Spectral Clustering: Preliminaries, Algorithms, and Fairness

Junhui Shen

June 1, 2022

Abstract

Spectral Clustering is one of the most successful clustering methods and has gain great attention in the study of machine learning. Clustering describes the behaviors of grouping data points so that those in the same cluster are more similar to each other than to those in different clusters. The process using the eigenpairs of the Laplacian matrix of a given graph to cluster is called spectral clustering.

This undergraduate thesis contains three parts. In the first part, we get familiar with the preliminary knowledge such as the concepts of the Laplacian matrix. In the second part, we study the algorithms of unnormalized and normalized spectral clustering and realize the advantages of normalized spectral clustering over unnormalized spectral clustering. Finally, we understand different versions of fairness along with the theorems and examples and then try to incorporate the fairness constraints into spectral clustering.

Contents

Part I
Preliminaries

In the first part, we present the preliminary knowledge of spectral clustering. We review different matrices in the graph theory and introduce the concepts of the Laplacian matrix and its properties. Then we show how the second smallest eigenvalue of the Laplacian matrix, namely the Fielder value, is related to the connectivity of the graph.

Following this, we define the density of a cuts and find out the relationship between the minimum density of all the cuts and the Fielder value of the graph. After that, we give an algorithm using the density of a cut to bi-partition a graph.

1 Graph and Laplacian Matrix

In the section, we would like to study the degree matrix, adjacency matrix, and incidence matrix of an undirected graph. Then we show the definition and some basic properties of the Laplacian matrix.

We use $G = (V, E)$ to represent an unweighted and undirected graph, where V is the set of vertices with $|V| = n$ and $E \subseteq V \times V$ is the set of edges with $|E| = m$.

Recognition: The definitions of the graph theory mainly come from chapter 10 "Graphs" of [8]. The properties of the Laplacian matrix mainly come from [10].

1.1 Graph Theory

Definition 1.1. *The matrix $A \in \mathbb{R}^{n\times n}$ is the **adjacency matrix** of G:*

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Definition 1.2. *Let $e_l = (v_i, v_j) \in E$ where $i < j$. The matrix $B \in \mathbb{R}^{n\times m}$ is the **incidence matrix** of G:*

$$b_{kl} = \begin{cases} 1 & \text{if } v_k = v_i \\ -1 & \text{if } v_k = v_j \\ 0 & \text{otherwise} \end{cases}$$

Definition 1.3. *For every vertex $v_i \in V$, the **degree** of v_i is the number of incident edges. That is,*

$$d_i = |\{vj \in V | (v_i, v_j) \in E\}|$$

*The matrix $D \in \mathbb{R}^{n\times n}$ is the **degree matrix** of G:*

$$D = diag(d_1, \ldots, d_n)$$

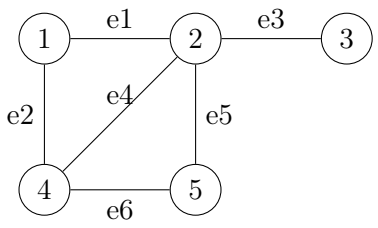

Figure 1: Graph $G = (V, E)$

Example 1.1. We can find the degree matrix, adjacency matrix, incidence matrix in the graph $G = (V, E)$ in Figure 1. The degree of every vertex is

$$d_1 = 2, \quad d_2 = 4, \quad d_3 = 1, \quad d_4 = 3, \quad d_5 = 2.$$

Thus, the degree matrix of G is:

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

The edges of G are:

$$e_1 = (v_1, v_2), e_2 = (v_1, v_4), e_3 = (v_2, v_3), e_4 = (v_2, v_4), e_5 = (v_2, v_5), e_6 = (v_4, v_5)$$

Then the adjacency matrix of G is:

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

The incidence matrix of G is

$$B = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}$$

where the columns of B follow the order of e_1, e_2, e_3, e_4, e_5, and e_6.

