MATH 567: Mathematical Techniques in Data Science
Clustering II

Dominique Guillot

Departments of Mathematical Sciences
University of Delaware

May 8, 2017

This lecture is based on U. von Luxburg, *A Tutorial on Spectral Clustering*, Statistics and Computing, 17 (4), 2007.

Spectral clustering: overview

Overview of spectral clustering:

1. Construct a *similarity matrix* measuring the similarity of pairs of objects (e.g. $s_{ij} = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$).

Spectral clustering: overview

Overview of spectral clustering:

1. Construct a *similarity matrix* measuring the similarity of pairs of objects (e.g. $s_{ij} = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$).
2. Use the similarity matrix to construct a (weighted or unweighted) graph.

Spectral clustering: overview

Overview of spectral clustering:

1. Construct a *similarity matrix* measuring the similarity of pairs of objects (e.g. $s_{ij} = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$).
2. Use the similarity matrix to construct a (weighted or unweighted) graph.
3. Compute eigenvectors of the (normalized or unnormalized) *graph Laplacian*:

$$L = D - W, \qquad L_{\mathrm{sym}} = D^{-1/2} L D^{-1/2}.$$

Spectral clustering: overview

Overview of spectral clustering:

1. Construct a *similarity matrix* measuring the similarity of pairs of objects (e.g. $s_{ij} = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$).
2. Use the similarity matrix to construct a (weighted or unweighted) graph.
3. Compute eigenvectors of the (normalized or unnormalized) *graph Laplacian*:

$$L = D - W, \qquad L_{\text{sym}} = D^{-1/2} L D^{-1/2}.$$

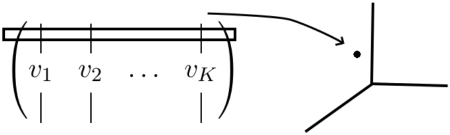
4. Construct a matrix containing the first K eigenvectors of L or L_{sym} as columns.

Overview of spectral clustering:

1. Construct a *similarity matrix* measuring the similarity of pairs of objects (e.g. $s_{ij} = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$).
2. Use the similarity matrix to construct a (weighted or unweighted) graph.
3. Compute eigenvectors of the (normalized or unnormalized) *graph Laplacian*:

$$L = D - W, \qquad L_{\text{sym}} = D^{-1/2} L D^{-1/2}.$$

4. Construct a matrix containing the first K eigenvectors of L or L_{sym} as columns.
5. Each row identifies a vertex of the graph to a point in $\mathbb{R}^K$.



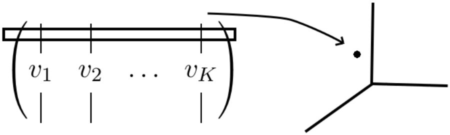
Spectral clustering: overview

Overview of spectral clustering:

1. Construct a *similarity matrix* measuring the similarity of pairs of objects (e.g. $s_{ij} = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$).
2. Use the similarity matrix to construct a (weighted or unweighted) graph.
3. Compute eigenvectors of the (normalized or unnormalized) *graph Laplacian*:

$$L = D - W, \qquad L_{\text{sym}} = D^{-1/2} L D^{-1/2}.$$

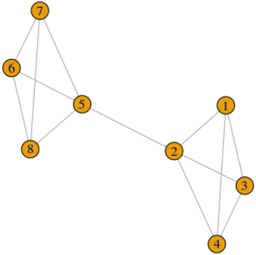
4. Construct a matrix containing the first K eigenvectors of L or L_{sym} as columns.
5. Each row identifies a vertex of the graph to a point in $\mathbb{R}^K$.



6. Cluster those points using the K-means algorithm.

Example

Let us try to cluster the following graph:



$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}, \qquad L = \begin{pmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 3 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 4 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 3 \end{pmatrix}$$

We have:

$$v_2 = (-0.3825277, -0.2470177, -0.3825277, -0.3825277, 0.2470177, 0.3825277, 0.3825277, 0.3825277)^T .$$

The unnormalized Laplacian

Proposition: The matrix L satisfies the following properties:

The unnormalized Laplacian

Proposition: The matrix L satisfies the following properties:

1. For any $f \in \mathbb{R}^n$:

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

The unnormalized Laplacian

Proposition: The matrix L satisfies the following properties:

1. For any $f \in \mathbb{R}^n$:

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

2. L is symmetric and positive semidefinite.

The unnormalized Laplacian

Proposition: The matrix L satisfies the following properties:

1. For any $f \in \mathbb{R}^n$:

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

2. L is symmetric and positive semidefinite.
3. 0 is an eigenvalue of L with associated constant eigenvector $\mathbb{1}$.

