now I set the larger standard deviation, i.e., $\sigma = 1$ instead of $\sigma = 0.25$.

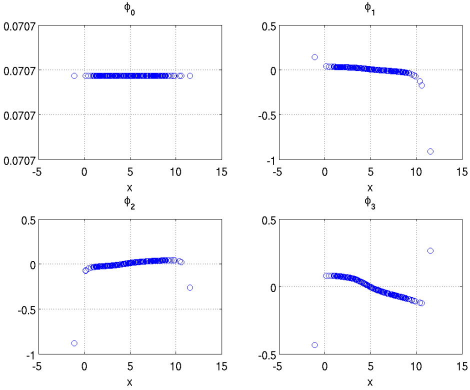


- Then let's repeat the same experiments and see how the situation changes.

Using L

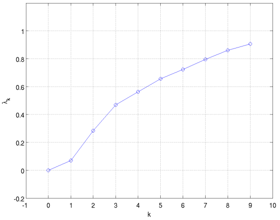


(a) λ_k

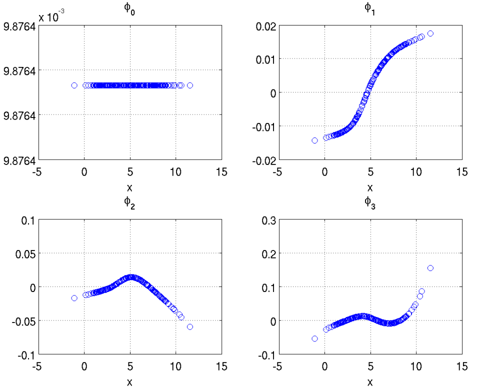


(b) ϕ_k

Using L_{rw}

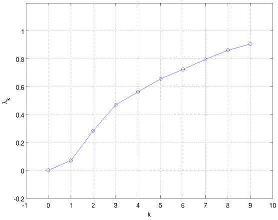


(a) λ_k

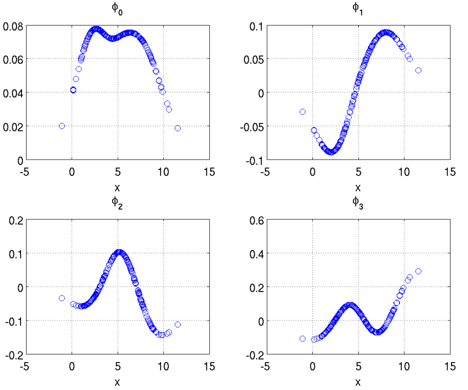


(b) $\boldsymbol{\phi}_k$

Using L_{sym}

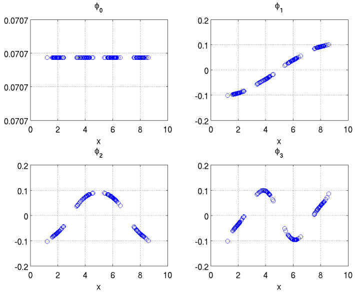


(a) λ_k

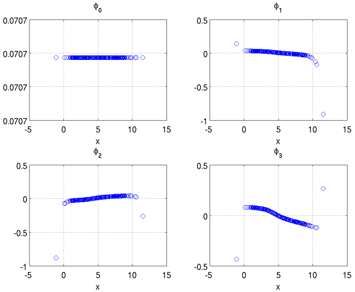


(b) $\boldsymbol{\phi}_k$

Using L

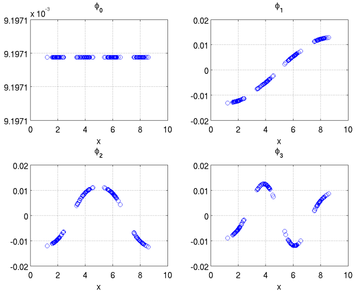


(a) $\sigma = 0.25$

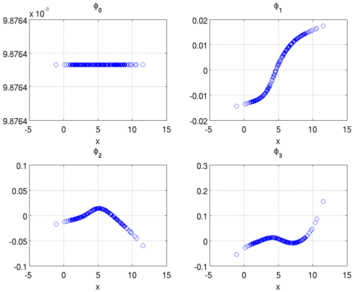


(b) $\sigma = 1$

Using L_{rw}

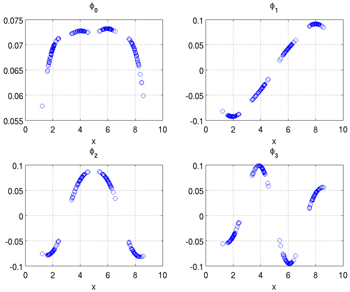


(a) $\sigma = 0.25$

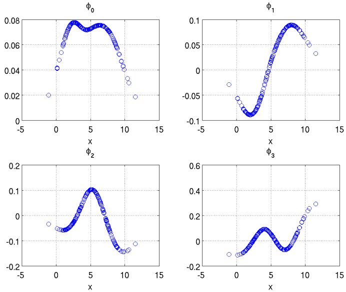


(b) $\sigma = 1$

Using L_{sym}



(a) $\sigma = 0.25$



(b) $\sigma = 1$

Observations

- For the clear cut case, L, L_{rw}, and L_{sym} all performed similarly.
- Yet, the eigenvalue distributions of L_{rw} and L_{sym} revealed the number of existing clusters more clearly than that of L.
- For the case with severer overlaps, L_{rw} and L_{sym} outperformed L.

Observations

- For the clear cut case, L, L_{rw}, and L_{sym} all performed similarly.
- Yet, the eigenvalue distributions of L_{rw} and L_{sym} revealed the number of existing clusters more clearly than that of L.
- For the case with severer overlaps, L_{rw} and L_{sym} outperformed L.

Observations

- For the clear cut case, L, L_{rw}, and L_{sym} all performed similarly.
- Yet, the eigenvalue distributions of L_{rw} and L_{sym} revealed the number of existing clusters more clearly than that of L.
- For the case with severer overlaps, L_{rw} and L_{sym} outperformed L.

Outline

Graph Cut Viewpoint

- Natural to consider spectral clustering as graph cut or graph partitioning.
- Let G be an undirected but weighted graph.
- Define a *measure of connectedness* or *cut cost* $C(X,Y)$ between two (not necessarily disjoint) vertex subsets $X, Y \subset V$ by

$$C(X,Y) := \sum_{x \in X, y \in Y} a_{xy}$$

- For a partition $V(G) = \bigcup_{i=1}^{k} X_i$, $X_i \cap X_j = \emptyset$, $i \neq j$, define the *cut* of $V(G)$ by

$$\mathrm{cut}(X_1, \ldots, X_k) := \frac{1}{2} \sum_{i=1}^{k} C(X_i, X_i^c),$$

where of course $X_i^c := V \setminus X_i$. Note: for $k=2$, $\mathrm{cut}(X_1,X_2) = C(X_1,X_2)$ with $X_2 = X_1^c$.