1.2 Laplacian Matrix

Definition 1.4. *The **Laplacian matrix** of $G = (V, E)$ is defined by*

$$L = D - A \in \mathbb{R}^{n \times n}$$

Definition 1.5. *A symmetric matrix $M \in \mathbb{R}^{n \times n}$ is **positive semidefinite** (PSD) if any of the following conditions is true:*

- $\mathbf{x}^T M\mathbf{x} \geq 0$ *for every* $\mathbf{x} \in \mathbb{R}^n$*. The expression is called the quadratic form of* M*.*
- *All eigenvalues of* M *are nonnegative.*
- *There exists a matrix* U *such that* $M = UU^T$*.*

Theorem 1.1. *L is symmetric positive definite. Specifically, If $\mathbf{x} = (x_1, \ldots, x_n)^T$, then*

$$\mathbf{x}^T L\mathbf{x} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2$$

Proof. $L = D^T - A^T = D - A = L$, so L is symmetric.
Note that

$$d_i = \sum_{j=1}^{n} a_{ij} = \sum_{j=1}^{n} 1$$

We have

$$\begin{aligned}
\mathbf{x}^T L\mathbf{x} &= \mathbf{x}^T(D - A)\mathbf{x} \\
&= \sum_{i=1}^{n} d_i x_i^2 - \sum_{i=1}^{n}\sum_{j=1}^{n} x_i a_{ij} x_j \\
&= \frac{1}{2}(\sum_{i=1}^{n} d_i x_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} x_i a_{ij} x_j + \sum_{j=1}^{n} d_j x_j^2) \\
&= \frac{1}{2}(\sum_{i=1}^{n}\sum_{j=1}^{n} x_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j + \sum_{i=1}^{n}\sum_{j=1}^{n} x_j^2) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i^2 + 2x_i x_j + x_j^2) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2
\end{aligned}$$

□

Theorem 1.2. *$\lambda_1 = 0$ is the smallest eigenvalue of L and the corresponding eigenvector $\mathbf{v}_1 = \mathbf{1}_n$ where $\mathbf{1}_n \in \mathbb{R}^n$ is an all-ones vector.*

Proof. By Definition 1.1 and Definition 1.3, for every vertex $v_i \in V$, $\sum_{j=1}^{n} a_{ij} = d_i$. Then $D\mathbf{1}_n = A\mathbf{1}_n$.
Thus, we have

$$\begin{aligned}
L\mathbf{1}_n &= (D - A)\mathbf{1}_n \\
&= D\mathbf{1}_n - A\mathbf{1}_n \\
&= D\mathbf{1}_n - D\mathbf{1}_n \\
&= \mathbf{0}_n \\
&= 0 \cdot \mathbf{1}_n
\end{aligned}$$

Therefore, $\lambda = 0$ is the smallest eigenvalue, and the corresponding eigenvectors is $\mathbf{v}_1 = \mathbf{1}_n$. □

Theorem 1.3.

$$BB^T = L$$

Proof. L is symmetric since both D and A are symmetric.
Let B be the incidence matrix of G. Then the (ij) entry of BB^T is:

$$(BB^T)_{(ij)} = \sum_{e \in E} b_{ie} b_{ej}^T = \sum_{e \in E} b_{ie} b_{je}$$

Suppose $i = j$. Then $b_{ie}b_{ie} = 1$ if e is incident to vertex v_i and 0 otherwise.
Suppose $i \neq j$. If $e = (v_i, v_j) \in E$, we get $b_{ie}b_{je} = (1)(-1) = -1$. We get 0 otherwise. Therefore,

$$(BB^T)_{ij} = \begin{cases} d_i & \text{if i =j} \\ -1 & i \neq j, (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

By Definition 1.1 and Definition 1.3, since $L = D - A$, we have

$$l_{ij} = \begin{cases} d_i & \text{if i =j} \\ -1 & i \neq j, \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Therefore, for every $i, j \in \{1, \ldots, n\}$, $(BB^T)_{ij} = L_{ij}$.
Therefore, $(BB^T) = L$.