The unnormalized Laplacian

Proposition: The matrix L satisfies the following properties:

1. For any $f \in \mathbb{R}^n$:

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

2. L is symmetric and positive semidefinite.
3. 0 is an eigenvalue of L with associated constant eigenvector $\mathbb{1}$.

Proof:

The unnormalized Laplacian

Proposition: The matrix L satisfies the following properties:

1. For any $f \in \mathbb{R}^n$:
$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$
2. L is symmetric and positive semidefinite.
3. 0 is an eigenvalue of L with associated constant eigenvector $\mathbb{1}$.

Proof: To prove (1),

$$\begin{aligned} f^T L f = f^T D f - f^T W f &= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} w_{ij} f_i f_j \\ &= \frac{1}{2} \left(\sum_{i=1}^{n} d_i f_i^2 - 2 \sum_{i,j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n} d_j f_j^2 \right) \\ &= \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2. \end{aligned}$$

(2) follows from (1). (3) is easy. □

The unnormalized Laplacian (cont.)

Proposition: Let G be an undirected graph with non-negative weights. Then:

The unnormalized Laplacian (cont.)

Proposition: Let G be an undirected graph with non-negative weights. Then:

1. The multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \ldots, A_k$ in the graph.

The unnormalized Laplacian (cont.)

Proposition: Let G be an undirected graph with non-negative weights. Then:

1. The multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \ldots, A_k$ in the graph.
2. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ of those components.

Proposition: Let G be an undirected graph with non-negative weights. Then:

1. The multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \ldots, A_k$ in the graph.
2. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ of those components.

Proof:

Proposition: Let G be an undirected graph with non-negative weights. Then:

1. The multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \ldots, A_k$ in the graph.
2. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ of those components.

Proof: If f is an eigenvector associated to $\lambda = 0$, then

$$0 = f^T L f = \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

It follows that $f_i = f_j$ whenever $w_{ij} > 0$. Thus f is constant on the connected components of G. We conclude that the eigenspace of 0 is contained in $\mathrm{span}(\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k})$. Conversely, it is not hard to see that each $\mathbb{1}_{A_i}$ is an eigenvector associated to 0 (write L in block diagonal form). $\square$

Proposition: The normalized Laplacians satisfy the following properties:

1. For every $f \in \mathbb{R}^n$, we have

$$f^T L_{\mathrm{sym}} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 .$$

2. λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ is an eigenvalue of L_{sym} with eigenvector $w = D^{1/2} u$.
3. λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ and u solve the generalized eigenproblem $Lu = \lambda Du$.

Proof: The proof of (1) is similar to the proof of the analogous result for the unnormalized Laplacian. (2) and (3) follow easily by using appropriate rescalings.

The normalized Laplacians (cont.)

Proposition: Let G be an undirected graph with non-negative weights. Then:

1. The multiplicity k of the eigenvalue 0 of both L_{sym} and L_{rw} equals the number of connected components $A_1, \ldots, A_k$ in the graph.
2. For L_{rw}, the eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_i}$, $i = 1, \ldots, k$.
3. For L_{sym}, the eigenspace of eigenvalue 0 is spanned by the vectors $D^{1/2}\mathbb{1}_{A_i}$, $i = 1, \ldots, k$.

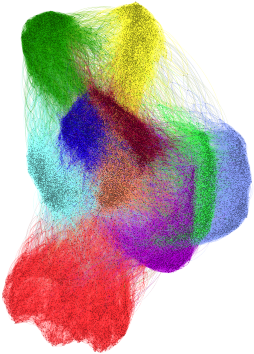
Proof: Similar to the proof of the analogous result for the unnormalized Laplacian.