Graph Cut Viewpoint

- Natural to consider spectral clustering as graph cut or graph partitioning.
- Let G be an undirected but weighted graph.
- Define a *measure of connectedness* or *cut cost* $C(X,Y)$ between two (not necessarily disjoint) vertex subsets $X, Y \subset V$ by

$$C(X,Y) := \sum_{x \in X, y \in Y} a_{xy}$$

- For a partition $V(G) = \bigcup_{i=1}^{k} X_i$, $X_i \cap X_j = \emptyset$, $i \neq j$, define the *cut* of $V(G)$ by

$$\mathrm{cut}(X_1, \ldots, X_k) := \frac{1}{2} \sum_{i=1}^{k} C(X_i, X_i^c),$$

where of course $X_i^c := V \setminus X_i$. Note: for $k=2$, $\mathrm{cut}(X_1, X_2) = C(X_1, X_2)$ with $X_2 = X_1^c$.

Graph Cut Viewpoint

- Natural to consider spectral clustering as graph cut or graph partitioning.
- Let G be an undirected but weighted graph.
- Define a *measure of connectedness* or *cut cost* $C(X,Y)$ between two (not necessarily disjoint) vertex subsets $X,Y \subset V$ by

$$C(X,Y) := \sum_{x \in X, y \in Y} a_{xy}$$

- For a partition $V(G) = \bigcup_{i=1}^{k} X_i$, $X_i \cap X_j = \emptyset$, $i \neq j$, define the *cut* of $V(G)$ by

$$\mathrm{cut}(X_1,\ldots,X_k) := \frac{1}{2}\sum_{i=1}^{k} C(X_i, X_i^c),$$

where of course $X_i^c := V \setminus X_i$. Note: for $k=2$, $\mathrm{cut}(X_1,X_2) = C(X_1,X_2)$ with $X_2 = X_1^c$.

Graph Cut Viewpoint

- Natural to consider spectral clustering as graph cut or graph partitioning.
- Let G be an undirected but weighted graph.
- Define a *measure of connectedness* or *cut cost* $C(X,Y)$ between two (not necessarily disjoint) vertex subsets $X, Y \subset V$ by

$$C(X,Y) := \sum_{x \in X, y \in Y} a_{xy}$$

- For a partition $V(G) = \bigcup_{i=1}^{k} X_i$, $X_i \cap X_j = \emptyset$, $i \neq j$, define the *cut* of $V(G)$ by

$$\mathrm{cut}(X_1, \ldots, X_k) := \frac{1}{2} \sum_{i=1}^{k} C(X_i, X_i^c),$$

where of course $X_i^c := V \setminus X_i$. Note: for $k = 2$, $\mathrm{cut}(X_1, X_2) = C(X_1, X_2)$ with $X_2 = X_1^c$.

Graph Cut Viewpoint ...

- Given a weighted adjacency matrix $A(G)$, the simplest and most direct way to construct a partition of G is to solve the *mincut* problem:

$$\min_{V=\bigcup_{i=1}^{k} X_i} \operatorname{cut}(X_1, \ldots, X_k).$$

- Unfortunately, this often does not lead to satisfactory partitions. Why? $\Longrightarrow$ In many cases, the solution of mincut simply separates one individual vertex from the rest.
- One way to circumvent this problem is to explicitly request each X_i is "reasonably large".
- Two options: *RatioCut* (Hagen and Kahng, 1992) and *normalized cut*, a.k.a. *Ncut* (Shi and Malik, 2000).

Graph Cut Viewpoint . . .

- Given a weighted adjacency matrix $A(G)$, the simplest and most direct way to construct a partition of G is to solve the *mincut* problem:

$$\min_{V=\bigcup_{i=1}^{k} X_i} \mathrm{cut}(X_1, \ldots, X_k).$$

- Unfortunately, this often does not lead to satisfactory partitions. Why?
$\Longrightarrow$ In many cases, the solution of mincut simply separates one individual vertex from the rest.
- One way to circumvent this problem is to explicitly request each X_i is "reasonably large".
- Two options: *RatioCut* (Hagen and Kahng, 1992) and *normalized cut*, a.k.a. *Ncut* (Shi and Malik, 2000).

Graph Cut Viewpoint . . .

- Given a weighted adjacency matrix $A(G)$, the simplest and most direct way to construct a partition of G is to solve the *mincut* problem:

$$\min_{V=\bigcup_{i=1}^{k} X_i} \operatorname{cut}(X_1, \ldots, X_k).$$

- Unfortunately, this often does not lead to satisfactory partitions. Why? $\Longrightarrow$ In many cases, the solution of mincut simply separates one individual vertex from the rest.
- One way to circumvent this problem is to explicitly request each X_i is "reasonably large".
- Two options: *RatioCut* (Hagen and Kahng, 1992) and *normalized cut*, a.k.a. *Ncut* (Shi and Malik, 2000).

Graph Cut Viewpoint . . .

- Given a weighted adjacency matrix $A(G)$, the simplest and most direct way to construct a partition of G is to solve the *mincut* problem:

$$\min_{V=\bigcup_{i=1}^{k} X_i} \mathrm{cut}(X_1, \ldots, X_k).$$

- Unfortunately, this often does not lead to satisfactory partitions. Why? $\Longrightarrow$ In many cases, the solution of mincut simply separates one individual vertex from the rest.
- One way to circumvent this problem is to explicitly request each X_i is "reasonably large".
- Two options: *RatioCut* (Hagen and Kahng, 1992) and *normalized cut*, a.k.a. *Ncut* (Shi and Malik, 2000).

Graph Cut Viewpoint . . .

- Given a weighted adjacency matrix $A(G)$, the simplest and most direct way to construct a partition of G is to solve the *mincut* problem:

$$\min_{V=\bigcup_{i=1}^{k} X_i} \mathrm{cut}(X_1, \ldots, X_k).$$

- Unfortunately, this often does not lead to satisfactory partitions. Why? $\Longrightarrow$ In many cases, the solution of mincut simply separates one individual vertex from the rest.
- One way to circumvent this problem is to explicitly request each X_i is "reasonably large".
- Two options: *RatioCut* (Hagen and Kahng, 1992) and *normalized cut*, a.k.a. *Ncut* (Shi and Malik, 2000).