□

Example 1.2. We continue to study the graph $G = (V, E)$ in Figure 1.
In Example 1.1, we find out D and A of G. We can write the Laplacian matrix.

$$L = D - A = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

Since $L = L^T$, L is symmetric.
Then we see that L is positive semidefinite using three ways.
First, we check that Theorem 1.1 is satisfied. Let $\mathbf{x} \in \mathbb{R}^5$. Then

$$\begin{aligned} \mathbf{x}^T L \mathbf{x} &= \begin{bmatrix} a & b & c & d & e \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & -1 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} \\ &= 2a^2 + 4b^2 + c^2 + 3d^2 + 2e^2 - 2ab - 2ad - 2bc - 2bd - 2be - 2de \\ &= (a-b)^2 + (a-d)^2 + (b-c)^2 + (b-d)^2 + (b-e)^2 + (d-e)^2 \\ &\geq 0 \end{aligned}$$

Secondly, we see that Theorem 1.2 is satisfied. The eigenvalues of L are

$$\lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 2, \lambda_4 = 4, \lambda_5 = 5$$

with the corresponding eigenvectors

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \mathbf{v}_2 = \begin{bmatrix} 1 \\ 0 \\ -3 \\ 1 \\ 1 \end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \mathbf{v}_4 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -2 \\ 1 \end{bmatrix}, \mathbf{v}_5 = \begin{bmatrix} 1 \\ -4 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Lastly, we verify Theorem 1.3. Since we already get B in Example 1.1, we have

$$BB^T = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix} = L$$

Therefore, L is positive semidefinite.

Definition 1.6. *A square matrix $A \in \mathbb{R}^{n \times n}$ is **diagonally dominant** if for every $i = 1, 2, \ldots, n$, $|a_{ii}| \geq \sum_{j=1, i \neq j}^{n} |a_{ij}|$.*

Lemma 1.1. *L is diagonally dominant.*

Proof. By Definition 1.1 and Definition 1.3, we have

$$\sum_{j=1}^{n} a_{ij} = d_i$$

Since $L = D - A$, we have: for $i = 1, 2, \ldots, n$,

$$\sum_{j=1, i \neq j}^{n} |l_{ij}| = |l_{ii}|$$

Therefore, L is diagonally dominant. □

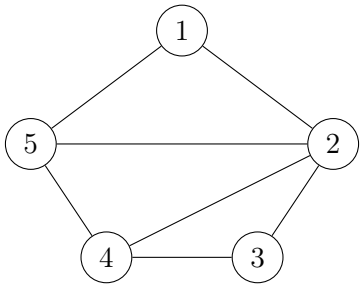

Figure 2: Graph $G = (V, E)$

Example 1.3. The graph $G = (V, E)$ in Figure 2 has the adjacency matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

We want to confirm that:

- L is diagonally dominant. (Lemma 1.1)
- $\lambda_1 = 0$ is the smallest eigenvalue of L and the corresponding eigenvector $\mathbf{v}_1$ is a multiple of $\mathbf{1}$ (Theorem 1.2)

The MATLAB code

```
format short
A = [0 1 0 0 1;1 0 1 1 1;0 1 0 1 0;0 1 1 0 1;1 1 0 1 0];
D = diag(sum(A,2));
L = D - A
[v,e] = eig(L)
```

prints the following

```
L =
     2    -1     0     0    -1
    -1     4    -1    -1    -1
     0    -1     2    -1     0
     0    -1    -1     3    -1
    -1    -1     0    -1     3
v =
   -0.4472   -0.6533   -0.5000    0.2706   -0.2236
   -0.4472    0.0000   -0.0000   -0.0000    0.8944
   -0.4472    0.6533   -0.5000   -0.2706   -0.2236
   -0.4472    0.2706    0.5000    0.6533   -0.2236
   -0.4472   -0.2706    0.5000   -0.6533   -0.2236
e =
   -0.0000         0         0         0         0
         0    1.5858         0         0         0
         0         0    3.0000         0         0
         0         0         0    4.4142         0
         0         0         0         0    5.0000
```

Therefore, the results are verified.

2 Connectivity

In this section, we see how the eigenvalues of the Laplacian matrix indicate the connectivity of the graph.

Recognition: The definitions and results mainly come from [10]. The DFS algorithms comes from [11].

2.1 Connected Graph and Connected Components

Definition 2.1. *An undirected graph is **connected** if there is a path to travel between every pair of distinct vertices. An undirected graph that is not connected is call a **disconnected** graph.*

Definition 2.2. *A **connected component** of an undirected graph is a sub-graph where each pair of two vertices is connected through a path.*

Algorithm. ([11]) The Depth-First Search (DFS) algorithm of an undirected graph $G = (V, E)$ can be used to identify the connected components of G. The DFS contains as many trees as G has connected components. Each vertex $v \in V$ is assigned an integer label $cc[v]$ between 1 and k where k is the number of connected components of G such that $cc[u] == cc[v]$ if and only if u and v are in the same connected component.