Graph cuts

- G graph with (weighted) adjacency matrix $W = (w_{ij})$.
- We define:

$$W(A,B) := \sum_{i \in A, j \in B} w_{ij}.$$

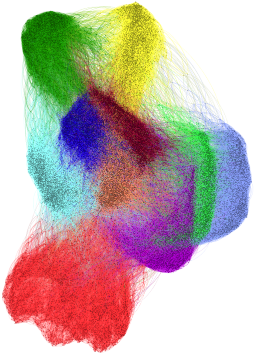
- $|A| :=$ number of vertices in A.
- $\mathrm{vol}(A) := \sum_{i \in A} d_i$.



Graph cuts

- G graph with (weighted) adjacency matrix $W = (w_{ij})$.
- We define:

$$W(A, B) := \sum_{i \in A, j \in B} w_{ij}.$$



- $|A| :=$ number of vertices in A.
- $\mathrm{vol}(A) := \sum_{i \in A} d_i$.

Given a partition $A_1, \ldots, A_k$ of the vertices of G, we let

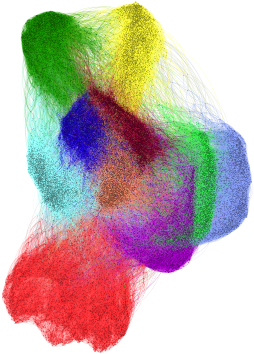
$$\mathrm{cut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A}_i).$$

Graph cuts

- G graph with (weighted) adjacency matrix $W = (w_{ij})$.
- We define:

$$W(A,B) := \sum_{i \in A, j \in B} w_{ij}.$$

- $|A| :=$ number of vertices in A.
- $\mathrm{vol}(A) := \sum_{i \in A} d_i$.



Given a partition $A_1, \dots, A_k$ of the vertices of G, we let

$$\mathrm{cut}(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A}_i).$$

The min-cut problem consists of solving:

$$\min_{\substack{V = A_1 \cup \cdots \cup A_k \\ A_i \cap A_j = \emptyset \ \forall i \neq j}} \mathrm{cut}(A_1, \dots, A_k).$$

Graph cuts (cont.)

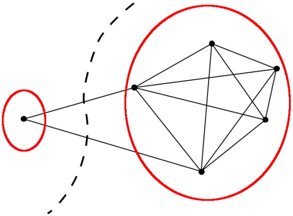
- The min-cut problem can be solved efficiently when $k = 2$ (see Stoer and Wagner 1997).

Graph cuts (cont.)

- The min-cut problem can be solved efficiently when $k = 2$ (see Stoer and Wagner 1997).
- In practice it often does not lead to satisfactory partitions.

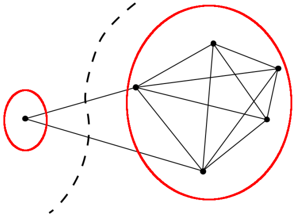
Graph cuts (cont.)

- The min-cut problem can be solved efficiently when $k = 2$ (see Stoer and Wagner 1997).
- In practice it often does not lead to satisfactory partitions.
- In many cases, the solution of min-cut simply separates one individual vertex from the rest of the graph.



Graph cuts (cont.)

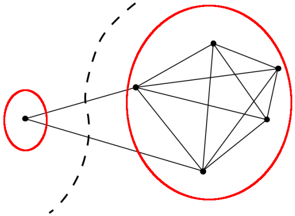
- The min-cut problem can be solved efficiently when $k = 2$ (see Stoer and Wagner 1997).
- In practice it often does not lead to satisfactory partitions.
- In many cases, the solution of min-cut simply separates one individual vertex from the rest of the graph.



- We would like clusters to have a reasonably large number of points.

Graph cuts (cont.)

- The min-cut problem can be solved efficiently when $k = 2$ (see Stoer and Wagner 1997).
- In practice it often does not lead to satisfactory partitions.
- In many cases, the solution of min-cut simply separates one individual vertex from the rest of the graph.



- We would like clusters to have a reasonably large number of points.
- We therefore modify the min-cut problem to enforce such constraints.