Graph Cut Viewpoint ...

$$\mathrm{RatioCut}(X_1,\ldots,X_k) := \sum_{i=1}^{k} \frac{C(X_i, X_i^c)}{|X_i|} = \sum_{i=1}^{k} \frac{\mathrm{cut}(X_i, X_i^c)}{|X_i|}.$$

where $|X_i|$ is the number of vertices in X_i.

$$\mathrm{Ncut}(X_1,\ldots,X_k) := \sum_{i=1}^{k} \frac{C(X_i, X_i^c)}{\mathrm{vol}(X_i)} = \sum_{i=1}^{k} \frac{\mathrm{cut}(X_i, X_i^c)}{\mathrm{vol}(X_i)},$$

where $\mathrm{vol}(X_i) := \sum_{v \in X_i} d(v)$ as we defined in Lecture 5.

Graph Cut Viewpoint ...

$$\begin{aligned}\mathrm{RatioCut}(X_1,\ldots,X_k) &:= \sum_{i=1}^{k} \frac{C(X_i, X_i^c)}{|X_i|} \\ &= \sum_{i=1}^{k} \frac{\mathrm{cut}(X_i, X_i^c)}{|X_i|}.\end{aligned}$$

where $|X_i|$ is the number of vertices in X_i.

$$\begin{aligned}\mathrm{Ncut}(X_1,\ldots,X_k) &:= \sum_{i=1}^{k} \frac{C(X_i, X_i^c)}{\mathrm{vol}(X_i)} \\ &= \sum_{i=1}^{k} \frac{\mathrm{cut}(X_i, X_i^c)}{\mathrm{vol}(X_i)},\end{aligned}$$

where $\mathrm{vol}(X_i) := \sum_{v \in X_i} d(v)$ as we defined in Lecture 5.

Graph Cut Viewpoint . . .

- Minimizing these quantities leads to a set of more *balanced* clusters $X_1, \dots, X_k$.
- Unfortunately, introducing these balancing conditions makes the minimization problem become *NP hard* (i.e., like the traveling salesman problem).
- Hence, we should be satisfied with *approximate* solutions that can be computed within a reasonable amount of time.

Graph Cut Viewpoint ...

- Minimizing these quantities leads to a set of more *balanced* clusters $X_1, \ldots, X_k$.
- Unfortunately, introducing these balancing conditions makes the minimization problem become *NP hard* (i.e., like the traveling salesman problem).
- Hence, we should be satisfied with *approximate* solutions that can be computed within a reasonable amount of time.

Graph Cut Viewpoint . . .

- Minimizing these quantities leads to a set of more *balanced* clusters $X_1, \ldots, X_k$.
- Unfortunately, introducing these balancing conditions makes the minimization problem become *NP hard* (i.e., like the traveling salesman problem).
- Hence, we should be satisfied with *approximate* solutions that can be computed within a reasonable amount of time.

Approximation of RatioCut for $k = 2$

- Want to achieve: $\min_{X \subset V} \mathrm{RatioCut}(X, X^c)$
- Define the vector $\boldsymbol{f} \in \mathbb{R}^n$ s.t.

$$f_i := \begin{cases} \sqrt{\frac{|X^c|}{|X|}} & \text{if } v_i \in X; \\ -\sqrt{\frac{|X|}{|X^c|}} & \text{if } v_i \in X^c. \end{cases}$$

- Then, the RatioCut objective function can be conveniently rewritten using L as follows:

$$\boldsymbol{f}^{\mathsf{T}} L \boldsymbol{f} = \frac{1}{2} \sum_{i,j=1}^{n} a_{ij} (f_i - f_j)^2$$

$$= \frac{1}{2} \sum_{v_i \in X; v_j \in X^c} a_{ij} \left(\sqrt{\frac{|X^c|}{|X|}} + \sqrt{\frac{|X|}{|X^c|}} \right)^2$$

$$+ \frac{1}{2} \sum_{v_i \in X^c; v_j \in X} a_{ij} \left(-\sqrt{\frac{|X^c|}{|X|}} - \sqrt{\frac{|X|}{|X^c|}} \right)^2$$

Approximation of RatioCut for $k = 2$

- Want to achieve: $\min_{X \subset V} \mathrm{RatioCut}(X, X^c)$
- Define the vector $\boldsymbol{f} \in \mathbb{R}^n$ s.t.

$$f_i := \begin{cases} \sqrt{\frac{|X^c|}{|X|}} & \text{if } v_i \in X; \\ -\sqrt{\frac{|X|}{|X^c|}} & \text{if } v_i \in X^c. \end{cases}$$

- Then, the RatioCut objective function can be conveniently rewritten using L as follows:

$$\begin{aligned} \boldsymbol{f}^{\mathsf{T}} L \boldsymbol{f} &= \frac{1}{2} \sum_{i,j=1}^{n} a_{ij} (f_i - f_j)^2 \\ &= \frac{1}{2} \sum_{v_i \in X; v_j \in X^c} a_{ij} \left(\sqrt{\frac{|X^c|}{|X|}} + \sqrt{\frac{|X|}{|X^c|}} \right)^2 \\ &\quad + \frac{1}{2} \sum_{v_i \in X^c; v_j \in X} a_{ij} \left(-\sqrt{\frac{|X^c|}{|X|}} - \sqrt{\frac{|X|}{|X^c|}} \right)^2 \end{aligned}$$

Approximation of RatioCut for $k = 2$

- Want to achieve: $\min_{X \subset V} \mathrm{RatioCut}(X, X^c)$
- Define the vector $\boldsymbol{f} \in \mathbb{R}^n$ s.t.

$$f_i := \begin{cases} \sqrt{\frac{|X^c|}{|X|}} & \text{if } v_i \in X; \\ -\sqrt{\frac{|X|}{|X^c|}} & \text{if } v_i \in X^c. \end{cases}$$

- Then, the RatioCut objective function can be conveniently rewritten using L as follows:

$$\boldsymbol{f}^\mathsf{T} L \boldsymbol{f} = \frac{1}{2} \sum_{i,j=1}^{n} a_{ij} (f_i - f_j)^2$$

$$= \frac{1}{2} \sum_{v_i \in X; v_j \in X^c} a_{ij} \left(\sqrt{\frac{|X^c|}{|X|}} + \sqrt{\frac{|X|}{|X^c|}} \right)^2$$

$$+ \frac{1}{2} \sum_{v_i \in X^c; v_j \in X} a_{ij} \left(-\sqrt{\frac{|X^c|}{|X|}} - \sqrt{\frac{|X|}{|X^c|}} \right)^2$$

Approximation of RatioCut for $k = 2$ ...