Algorithm 1 DFS

Input: Graph G
Output: $cc[u]$ for each $u \in V$

```
1: for each u ∈ V do
2:    color[u] = white
3: end for
4: cc is a new array
5: time = 0
6: k = 0
7: for each vertex u ∈ V do
8:    if color[u] = white then
9:       k = k+1
10:      cc[u] = k
11:      DFS_Visit(u)
12:   end if
13: end for
```

Algorithm 2 DFS_Visit

Input: Vertex u
Output: $cc[u]$

```
1: color[u] = gray
2: time = time +1
3: d[u] = time
4: for each v ∈ adj[u] do
5:    cc[v] = cc[u]
6:    if color[v] = white then
7:       DFS_Visit(v)
8:    end if
9: end for
10: color[u] = black
11: time = time + 1
12: f[u] = time
```

2.2 Relationship between Laplacian Matrix and Connectivity

Definition 2.3. *The second smallest eigenvalue λ_2 of L is known as* ***the Fielder value of*** *G.*

Remark 2.1. *By Theorem 1.2, the Fielder value of G is always nonnegative.*

Theorem 2.1. *If $G = (V, E)$ is disconnected, then L is a block diagonal matrix where each block corresponds to a connected component of G.*

Proof. We run the DFS so that we find the connected components of G and determine the total number of them.
Then $G = G_1 \cup G_2 \cdots \cup G_k$ where $G_i, i = \{1, 2 \dots k\}$ is a nonempty connected component. For each G_i, there exists a corresponding adjacency matrix A_i.
Then

$$A = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_k \end{bmatrix}$$

There is no edge between each of the connected components, so values outside $A_i, i = \{1, 2 \dots k\}$ are all 0.
In addition, the degree matrix D is diagonal where each connected component has a corresponding degree matrix D_i on it.
Hence, $L = A - D$ is a block diagonal matrix.

$$L = \begin{bmatrix} A_1 - D_1 & & & \\ & A_2 - D_2 & & \\ & & \ddots & \\ & & & A_k - L_k \end{bmatrix} = \begin{bmatrix} L_1 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_k \end{bmatrix}$$

Each diagonal block of L, namely L_i, corresponds to a connected component. Then each L_i has an eigenvalue 0 with eigenvector $\mathbf{1}$. Since there are k connected components, the multiplicity of eigenvalues of L would also be k such that $0 = \lambda_1 = \lambda_2 = \cdots = \lambda_k \leq \cdots \leq \lambda_n$. □

Lemma 2.1. *Let $G = (V, E)$ be a graph with Laplacian matrix L whose eigenvalue are $0 = \lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$. Then G is connected if and only if $\lambda_2 > 0$.*

Proof. $\Longleftarrow$: $\lambda_2 > 0$ implies that G is connected.
Suppose G has $k \geq 2$ connected components. Then L is a block diagonal matrix, where each block corresponds to the Laplacian matrix for each component. By stacking 0s and the $\mathbf{1}$ eigenvector, we can build k linearly independent eigenvectors of L with eigenvalue 0. Hence, the multiplicity of 0 is k.
Then $\lambda_2 = 0$ implies that G is disconnected. $\lambda_2 > 0$ implies that there is only one connected component of G.

$\Longrightarrow$: G is connected implies $\lambda_2 > 0$.
Let G be a connected graph. Let $\mathbf{u}$ be an eigenvector of L with the eigenvalue 0. Then $\mathbf{u} \neq \mathbf{0}$. By Theorem 1.1, $\mathbf{u}^T L\mathbf{u} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (u_i - u_j)^2$. Since $L\mathbf{u} = \mathbf{0}$, we get $0 = \mathbf{u}^T L\mathbf{u} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (u_i - u_j)^2$.
This implies that $u_i = u_j$ for every $(v_i, v_j) \in E$. Since G is connected, there is a way to walk from v_1 to any other vertex v_j along edges of G. If $u_i = c, c \in \mathbb{R}$, then $u_k = c$ because we can always find