Balanced cuts

The two most common objective functions that are used as a replacement to the min-cut objective are:

1. RatioCut (Hagen and Kahng, 1992):

$$\mathrm{RatioCut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{|A_i|} = \sum_{i=1}^{k} \frac{\mathrm{cut}(A_i, \overline{A}_i)}{|A_i|}.$$

Balanced cuts

The two most common objective functions that are used as a replacement to the min-cut objective are:

1. RatioCut (Hagen and Kahng, 1992):

$$\text{RatioCut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{|A_i|} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{|A_i|}.$$

2. Normalized cut (Shi and Malik, 2000):

$$\text{Ncut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{\text{vol}(A_i)} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{\text{vol}(A_i)}.$$

Balanced cuts

The two most common objective functions that are used as a replacement to the min-cut objective are:

1. RatioCut (Hagen and Kahng, 1992):

$$\text{RatioCut}(A_1, \ldots, A_k) := \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{|A_i|} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{|A_i|}.$$

2. Normalized cut (Shi and Malik, 2000):

$$\text{Ncut}(A_1, \ldots, A_k) := \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{\text{vol}(A_i)} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{\text{vol}(A_i)}.$$

- Note: both objective functions take larger values when the clusters A_i are “small”.

Balanced cuts

The two most common objective functions that are used as a replacement to the min-cut objective are:

1. RatioCut (Hagen and Kahng, 1992):

$$\mathrm{RatioCut}(A_1, \dots, A_k) := \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{|A_i|} = \sum_{i=1}^{k} \frac{\mathrm{cut}(A_i, \overline{A}_i)}{|A_i|}.$$

2. Normalized cut (Shi and Malik, 2000):

$$\mathrm{Ncut}(A_1, \dots, A_k) := \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{\mathrm{vol}(A_i)} = \sum_{i=1}^{k} \frac{\mathrm{cut}(A_i, \overline{A}_i)}{\mathrm{vol}(A_i)}.$$

- Note: both objective functions take larger values when the clusters A_i are "small".
- Resulting clusters are more "balanced".

Balanced cuts

The two most common objective functions that are used as a replacement to the min-cut objective are:

1. RatioCut (Hagen and Kahng, 1992):

$$\text{RatioCut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{|A_i|} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{|A_i|}.$$

2. Normalized cut (Shi and Malik, 2000):

$$\text{Ncut}(A_1, \ldots, A_k) := \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{\text{vol}(A_i)} = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{\text{vol}(A_i)}.$$

- Note: both objective functions take larger values when the clusters A_i are "small".
- Resulting clusters are more "balanced".
- However, the resulting problems are NP hard - see Wagner and Wagner (1993).

Spectral clustering

Spectral clustering provides a way to *relax* the RatioCut and the Normalized cut problems.

Spectral clustering

Spectral clustering provides a way to *relax* the RatioCut and the Normalized cut problems.

Strategy:

1. Express the original problem as a linear algebra problem involving discrete/combinatorial constraints.

Spectral clustering

Spectral clustering provides a way to *relax* the RatioCut and the Normalized cut problems.

Strategy:

1. Express the original problem as a linear algebra problem involving discrete/combinatorial constraints.
2. Relax/remove the constraints.

Spectral clustering provides a way to *relax* the RatioCut and the Normalized cut problems.

Strategy:

1. Express the original problem as a linear algebra problem involving discrete/combinatorial constraints.
2. Relax/remove the constraints.

RatioCut with $k = 2$: solve

$$\min_{A \subset V} \mathrm{RatioCut}(A, \overline{A}).$$

Spectral clustering

Spectral clustering provides a way to *relax* the RatioCut and the Normalized cut problems.

Strategy:

1. Express the original problem as a linear algebra problem involving discrete/combinatorial constraints.
2. Relax/remove the constraints.

RatioCut with $k = 2$: solve

$$\min_{A \subset V} \mathrm{RatioCut}(A, \overline{A}).$$

Given $A \subset V$, let $f \in \mathbb{R}^n$ be given by

$$f_i := \begin{cases} \sqrt{|\overline{A}|/|A|} & \text{if } v_i \in A \\ -\sqrt{|A|/|\overline{A}|} & \text{if } v_i \notin A. \end{cases}$$