$$\vdots$$

$$\begin{aligned} \boldsymbol{f}^{\mathsf{T}} L \boldsymbol{f} &= \mathrm{cut}(X, X^c) \left(\frac{|X^c|}{|X|} + \frac{|X|}{|X^c|} + 2 \right) \\ &= \mathrm{cut}(X, X^c) \left(\frac{|X| + |X^c|}{|X|} + \frac{|X| + |X^c|}{|X^c|} \right) \\ &= |V| \, \mathrm{RatioCut}(X, X^c). \end{aligned}$$

In addition, we have

$$\begin{aligned} \sum_{i=1}^{n} f_i &= \sum_{v_i \in X} \sqrt{\frac{|X^c|}{|X|}} - \sum_{v_i \in X^c} \sqrt{\frac{|X|}{|X^c|}} \\ &= |X| \sqrt{\frac{|X^c|}{|X|}} - |X^c| \sqrt{\frac{|X|}{|X^c|}} = 0. \end{aligned}$$

Approximation of RatioCut for $k = 2$ ...

$$\vdots$$

$$\begin{aligned} \boldsymbol{f}^{\mathsf{T}} L \boldsymbol{f} &= \operatorname{cut}(X, X^c) \left(\frac{|X^c|}{|X|} + \frac{|X|}{|X^c|} + 2 \right) \\ &= \operatorname{cut}(X, X^c) \left(\frac{|X| + |X^c|}{|X|} + \frac{|X| + |X^c|}{|X^c|} \right) \\ &= |V| \operatorname{RatioCut}(X, X^c). \end{aligned}$$

In addition, we have

$$\begin{aligned} \sum_{i=1}^{n} f_i &= \sum_{v_i \in X} \sqrt{\frac{|X^c|}{|X|}} - \sum_{v_i \in X^c} \sqrt{\frac{|X|}{|X^c|}} \\ &= |X| \sqrt{\frac{|X^c|}{|X|}} - |X^c| \sqrt{\frac{|X|}{|X^c|}} = 0. \end{aligned}$$

Approximation of RatioCut for $k = 2$ ...

- Moreover,

$$\begin{aligned} \|\boldsymbol{f}\|^2 &= \sum_{i=1}^{n} f_i^2 = |X|\frac{|X^c|}{|X|} + |X^c|\frac{|X|}{|X^c|} \\ &= |X| + |X^c| = |V| = n. \end{aligned}$$

- Hence we have the following equivalent minimization problem:

$$\min_{X \subset V} \boldsymbol{f}^{\top} L \boldsymbol{f} \quad \text{subject to } \boldsymbol{f} \perp \boldsymbol{1}_n;\ f_i \text{ defined as above; } \|\boldsymbol{f}\| = \sqrt{n}.$$

- Unfortunately, this is still NP hard due to the definition of f_i.
- Now, relaxing the constraints, i.e., allowing f_i to take arbitrary values in $\mathbb{R}$ leads to the following:

$$\min_{\boldsymbol{f} \in \mathbb{R}^n} \boldsymbol{f}^{\top} L \boldsymbol{f} \quad \text{subject to } \boldsymbol{f} \perp \boldsymbol{1}_n;\ \|\boldsymbol{f}\| = \sqrt{n}.$$

- The solution to the above problem is nothing but $\boldsymbol{f} = \boldsymbol{\phi}_1$, *the Fiedler vector* of G thanks to the Rayleigh-Ritz Theorem!!

Approximation of RatioCut for $k = 2$ ...

- Moreover,

$$\begin{aligned} \|\boldsymbol{f}\|^2 &= \sum_{i=1}^{n} f_i^2 = |X| \frac{|X^c|}{|X|} + |X^c| \frac{|X|}{|X^c|} \\ &= |X| + |X^c| = |V| = n. \end{aligned}$$

- Hence we have the following equivalent minimization problem:

$$\min_{X \subset V} \boldsymbol{f}^{\top} L \boldsymbol{f} \quad \text{subject to } \boldsymbol{f} \perp \mathbf{1}_n;\ f_i \text{ defined as above; } \|\boldsymbol{f}\| = \sqrt{n}.$$

- Unfortunately, this is still NP hard due to the definition of f_i.
- Now, relaxing the constraints, i.e., allowing f_i to take arbitrary values in $\mathbb{R}$ leads to the following:

$$\min_{\boldsymbol{f} \in \mathbb{R}^n} \boldsymbol{f}^{\top} L \boldsymbol{f} \quad \text{subject to } \boldsymbol{f} \perp \mathbf{1}_n;\ \|\boldsymbol{f}\| = \sqrt{n}.$$

- The solution to the above problem is nothing but $\boldsymbol{f} = \boldsymbol{\phi}_1$, *the Fiedler vector* of G thanks to the Rayleigh-Ritz Theorem!!

Approximation of RatioCut for $k = 2$ ...

- Moreover,

$$\begin{aligned} \|\boldsymbol{f}\|^2 &= \sum_{i=1}^{n} f_i^2 = |X|\frac{|X^c|}{|X|} + |X^c|\frac{|X|}{|X^c|} \\ &= |X| + |X^c| = |V| = n. \end{aligned}$$

- Hence we have the following equivalent minimization problem:

$$\min_{X \subset V} \boldsymbol{f}^{\mathsf{T}} L \boldsymbol{f} \quad \text{subject to } \boldsymbol{f} \perp \mathbf{1}_n;\ f_i \text{ defined as above; } \|\boldsymbol{f}\| = \sqrt{n}.$$

- Unfortunately, this is still NP hard due to the definition of f_i.
- Now, relaxing the constraints, i.e., allowing f_i to take arbitrary values in $\mathbb{R}$ leads to the following:

$$\min_{\boldsymbol{f} \in \mathbb{R}^n} \boldsymbol{f}^{\mathsf{T}} L \boldsymbol{f} \quad \text{subject to } \boldsymbol{f} \perp \mathbf{1}_n;\ \|\boldsymbol{f}\| = \sqrt{n}.$$

- The solution to the above problem is nothing but $\boldsymbol{f} = \boldsymbol{\phi}_1$, *the Fiedler vector* of G thanks to the Rayleigh-Ritz Theorem!!

Approximation of RatioCut for $k = 2$ ...