a path from vertex v_i to vertex v_k. Then $\mathbf{u} = c\mathbf{1}$. Since $\mathbf{u} \neq \mathbf{0}$, $u_1 \neq 0$ and every eigenvector of 0 is a multiple of $\mathbf{1}$. Then the multiplicity of eigenvalue 0 is 1. Additionally, $\lambda_2 \neq \lambda_1$ where $\lambda_1 = 0$. Since L is positive semidefinite, all its eigenvalues are nonnegative. We get that $\lambda_2 > 0$. □

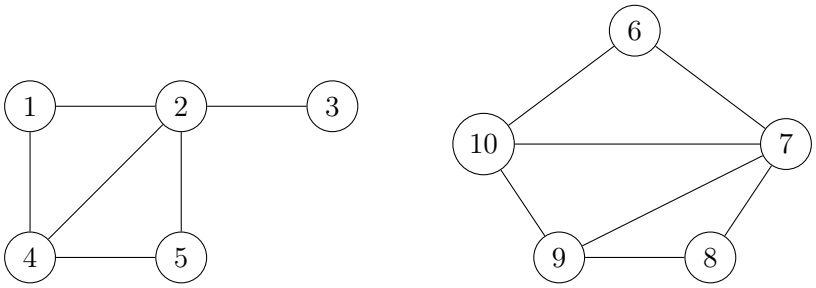

Figure 3: Graph $G = (V, E)$

Example 2.1. The graph $G = (V, E)$ in Figure 3 is a disconnected graph with 2 connected components. The left-hand component corresponds to Example 1.2, and the right-hand connected component corresponds to Example 1.3. We have found out the Laplacian matrix and the corresponding eigenvalues and eigenvectors of each connected component. This time we want to confirm that L is a block diagonal matrix and that the Fielder value of G is 0.
The MATLAB code

```
format short
A1 = [0 1 0 1 0;1 0 1 1 1;0 1 0 0 0;1 1 0 0 1;0 1 0 1 0];
A2 = [0 1 0 0 1;1 0 1 1 1;0 1 0 1 0;0 1 1 0 1;1 1 0 1 0];
A = blkdiag(A1,A2);
D = diag(sum(A,2));
L = D - A
e = eig(L)
```

printed the following

```
L =

     2    -1     0    -1     0     0     0     0     0     0
    -1     4    -1    -1    -1     0     0     0     0     0
     0    -1     1     0     0     0     0     0     0     0
    -1    -1     0     3    -1     0     0     0     0     0
     0    -1     0    -1     2     0     0     0     0     0
     0     0     0     0     0     2    -1     0     0    -1
     0     0     0     0     0    -1     4    -1    -1    -1
     0     0     0     0     0     0    -1     2    -1     0
     0     0     0     0     0     0    -1    -1     3    -1
     0     0     0     0     0    -1    -1     0    -1     3


e =
```

```
-0.0000
-0.0000
 1.0000
 1.5858
 2.0000
 3.0000
 4.0000
 4.4142
 5.0000
 5.0000
```

Therefore, L is a block diagonal matrix with eigenvalues $\lambda_1 = \lambda_2 = 0$.

3 Relationship between Laplacian Matrix and Sparsest Cut

In this section, we give the definition of the density of a cut and relate the Fielder value to the minimum density of all the cuts. After that, we introduce an algorithm so that we can break a graph into two clusters.

Recognition: The definitions and results mainly come from [11].

Definition 3.1. *Let $A \subset V \setminus \emptyset$ be a subset of vertices and $\bar{A} = V \setminus A$. A **cut induced by A** is a partition of V into two sets: A and $\bar{A}$.*

Definition 3.2. *The **density** of a cut induced by A is*

$$\phi(A, \bar{A}) = n \cdot \frac{|E(A, \bar{A})|}{|A| \cdot |\bar{A}|}$$

where $E(A, \bar{A}) = \{(v_i, v_j) \mid v_i \in A, v_j \in \bar{A}\}$.