Relaxing RatioCut

Let $L = D - W$ be the (unnormalized) Laplacian of G. Then

$$
\begin{aligned}
f^T L f &= \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2 \\
&= \frac{1}{2} \sum_{i \in A, j \in \overline{A}} w_{ij} \left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}} \right)^2 + \frac{1}{2} \sum_{i \in \overline{A}, j \in A} w_{ij} \left(-\sqrt{\frac{|\overline{A}|}{|A|}} - \sqrt{\frac{|A|}{|\overline{A}|}} \right)^2 \\
&= W(A, \overline{A}) \left(2 + \frac{|\overline{A}|}{|A|} + \frac{|A|}{|\overline{A}|} \right) \\
&= W(A, \overline{A}) \left(\frac{|A| + |\overline{A}|}{|A|} + \frac{|A| + |\overline{A}|}{|\overline{A}|} \right) \\
&= |V| \cdot \frac{1}{2} \left(\frac{W(A, \overline{A})}{|A|} + \frac{W(\overline{A}, A)}{|\overline{A}|} \right) \\
&= |V| \cdot \mathrm{RatioCut}(A, \overline{A}).
\end{aligned}
$$

since $|A| + |\overline{A}| = |V|$, and $W(A, \overline{A}) = W(\overline{A}, A)$.

Relaxing RatioCut (cont.)

- We showed:

$$f^T L f = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 = |V| \cdot \mathrm{RatioCut}(A, \overline{A}).$$

Relaxing RatioCut (cont.)

- We showed:

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2 = |V| \cdot \mathrm{RatioCut}(A, \overline{A}).$$

- Moreover, note that

$$\sum_{i=1}^{n} f_i = \sum_{i \in A} \sqrt{\frac{|\overline{A}|}{|A|}} - \sum_{i \in \overline{A}} \sqrt{\frac{|A|}{|\overline{A}|}} = |A| \cdot \sqrt{\frac{|\overline{A}|}{|A|}} - |\overline{A}| \cdot \sqrt{\frac{|A|}{|\overline{A}|}} = 0.$$

Relaxing RatioCut (cont.)

- We showed:

$$f^T L f = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 = |V| \cdot \text{RatioCut}(A, \overline{A}).$$

- Moreover, note that

$$\sum_{i=1}^{n} f_i = \sum_{i \in A} \sqrt{\frac{|\overline{A}|}{|A|}} - \sum_{i \in \overline{A}} \sqrt{\frac{|A|}{|\overline{A}|}} = |A| \cdot \sqrt{\frac{|\overline{A}|}{|A|}} - |\overline{A}| \cdot \sqrt{\frac{|A|}{|\overline{A}|}} = 0.$$

Thus $f \perp \mathbb{1}$.

Relaxing RatioCut (cont.)

- We showed:

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2 = |V| \cdot \mathrm{RatioCut}(A, \overline{A}).$$

- Moreover, note that

$$\sum_{i=1}^{n} f_i = \sum_{i \in A} \sqrt{\frac{|\overline{A}|}{|A|}} - \sum_{i \in \overline{A}} \sqrt{\frac{|A|}{|\overline{A}|}} = |A| \cdot \sqrt{\frac{|\overline{A}|}{|A|}} - |\overline{A}| \cdot \sqrt{\frac{|A|}{|\overline{A}|}} = 0.$$

Thus $f \perp \mathbb{1}$.

- Finally,

$$\|f\|_2^2 = \sum_{i=1}^{n} f_i^2 = |A| \cdot \frac{|\overline{A}|}{|A|} + |\overline{A}| \cdot \frac{|A|}{|\overline{A}|} = |V| = n.$$

Relaxing RatioCut (cont.)

- We showed:

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2 = |V| \cdot \mathrm{RatioCut}(A, \overline{A}).$$

- Moreover, note that

$$\sum_{i=1}^{n} f_i = \sum_{i \in A} \sqrt{\frac{|\overline{A}|}{|A|}} - \sum_{i \in \overline{A}} \sqrt{\frac{|A|}{|\overline{A}|}} = |A| \cdot \sqrt{\frac{|\overline{A}|}{|A|}} - |\overline{A}| \cdot \sqrt{\frac{|A|}{|\overline{A}|}} = 0.$$

Thus $f \perp \mathbb{1}$.