- Moreover,

$$\begin{aligned}\|\boldsymbol{f}\|^2 &= \sum_{i=1}^{n} f_i^2 = |X|\frac{|X^c|}{|X|} + |X^c|\frac{|X|}{|X^c|} \\ &= |X| + |X^c| = |V| = n.\end{aligned}$$

- Hence we have the following equivalent minimization problem:

$$\min_{X \subset V} \boldsymbol{f}^{\top} L \boldsymbol{f} \quad \text{subject to } \boldsymbol{f} \perp \mathbf{1}_n;\ f_i \text{ defined as above; } \|\boldsymbol{f}\| = \sqrt{n}.$$

- Unfortunately, this is still NP hard due to the definition of f_i.
- Now, relaxing the constraints, i.e., allowing f_i to take arbitrary values in $\mathbb{R}$ leads to the following:

$$\min_{\boldsymbol{f} \in \mathbb{R}^n} \boldsymbol{f}^{\top} L \boldsymbol{f} \quad \text{subject to } \boldsymbol{f} \perp \mathbf{1}_n;\ \|\boldsymbol{f}\| = \sqrt{n}.$$

- The solution to the above problem is nothing but $\boldsymbol{f} = \boldsymbol{\phi}_1$, *the Fiedler vector* of G thanks to the Rayleigh-Ritz Theorem!!

Approximation of RatioCut for $k = 2$ ...

- Moreover,

$$\begin{aligned} \|\boldsymbol{f}\|^2 &= \sum_{i=1}^n f_i^2 = |X| \frac{|X^c|}{|X|} + |X^c| \frac{|X|}{|X^c|} \\ &= |X| + |X^c| = |V| = n. \end{aligned}$$

- Hence we have the following equivalent minimization problem:

$$\min_{X \subset V} \boldsymbol{f}^\top L \boldsymbol{f} \quad \text{subject to } \boldsymbol{f} \perp \mathbf{1}_n;\ f_i \text{ defined as above; } \|\boldsymbol{f}\| = \sqrt{n}.$$

- Unfortunately, this is still NP hard due to the definition of f_i.
- Now, relaxing the constraints, i.e., allowing f_i to take arbitrary values in $\mathbb{R}$ leads to the following:

$$\min_{\boldsymbol{f} \in \mathbb{R}^n} \boldsymbol{f}^\top L \boldsymbol{f} \quad \text{subject to } \boldsymbol{f} \perp \mathbf{1}_n;\ \|\boldsymbol{f}\| = \sqrt{n}.$$

- The solution to the above problem is nothing but $\boldsymbol{f} = \boldsymbol{\phi}_1$, *the Fiedler vector* of G thanks to the Rayleigh-Ritz Theorem!!

Approximation of RatioCut for arbitrary k

- In principle, the approach is similar to $k = 2$ case.
- Define the k indicator vectors $\boldsymbol{h}_j \in \mathbb{R}^n$, $j = 1, \ldots, k$, s.t.

$$h_{ij} := \begin{cases} 1/\sqrt{|X_j|} & \text{if } v_i \in X_j; \\ 0 & \text{otherwise.} \end{cases}$$

  Then, define $H := [\boldsymbol{h}_1 \cdots \boldsymbol{h}_k] \in \mathbb{R}^{n \times k}$. Observe also that $H^\top H = I_k$.
- Using the similar calculation as the $k = 2$ case, we have

$$(H^\top L H)_{ii} = \boldsymbol{h}_i^\top L \boldsymbol{h}_i = \frac{\mathrm{cut}(X_i, X_i^c)}{|X_i|}.$$

- Hence, we have

$$\mathrm{RatioCut}(X_1, \ldots, X_k) = \sum_{i=1}^{k} \boldsymbol{h}_i^\top L \boldsymbol{h}_i = \sum_{i=1}^{k} (H^\top L H)_{ii} = \mathrm{tr}(H^\top L H).$$

Approximation of RatioCut for arbitrary k

- In principle, the approach is similar to $k=2$ case.
- Define the k indicator vectors $\boldsymbol{h}_j \in \mathbb{R}^n$, $j=1,\ldots,k$, s.t.

$$h_{ij} := \begin{cases} 1/\sqrt{|X_j|} & \text{if } v_i \in X_j; \\ 0 & \text{otherwise.} \end{cases}$$

Then, define $H := [\boldsymbol{h}_1 \cdots \boldsymbol{h}_k] \in \mathbb{R}^{n\times k}$. Observe also that $H^\top H = I_k$.

- Using the similar calculation as the $k=2$ case, we have

$$(H^\top L H)_{ii} = \boldsymbol{h}_i^\top L \boldsymbol{h}_i = \frac{\operatorname{cut}(X_i, X_i^c)}{|X_i|}.$$

- Hence, we have

$$\operatorname{RatioCut}(X_1,\ldots,X_k) = \sum_{i=1}^k \boldsymbol{h}_i^\top L \boldsymbol{h}_i = \sum_{i=1}^k (H^\top L H)_{ii} = \operatorname{tr}(H^\top L H).$$

Approximation of RatioCut for arbitrary k

- In principle, the approach is similar to $k = 2$ case.
- Define the k indicator vectors $\boldsymbol{h}_j \in \mathbb{R}^n$, $j = 1, \ldots, k$, s.t.

$$h_{ij} := \begin{cases} 1/\sqrt{|X_j|} & \text{if } v_i \in X_j; \\ 0 & \text{otherwise.} \end{cases}$$

  Then, define $H := [\boldsymbol{h}_1 \cdots \boldsymbol{h}_k] \in \mathbb{R}^{n \times k}$. Observe also that $H^{\mathsf{T}} H = I_k$.
- Using the similar calculation as the $k = 2$ case, we have

$$(H^{\mathsf{T}} L H)_{ii} = \boldsymbol{h}_i^{\mathsf{T}} L \boldsymbol{h}_i = \frac{\mathrm{cut}(X_i, X_i^c)}{|X_i|}.$$

- Hence, we have

$$\mathrm{RatioCut}(X_1, \ldots, X_k) = \sum_{i=1}^{k} \boldsymbol{h}_i^{\mathsf{T}} L \boldsymbol{h}_i = \sum_{i=1}^{k} (H^{\mathsf{T}} L H)_{ii} = \mathrm{tr}(H^{\mathsf{T}} L H).$$

Approximation of RatioCut for arbitrary k

- In principle, the approach is similar to $k = 2$ case.
- Define the k indicator vectors $\boldsymbol{h}_j \in \mathbb{R}^n$, $j = 1, \ldots, k$, s.t.

$$h_{ij} := \begin{cases} 1/\sqrt{|X_j|} & \text{if } v_i \in X_j; \\ 0 & \text{otherwise.} \end{cases}$$

  Then, define $H := [\boldsymbol{h}_1 \cdots \boldsymbol{h}_k] \in \mathbb{R}^{n \times k}$. Observe also that $H^\top H = I_k$.
- Using the similar calculation as the $k = 2$ case, we have

$$(H^\top L H)_{ii} = \boldsymbol{h}_i^\top L \boldsymbol{h}_i = \frac{\mathrm{cut}(X_i, X_i^c)}{|X_i|}.$$

- Hence, we have

$$\mathrm{RatioCut}(X_1, \ldots, X_k) = \sum_{i=1}^{k} \boldsymbol{h}_i^\top L \boldsymbol{h}_i = \sum_{i=1}^{k} (H^\top L H)_{ii} = \mathrm{tr}(H^\top L H).$$

Approximation of RatioCut for arbitrary k ...