Definition 3.3. *Let $\phi_G = \min_{A \subset V \setminus \emptyset} \phi(A, \bar{A})$ denote the minimum density of all the cuts. A cut with density ϕ_G is called the **sparsest cut**.*

Definition 3.4. *For any subset $A \subset V \setminus \emptyset$, $\mathbf{c}_A \in \mathbb{R}^n$ is a **characteristic vector of** A:*

$$\mathbf{c}_{A_i} = \begin{cases} 1, & \text{if } v_i \in A \\ 0, & \text{otherwise} \end{cases}$$

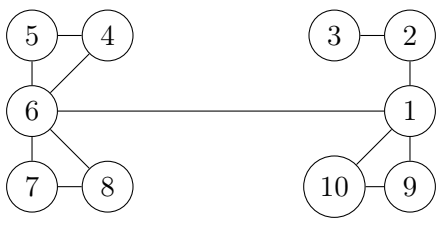

Figure 4: Graph $G = (V, E)$

Example 3.1. *Consider the graph in Figure 4. Suppose we choose* $A = \{v_1, v_3, v_5, v_7, v_9\}$.
The characteristic vector is $\mathbf{c}_A = \begin{bmatrix}1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0\end{bmatrix}$.
Then $E(A, \bar{A}) = \{(v_1, v_2), (v_2, v_3), (v_1, v_{10}), (v_9, v_{10}), (v_1, v_6), (v_4, v_5), (v_5, v_6), (v_6, v_7), (v_7, v_8)\}$.
The density of this cut is

$$\phi(A, \bar{A}) = n \cdot \frac{|E(A, \bar{A})|}{|A| \cdot |\bar{A}|} = 10 \cdot \frac{9}{5 \cdot 5} = 10 \cdot \frac{9}{25} = \frac{18}{5}$$

In comparison, we can choose $A = \{v_4, v_5, v_6, v_7, v_8\}$.
The characteristic vector is $\mathbf{c}_A = \begin{bmatrix}1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1\end{bmatrix}$.
Then $E(A, \bar{A}) = \{(v_1, v_6)\}$.
The density of all the cut is

$$\phi_G = \phi(A, \bar{A}) = n \cdot \frac{|E(A, \bar{A})|}{|A| \cdot |\bar{A}|} = 10 \cdot \frac{1}{5 \cdot 5} = \frac{2}{5}$$

The density using $A = \{v_4, v_5, v_6, v_7, v_8\}$ *is much smaller than using* $A = \{v_1, v_3, v_5, v_7, v_9\}$.

Definition 3.5. *A vector* $\mathbf{x} \in \mathbb{R}^n$ *is* ***non-constant*** *if it is not a multiple of* $\mathbf{1}$*, i.e.,* $\mathbf{x} \neq c\mathbf{1}, c \in \mathbb{R}$.

For $\mathbf{x} = (x_1, \ldots, x_n)$, we define the function

$$Q(x) = n \cdot \frac{\sum\limits_{(v_i, v_j) \in E} (x_i - x_j)^2}{\sum\limits_{1 \leq i < j \leq n} (x_i - x_j)^2}$$

Remark 3.1. *In Theorem 1.1, we show that* $\mathbf{x}^T L \mathbf{x} = \frac{1}{2} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} (x_i - x_j)^2$.
Then

$$n \cdot \sum_{(v_i, v_j) \in E} (x_i - x_j)^2 = n \cdot \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_j)^2 = n \cdot \mathbf{x}^T L \mathbf{x}$$

Also, $\sum\limits_{1 \leq i < j \leq n} (x_i - x_j)^2 = \mathbf{x}^T L_{K_n} \mathbf{x}$ *where* L_{K_n} *is the Laplacian matrix of the complete graph with* k *vertices.*

Since we can express every cut of A *in the form of* $\mathbf{c}_A$, $\phi_G = \min\{Q(\mathbf{x}) \mid \mathbf{x} \in \{\mathbf{c}_A \mid A \subset V \setminus \emptyset\}$.

If we try to to find ϕ_G of a graph by computing $\phi_G = \min\{Q(\mathbf{x}) \mid \mathbf{x} \in \{\mathbf{c}_A \mid A \subset V \setminus \emptyset\}\}$, we need to check $2^n - 2$ $\mathbf{c}_A$'s where $|V| = n$.