- Finally,

$$\|f\|_2^2 = \sum_{i=1}^{n} f_i^2 = |A| \cdot \frac{|\overline{A}|}{|A|} + |\overline{A}| \cdot \frac{|A|}{|\overline{A}|} = |V| = n.$$

Thus, we have showed that the Ratio-Cut problem is equivalent to

$$\min_{A \subset V} f^T L f$$

$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}, f_i \text{ defined as above.}$$

Relaxing RatioCut (cont.)

We have:

$$\min_{A \subset V} f^T L f$$
$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}, f_i \text{ defined as above.}$$

Relaxing RatioCut (cont.)

We have:

$$\min_{A \subset V} f^T L f$$
$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}, f_i \text{ defined as above.}$$

- This is a discrete optimization problem since the entries of f can only take two values: $\sqrt{|\overline{A}|/|A|}$ and $-\sqrt{|A|/|\overline{A}|}$.

Relaxing RatioCut (cont.)

We have:

$$\min_{A \subset V} f^T L f$$
$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}, f_i \text{ defined as above.}$$

- This is a discrete optimization problem since the entries of f can only take two values: $\sqrt{|\overline{A}|/|A|}$ and $-\sqrt{|A|/|\overline{A}|}$.
- The problem is NP hard.

Relaxing RatioCut (cont.)

We have:

$$\min_{A \subset V} f^T L f$$
$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}, f_i \text{ defined as above.}$$

- This is a discrete optimization problem since the entries of f can only take two values: $\sqrt{|\overline{A}|/|A|}$ and $-\sqrt{|A|/|\overline{A}|}$.
- The problem is NP hard.

The natural relaxation of the problem is to **remove the discreteness condition** on f and solve

$$\min_{f \in \mathbb{R}^n} f^T L f$$
$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}.$$

Relaxing RatioCut (cont.)

- Using properties of the Rayleigh quotient, it is not hard to show that the solution of

$$\min_{f \in \mathbb{R}^n} f^T L f$$
$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}.$$

is an eigenvector of L corresponding to the second eigenvalue.

Relaxing RatioCut (cont.)

- Using properties of the Rayleigh quotient, it is not hard to show that the solution of

$$\min_{f \in \mathbb{R}^n} f^T L f$$
$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}.$$

is an eigenvector of L corresponding to the second eigenvalue.

- Clearly, if $\tilde{f}$ is the solution of the problem, then

$$\tilde{f}^T L \tilde{f} \leq \min_{A \subset V} \text{RatioCut}(A, \overline{A}).$$

Relaxing RatioCut (cont.)

- Using properties of the Rayleigh quotient, it is not hard to show that the solution of

$$\min_{f \in \mathbb{R}^n} f^T L f$$
$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}.$$

is an eigenvector of L corresponding to the second eigenvalue.

- Clearly, if $\tilde{f}$ is the solution of the problem, then

$$\tilde{f}^T L \tilde{f} \leq \min_{A \subset V} \text{RatioCut}(A, \overline{A}).$$

- How do we get the clusters from $\tilde{f}$?
- We could set

$$\begin{cases} v_i \in A & \text{if } f_i \geq 0 \\ v_i \in \overline{A} & \text{if } f_i < 0. \end{cases}$$

Relaxing RatioCut (cont.)

- Using properties of the Rayleigh quotient, it is not hard to show that the solution of

$$\min_{f\in\mathbb{R}^n} f^T L f$$
$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}.$$

is an eigenvector of L corresponding to the second eigenvalue.

- Clearly, if $\tilde{f}$ is the solution of the problem, then

$$\tilde{f}^T L \tilde{f} \leq \min_{A\subset V} \mathrm{RatioCut}(A, \overline{A}).$$

- How do we get the clusters from $\tilde{f}$?
- We could set

$$\begin{cases} v_i \in A & \text{if } f_i \geq 0 \\ v_i \in \overline{A} & \text{if } f_i < 0. \end{cases}$$

- More generally, we *cluster* the coordinates of f using K-means.

Relaxing RatioCut (cont.)

- Using properties of the Rayleigh quotient, it is not hard to show that the solution of

$$\min_{f \in \mathbb{R}^n} f^T L f$$
$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}.$$

is an eigenvector of L corresponding to the second eigenvalue.