- Hence, the problem to optimize is:

$$\min_{V=\bigcup_{i=1}^{k} X_i} \operatorname{tr}(H^{\mathsf{T}} L H) \quad \text{subject to } H^{\mathsf{T}} H = I_k;\ H \text{ defined as above.}$$

- Again, this is still NP hard. Hence, relaxing this by allowing each h_{ij} to have arbitrary values in $\mathbb{R}$, we have

$$\min_{H \in \mathbb{R}^{n \times k}} \operatorname{tr}(H^{\mathsf{T}} L H) \quad \text{subject to } H^{\mathsf{T}} H = I_k.$$

- This can be solved by choosing $H = \Phi = [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}]$, i.e., the matrix consisting of the first k eigenvectors of L!

Approximation of RatioCut for arbitrary k ...

- Hence, the problem to optimize is:

$$\min_{V=\bigcup_{i=1}^{k} X_i} \operatorname{tr}(H^{\mathsf{T}} L H) \quad \text{subject to } H^{\mathsf{T}} H = I_k;\ H \text{ defined as above.}$$

- Again, this is still NP hard. Hence, relaxing this by allowing each h_{ij} to have arbitrary values in $\mathbb{R}$, we have

$$\min_{H \in \mathbb{R}^{n\times k}} \operatorname{tr}(H^{\mathsf{T}} L H) \quad \text{subject to } H^{\mathsf{T}} H = I_k.$$

- This can be solved by choosing $H = \Phi = [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}]$, i.e., the matrix consisting of the first k eigenvectors of L!

Approximation of RatioCut for arbitrary k . . .

- Hence, the problem to optimize is:

$$\min_{V=\bigcup_{i=1}^{k} X_i} \mathrm{tr}(H^{\mathsf{T}} L H) \quad \text{subject to } H^{\mathsf{T}} H = I_k;\ H \text{ defined as above.}$$

- Again, this is still NP hard. Hence, relaxing this by allowing each h_{ij} to have arbitrary values in $\mathbb{R}$, we have

$$\min_{H \in \mathbb{R}^{n \times k}} \mathrm{tr}(H^{\mathsf{T}} L H) \quad \text{subject to } H^{\mathsf{T}} H = I_k.$$

- This can be solved by choosing $H = \Phi = [\boldsymbol{\phi}_0 \cdots \boldsymbol{\phi}_{k-1}]$, i.e., the matrix consisting of the first k eigenvectors of L!

Approximation of Ncut

- In principle, the approach is similar to the RatioCut case. Replace $|X_i|$, $|X_i^c|$ in the RatioCut arguments by $\mathrm{vol}(X_i)$, $\mathrm{vol}(X_i^c)$, respectively.
- Then, after the similar relaxation of the constraints, for $k = 2$, we have

$$\min_{\boldsymbol{f} \in \mathbb{R}^n} \boldsymbol{f}^\top L \boldsymbol{f} \quad \text{subject to } D\boldsymbol{f} \perp \mathbf{1}_n; \boldsymbol{f}^\top D \boldsymbol{f} = \mathrm{vol}(V).$$

- Substituting $\boldsymbol{g} := D^{1/2}\boldsymbol{f}$ in the above minimization yields:

$$\min_{\boldsymbol{g} \in \mathbb{R}^n} \boldsymbol{g}^\top L_{\mathrm{sym}} \boldsymbol{g} \quad \text{subject to } \boldsymbol{g} \perp D^{1/2}\mathbf{1}_n; \|\boldsymbol{g}\|^2 = \mathrm{vol}(V).$$

- Hence, again by the Rayleigh-Ritz Theorem, $\boldsymbol{g} = \boldsymbol{\phi}_1^{\mathrm{sym}}$ (i.e., the eigenvector of L_{sym} corresponding to the smallest nonzero eigenvalue) is the solution, which leads to $\boldsymbol{f} = D^{-1/2}\boldsymbol{\phi}_1^{\mathrm{sym}} = \boldsymbol{\phi}_1^{\mathrm{rw}}$, i.e., the eigenvector corresponding to the smallest nonzero eigenvalue of the *generalized* eigenproblem: $L\boldsymbol{f} = \lambda D\boldsymbol{f}$.

Approximation of Ncut

- In principle, the approach is similar to the RatioCut case. Replace $|X_i|$, $|X_i^c|$ in the RatioCut arguments by $\mathrm{vol}(X_i)$, $\mathrm{vol}(X_i^c)$, respectively.
- Then, after the similar relaxation of the constraints, for $k = 2$, we have

$$\min_{\boldsymbol{f} \in \mathbb{R}^n} \boldsymbol{f}^\top L \boldsymbol{f} \quad \text{subject to } D\boldsymbol{f} \perp \mathbf{1}_n; \boldsymbol{f}^\top D \boldsymbol{f} = \mathrm{vol}(V).$$

- Substituting $\boldsymbol{g} := D^{1/2}\boldsymbol{f}$ in the above minimization yields:

$$\min_{\boldsymbol{g} \in \mathbb{R}^n} \boldsymbol{g}^\top L_{\mathrm{sym}} \boldsymbol{g} \quad \text{subject to } \boldsymbol{g} \perp D^{1/2}\mathbf{1}_n; \|\boldsymbol{g}\|^2 = \mathrm{vol}(V).$$

- Hence, again by the Rayleigh-Ritz Theorem, $\boldsymbol{g} = \boldsymbol{\phi}_1^{\mathrm{sym}}$ (i.e., the eigenvector of L_{sym} corresponding to the smallest nonzero eigenvalue) is the solution, which leads to $\boldsymbol{f} = D^{-1/2}\boldsymbol{\phi}_1^{\mathrm{sym}} = \boldsymbol{\phi}_1^{\mathrm{rw}}$, i.e., the eigenvector corresponding to the smallest nonzero eigenvalue of the *generalized* eigenproblem: $L\boldsymbol{f} = \lambda D\boldsymbol{f}$.