For instance, to find ϕ_G in Example 3.1, , we need to compute $2^{10} - 2 = 1022$ $\mathbf{c}'_A s$. It is not a practical way to find ϕ_G.

Thus, we need to relax the problem. We instead minimize $Q(x)$ over all non-constant vectors $\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq c\mathbf{1}$. Let $\mu = \min\{Q(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq c \cdot \mathbf{1}\}$.

The procedure is called a relaxation as we allow more vectors to be plugged in, namely from the characteristic vectors $\mathbf{c}_A$ to all non-constant vectors.

We show that $\phi_G \leq \lambda_2$ where λ_2 is the Fielder value of G. The idea is that we first show μ is no greater than ϕ_G, and then we show that μ is equivalent to the Fielder value.

Lemma 3.1. *Let* $\mu = \min\{Q(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq c \cdot \mathbf{1}\}$. *Then* $\mu \leq \phi_G$.

Proof. Since we are doing a relaxation on the allowable vectors, we minimize over a larger set than what we are supposed to. Then

$$\mu = \min\{Q(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq c \cdot \mathbf{1}\} \leq \min\{Q(\mathbf{x}) \mid \mathbf{x} \in \{\mathbf{c}_A | A \subset V \setminus \emptyset\}\} = \phi_G$$

□

Lemma 3.2. *$Q(\mathbf{x}) = Q(\mathbf{x} + t\mathbf{1})$ for every $t \in \mathbb{R}$.*

Proof.

$$\begin{aligned} Q(\mathbf{x}) &= n \cdot \frac{\sum\limits_{(v_i,v_j)\in E} (x_i - x_j)^2}{\sum\limits_{1\leq i<j\leq n} (x_i - x_j)^2} \\ &= n \cdot \frac{\sum\limits_{(v_i,v_j)\in E} [(x_i + t) - (x_j + t)]^2}{\sum\limits_{1\leq i<j\leq n} [(x_i + t) - (x_j + t)]^2} \\ &= Q(\mathbf{x} + t\mathbf{1}) \end{aligned}$$

Thus, $Q(\mathbf{x}) = Q(\mathbf{x} + t\mathbf{1})$ for every $t \in \mathbb{R}$. □

Lemma 3.3. $\mu = \min\{Q(x) \mid x \in \mathbb{R}^n \setminus \{\mathbf{0}\}, \mathbf{1}^T\mathbf{x} = 0\}$

Proof. Let $\mathbf{x} \in \mathbb{R}^n$ be arbitrary. We can write it as $\mathbf{x} = c\mathbf{1} + \mathbf{1}^\perp, c \in \mathbb{R}$. Then $Q(x) = Q(c\mathbf{1} + \mathbf{1}^\perp) = Q(\mathbf{1}^\perp)$ because $Q(\mathbf{x}) = Q(\mathbf{x} + t\mathbf{1})$.
Thus, $\mu = \min\{Q(x) \mid x \in \mathbb{R}^n \setminus \{\mathbf{0}\}, \mathbf{1}^T\mathbf{x} = 0\}$. □

We see another way to approach μ above. Instead of checking all non-constant vectors, we can reduce the range of the vectors that we look for. Next, we seek for a more efficient way so that we can get μ even without knowing $Q(x)$.

Lemma 3.4. *The Laplacian matrix of a complete graph L_{K_n} is:*

$$L_{K_n} = nI_n - J_n$$

where $J_n = \mathbf{1} \cdot \mathbf{1}^T \in \mathbb{R}^{n\times n}$.

Proof. For a complete graph, the Laplacian matrix is

$$D_{K_n} = \begin{bmatrix} n-1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & n-1 \end{bmatrix}, A_{K_n} = \begin{bmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \dots & 0 & 1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix}$$

$$L_{K_n} = \begin{bmatrix} n-1 & -1 & -1 & \dots & -1 \\ -1 & n-1 & -1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -1 & \dots & n-1 & -1 \\ -1 & -1 & \dots & -1 & n-1 \end{bmatrix}$$

Also , we have

$$nI_n - J_n = \begin{bmatrix} n & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & n \end{bmatrix} - \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} n-1 & -1 & -1 & \dots & -1 \\ -1 & n-1 & -1 & \dots & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -1 & \dots & n-1 & -1 \\ -1 & -1 & \dots & -1 & n-1 \end{bmatrix}$$

Thus, $L_{K_n} = nI_n - J_n$. □

Corollary 3.1. $J_n\mathbf{x} = \mathbf{0}$.