- Clearly, if $\tilde{f}$ is the solution of the problem, then

$$\tilde{f}^T L \tilde{f} \leq \min_{A \subset V} \text{RatioCut}(A, \overline{A}).$$

- How do we get the clusters from $\tilde{f}$?
- We could set

$$\begin{cases} v_i \in A & \text{if } f_i \geq 0 \\ v_i \in \overline{A} & \text{if } f_i < 0. \end{cases}$$

- More generally, we *cluster* the coordinates of f using K-means.

This is the **unnormalized spectral clustering algorithm** for $k = 2$.

Relaxing RatioCut (cont.)

- Using properties of the Rayleigh quotient, it is not hard to show that the solution of

$$\min_{f \in \mathbb{R}^n} f^T L f$$
$$\text{subject to } f \perp \mathbb{1}, \|f\| = \sqrt{n}.$$

is an eigenvector of L corresponding to the second eigenvalue.

- Clearly, if $\tilde{f}$ is the solution of the problem, then

$$\tilde{f}^T L \tilde{f} \leq \min_{A \subset V} \text{RatioCut}(A, \overline{A}).$$

- How do we get the clusters from $\tilde{f}$?
- We could set

$$\begin{cases} v_i \in A & \text{if } f_i \geq 0 \\ v_i \in \overline{A} & \text{if } f_i < 0. \end{cases}$$

- More generally, we *cluster* the coordinates of f using K-means.

  This is the **unnormalized spectral clustering algorithm** for $k = 2$.

- The above process can be generalized to $k \geq 2$ clusters.

Unnormalized spectral clustering: summary

The **unnormalized spectral clustering algorithm**:

Unnormalized spectral clustering

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $u_1, \dots, u_k$ of L.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \dots, u_k$ as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \dots, C_k$.

Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

Source: von Luxburg, 2007.

Normalized spectral clustering

- Relaxing the RatioCut leads to unnormalized spectral clustering.

Normalized spectral clustering

- Relaxing the RatioCut leads to unnormalized spectral clustering.
- By relaxing the Ncut problem, we obtain the **Normalized spectral clustering** algorithm of Shi and Malik (2000).

Normalized spectral clustering

- Relaxing the RatioCut leads to unnormalized spectral clustering.
- By relaxing the Ncut problem, we obtain the **Normalized spectral clustering** algorithm of Shi and Malik (2000).

Normalized spectral clustering according to Shi and Malik (2000)

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k generalized eigenvectors $u_1, \dots, u_k$ of the generalized eigenproblem $Lu = \lambda Du$.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \dots, u_k$ as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \dots, C_k$.

Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

Source: von Luxburg, 2007.

- Relaxing the RatioCut leads to unnormalized spectral clustering.
- By relaxing the Ncut problem, we obtain the **Normalized spectral clustering** algorithm of Shi and Malik (2000).

Normalized spectral clustering according to Shi and Malik (2000)

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k generalized eigenvectors $u_1, \dots, u_k$ of the generalized eigenproblem $Lu = \lambda Du$.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \dots, u_k$ as columns.
- For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \dots, C_k$.

Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

Source: von Luxburg, 2007.

- Note: The solutions of $Lu = \lambda Du$ are the eigenvectors of L_{rw}. See von Luxburg (2007) for details.

The normalized clustering algorithm of Ng et al.

- Another popular variant of the spectral clustering algorithm was provided by Ng, Jordan, and Weiss (2002).
- The algorithm uses L_{sym} instead of L (unnormalized clustering) or L_{rw} (Shi and Malik's normalized clustering).

The normalized clustering algorithm of Ng et al.

- Another popular variant of the spectral clustering algorithm was provided by Ng, Jordan, and Weiss (2002).
- The algorithm uses L_{sym} instead of L (unnormalized clustering) or L_{rw} (Shi and Malik's normalized clustering).

Normalized spectral clustering according to Ng, Jordan, and Weiss (2002)

Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the normalized Laplacian L_{sym}.
- **Compute the first k eigenvectors $u_1, \ldots, u_k$ of L_{sym}.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
- **Form the matrix $T \in \mathbb{R}^{n \times k}$ from U by normalizing the rows to norm 1,** that is set $t_{ij} = u_{ij} / (\sum_k u_{ik}^2)^{1/2}$.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of T.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$.

Source: von Luxburg, 2007.

See von Luxburg (2007) for details.