Approximation of Ncut

- In principle, the approach is similar to the RatioCut case. Replace $|X_i|$, $|X_i^c|$ in the RatioCut arguments by $\mathrm{vol}(X_i)$, $\mathrm{vol}(X_i^c)$, respectively.
- Then, after the similar relaxation of the constraints, for $k = 2$, we have

$$\min_{\boldsymbol{f}\in\mathbb{R}^n} \boldsymbol{f}^{\mathsf{T}} L \boldsymbol{f} \quad \text{subject to } D\boldsymbol{f} \perp \mathbf{1}_n; \boldsymbol{f}^{\mathsf{T}} D \boldsymbol{f} = \mathrm{vol}(V).$$

- Substituting $\boldsymbol{g} := D^{1/2}\boldsymbol{f}$ in the above minimization yields:

$$\min_{\boldsymbol{g}\in\mathbb{R}^n} \boldsymbol{g}^{\mathsf{T}} L_{\mathrm{sym}} \boldsymbol{g} \quad \text{subject to } \boldsymbol{g} \perp D^{1/2}\mathbf{1}_n; \|\boldsymbol{g}\|^2 = \mathrm{vol}(V).$$

- Hence, again by the Rayleigh-Ritz Theorem, $\boldsymbol{g} = \boldsymbol{\phi}_1^{\mathrm{sym}}$ (i.e., the eigenvector of L_{sym} corresponding to the smallest nonzero eigenvalue) is the solution, which leads to $\boldsymbol{f} = D^{-1/2}\boldsymbol{\phi}_1^{\mathrm{sym}} = \boldsymbol{\phi}_1^{\mathrm{rw}}$, i.e., the eigenvector corresponding to the smallest nonzero eigenvalue of the *generalized* eigenproblem: $L\boldsymbol{f} = \lambda D\boldsymbol{f}$.

Approximation of Ncut

- In principle, the approach is similar to the RatioCut case. Replace $|X_i|$, $|X_i^c|$ in the RatioCut arguments by $\mathrm{vol}(X_i)$, $\mathrm{vol}(X_i^c)$, respectively.
- Then, after the similar relaxation of the constraints, for $k = 2$, we have

$$\min_{\boldsymbol{f} \in \mathbb{R}^n} \boldsymbol{f}^{\mathsf{T}} L \boldsymbol{f} \quad \text{subject to } D\boldsymbol{f} \perp \mathbf{1}_n; \boldsymbol{f}^{\mathsf{T}} D \boldsymbol{f} = \mathrm{vol}(V).$$

- Substituting $\boldsymbol{g} := D^{1/2}\boldsymbol{f}$ in the above minimization yields:

$$\min_{\boldsymbol{g} \in \mathbb{R}^n} \boldsymbol{g}^{\mathsf{T}} L_{\mathrm{sym}} \boldsymbol{g} \quad \text{subject to } \boldsymbol{g} \perp D^{1/2}\mathbf{1}_n; \|\boldsymbol{g}\|^2 = \mathrm{vol}(V).$$

- Hence, again by the Rayleigh-Ritz Theorem, $\boldsymbol{g} = \boldsymbol{\phi}_1^{\mathrm{sym}}$ (i.e., the eigenvector of L_{sym} corresponding to the smallest nonzero eigenvalue) is the solution, which leads to $\boldsymbol{f} = D^{-1/2}\boldsymbol{\phi}_1^{\mathrm{sym}} = \boldsymbol{\phi}_1^{\mathrm{rw}}$, i.e., the eigenvector corresponding to the smallest nonzero eigenvalue of the *generalized* eigenproblem: $L\boldsymbol{f} = \lambda D\boldsymbol{f}$.

Approximation of Ncut . . .

- Finally for $k > 2$, let us first define $H = (h_{ij}) \in \mathbb{R}^{n \times k}$ by

$$h_{ij} := \begin{cases} 1/\sqrt{\mathrm{vol}(X_j)} & \text{if } v_i \in X_j; \\ 0 & \text{otherwise.} \end{cases}$$

- Then, using the reweighted matrix $\widetilde{H} := D^{1/2} H$ and the similar argument leads us to the following relaxed minimization problem:

$$\min_{\widetilde{H} \in \mathbb{R}^{n \times k}} \mathrm{tr}(\widetilde{H}^{\mathsf{T}} L_{\mathrm{sym}} \widetilde{H}) \quad \text{subject to } \widetilde{H}^{\mathsf{T}} \widetilde{H} = I_k.$$

- Its solution is $\widetilde{H} = \Phi^{\mathrm{sym}} = [\boldsymbol{\phi}_0^{\mathrm{sym}} \cdots \boldsymbol{\phi}_{k-1}^{\mathrm{sym}}]$, i.e., the first k eigenvectors of L_{sym}.
- The final solution $H = D^{-1/2} \widetilde{H}$ is therefore

$$H = \Phi^{\mathrm{rw}} = [\boldsymbol{\phi}_0^{\mathrm{rw}} \cdots \boldsymbol{\phi}_{k-1}^{\mathrm{rw}}],$$

i.e., *the first k eigenvectors of L_{rw} !!*

Approximation of Ncut . . .

- Finally for $k > 2$, let us first define $H = (h_{ij}) \in \mathbb{R}^{n \times k}$ by

$$h_{ij} := \begin{cases} 1/\sqrt{\mathrm{vol}(X_j)} & \text{if } v_i \in X_j; \\ 0 & \text{otherwise.} \end{cases}$$

- Then, using the reweighted matrix $\widetilde{H} := D^{1/2} H$ and the similar argument leads us to the following relaxed minimization problem:

$$\min_{\widetilde{H} \in \mathbb{R}^{n \times k}} \mathrm{tr}(\widetilde{H}^{\mathsf{T}} L_{\mathrm{sym}} \widetilde{H}) \quad \text{subject to } \widetilde{H}^{\mathsf{T}} \widetilde{H} = I_k.$$

- Its solution is $\widetilde{H} = \Phi^{\mathrm{sym}} = [\boldsymbol{\phi}_0^{\mathrm{sym}} \cdots \boldsymbol{\phi}_{k-1}^{\mathrm{sym}}]$, i.e., the first k eigenvectors of L_{sym}.
- The final solution $H = D^{-1/2} \widetilde{H}$ is therefore

$$H = \Phi^{\mathrm{rw}} = [\boldsymbol{\phi}_0^{\mathrm{rw}} \cdots \boldsymbol{\phi}_{k-1}^{\mathrm{rw}}],$$

i.e., *the first k eigenvectors of L_{rw} !!*

Approximation of Ncut ...

- Finally for $k > 2$, let us first define $H = (h_{ij}) \in \mathbb{R}^{n \times k}$ by

$$h_{ij} := \begin{cases} 1/\sqrt{\mathrm{vol}(X_j)} & \text{if } v_i \in X_j; \\ 0 & \text{otherwise.} \end{cases}$$

- Then, using the reweighted matrix $\widetilde{H} := D^{1/2} H$ and the similar argument leads us to the following relaxed minimization problem:

$$\min_{\widetilde{H} \in \mathbb{R}^{n \times k}} \mathrm{tr}(\widetilde{H}^{\mathsf{T}} L_{\mathrm{sym}} \widetilde{H}) \quad \text{subject to } \widetilde{H}^{\mathsf{T}} \widetilde{H} = I_k.$$

- Its solution is $\widetilde{H} = \Phi^{\mathrm{sym}} = [\boldsymbol{\phi}_0^{\mathrm{sym}} \cdots \boldsymbol{\phi}_{k-1}^{\mathrm{sym}}]$, i.e., the first k eigenvectors of L_{sym}.
- The final solution $H = D^{-1/2} \widetilde{H}$ is therefore

$$H = \Phi^{\mathrm{rw}} = [\boldsymbol{\phi}_0^{\mathrm{rw}} \cdots \boldsymbol{\phi}_{k-1}^{\mathrm{rw}}],$$

i.e., *the first k eigenvectors of L_{rw} !!*

Approximation of Ncut . . .

- Finally for $k > 2$, let us first define $H = (h_{ij}) \in \mathbb{R}^{n \times k}$ by

$$h_{ij} := \begin{cases} 1/\sqrt{\mathrm{vol}(X_j)} & \text{if } v_i \in X_j; \\ 0 & \text{otherwise.} \end{cases}$$

- Then, using the reweighted matrix $\widetilde{H} := D^{1/2} H$ and the similar argument leads us to the following relaxed minimization problem:

$$\min_{\widetilde{H} \in \mathbb{R}^{n \times k}} \mathrm{tr}(\widetilde{H}^{\mathsf{T}} L_{\mathrm{sym}} \widetilde{H}) \quad \text{subject to } \widetilde{H}^{\mathsf{T}} \widetilde{H} = I_k.$$

- Its solution is $\widetilde{H} = \Phi^{\mathrm{sym}} = [\boldsymbol{\phi}_0^{\mathrm{sym}} \cdots \boldsymbol{\phi}_{k-1}^{\mathrm{sym}}]$, i.e., the first k eigenvectors of L_{sym}.
- The final solution $H = D^{-1/2} \widetilde{H}$ is therefore

$$H = \Phi^{\mathrm{rw}} = [\boldsymbol{\phi}_0^{\mathrm{rw}} \cdots \boldsymbol{\phi}_{k-1}^{\mathrm{rw}}],$$

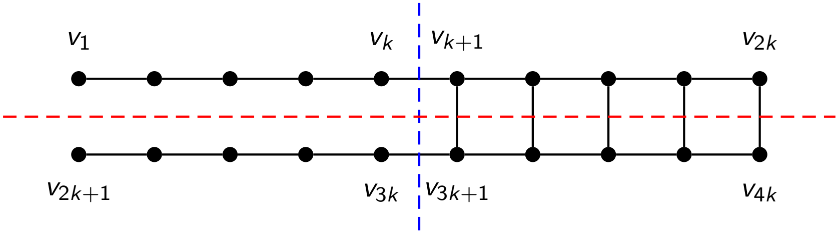
i.e., *the first k eigenvectors of L_{rw} !!*

Caveat

- The partition obtained by the relaxed approximate minimization problem is not necessarily the same as the solution of the exact mincut problem.
- An example of "cockroach graphs" found by Guattery and Miller (1998).
- The ideal RatioCut splits V into $X = \{v_1, \ldots, v_k, v_{2k+1}, \ldots, v_{3k}\}$ and $X^c = \{v_{k+1} \ldots, v_{2k}, v_{3k+1}, \ldots, v_{4k}\}$.
- On the other hand, the spectral clustering using ϕ_1 of $L(G)$ splits V into $Y = \{v_1, \ldots, v_{2k}\}$ and $Y^c = \{v_{2k+1}, \ldots, v_{4k}\}$.

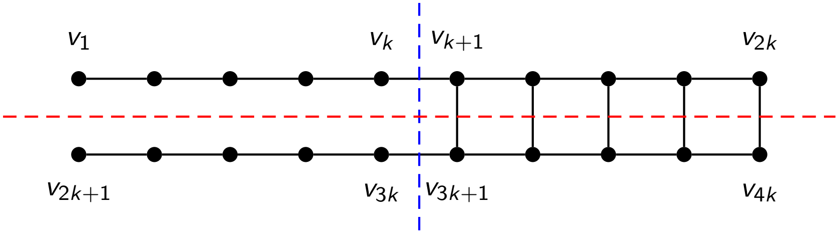
Caveat

- The partition obtained by the relaxed approximate minimization problem is not necessarily the same as the solution of the exact mincut problem.
- An example of "cockroach graphs" found by Guattery and Miller (1998).
- The ideal RatioCut splits V into $X = \{v_1, \ldots, v_k, v_{2k+1}, \ldots, v_{3k}\}$ and $X^c = \{v_{k+1} \ldots, v_{2k}, v_{3k+1}, \ldots, v_{4k}\}$.
- On the other hand, the spectral clustering using ϕ_1 of $L(G)$ splits V into $Y = \{v_1, \ldots, v_{2k}\}$ and $Y^c = \{v_{2k+1}, \ldots, v_{4k}\}$.



Caveat

- The partition obtained by the relaxed approximate minimization problem is not necessarily the same as the solution of the exact mincut problem.
- An example of "cockroach graphs" found by Guattery and Miller (1998).
- The ideal RatioCut splits V into $X = \{v_1, \ldots, v_k, v_{2k+1}, \ldots, v_{3k}\}$ and $X^c = \{v_{k+1} \ldots, v_{2k}, v_{3k+1}, \ldots, v_{4k}\}$.
- On the other hand, the spectral clustering using ϕ_1 of $L(G)$ splits V into $Y = \{v_1, \ldots, v_{2k}\}$ and $Y^c = \{v_{2k+1}, \ldots, v_{4k}\}$.



Caveat

- The partition obtained by the relaxed approximate minimization problem is not necessarily the same as the solution of the exact mincut problem.
- An example of "cockroach graphs" found by Guattery and Miller (1998).
- The ideal RatioCut splits V into $X = \{v_1, \dots, v_k, v_{2k+1}, \dots, v_{3k}\}$ and $X^c = \{v_{k+1} \dots, v_{2k}, v_{3k+1}, \dots, v_{4k}\}$.
- On the other hand, the spectral clustering using $\boldsymbol{\phi}_1$ of $L(G)$ splits V into $Y = \{v_1, \dots, v_{2k}\}$ and $Y^c = \{v_{2k+1}, \dots, v_{4k}\}$.