Proof. Since $\mathbf{1}^T\mathbf{x} = 0$, we get

$$J_n\mathbf{x} = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \mathbf{1}^T\mathbf{x} \\ \vdots \\ \mathbf{1}^T\mathbf{x} \end{bmatrix} = \mathbf{0}$$

□

Corollary 3.2. $\mathbf{x}^T L_{K_n}\mathbf{x} = n\|\mathbf{x}\|^2$

Proof. By Lemma 3.4, $L_{K_n} = nI_n - J_n$. By Corollary 3.1, $J_n\mathbf{x} = \mathbf{0}$. Then

$$\begin{aligned} \mathbf{x}^T L_{K_n}\mathbf{x} &= \mathbf{x}^T(nI_n - J_n)\mathbf{x} \\ &= \mathbf{x}^T nI_n\mathbf{x} - \mathbf{x}^T J_n\mathbf{x} \\ &= n\mathbf{x}^T\mathbf{x} - \mathbf{x}^T J_n\mathbf{x} \\ &= n\mathbf{x}^T\mathbf{x} \\ &= n\|\mathbf{x}\|^2 \end{aligned}$$

□

Corollary 3.3. $Q(\alpha\mathbf{x}) = Q(\mathbf{x})$ *where* $\alpha \in \mathbb{R} \setminus \{0\}$.

Proof.

$$\begin{aligned} Q(\alpha\mathbf{x}) &= n \cdot \frac{\alpha\mathbf{x}^T L\alpha\mathbf{x}}{\alpha\mathbf{x}^T L_{K_n}\alpha\mathbf{x}} \\ &= n \cdot \frac{\alpha^2\mathbf{x}^T L\mathbf{x}}{\alpha^2\mathbf{x}^T L_{K_n}\mathbf{x}} \\ &= n \cdot \frac{\mathbf{x}^T L\mathbf{x}}{\mathbf{x}^T L_{K_n}\mathbf{x}} \\ &= Q(x) \end{aligned}$$

□

Lemma 3.5. $\mu = \min\{\mathbf{x}^T L\mathbf{x} \mid \|\mathbf{x}\| = 1, \mathbf{1}^T\mathbf{x} = 0\}$.

Proof. By Corollary 3.2, $\mathbf{x}^T L_{K_n} \mathbf{x} = n \|x\|^2$. Then

$$Q(x) = n \cdot \frac{\sum\limits_{(v_i,v_j)\in E} (x_i - x_j)^2}{\sum\limits_{1\le i<j\le n} (x_i - x_j)^2} = n \cdot \frac{\mathbf{x}^T L \mathbf{x}}{\mathbf{x}^T L_{K_n} \mathbf{x}} = n \cdot \frac{\mathbf{x}^T L \mathbf{x}}{n \|\mathbf{x}\|^2} = \frac{\mathbf{x}^T L \mathbf{x}}{\|\mathbf{x}\|^2}$$

Then

$$\mu = \min\{Q(x) \mid x \in \mathbb{R}^n \setminus \{\mathbf{0}\}, \mathbf{1}^T \mathbf{x} = 0\} = \min\{\frac{\mathbf{x}^T L \mathbf{x}}{\|\mathbf{x}\|^2} \mid x \in \mathbb{R}^n \setminus \{\mathbf{0}\}, \mathbf{1}^T \mathbf{x} = 0\}$$

By Corollary 3.3, $Q(\alpha \mathbf{x}) = Q(\mathbf{x})$ where $\alpha \in \mathbb{R} \setminus \{0\}$. Clearly, $\|\mathbf{x}\|^2$ is a scalar.
Thus,

$$\mu = \min\{\mathbf{x}^T L \mathbf{x} \mid \|\mathbf{x}\| = 1, \mathbf{1}^T \mathbf{x} = 0\}$$

□

Let $\mathbf{v}_1 = \mathbf{1}, \mathbf{v}_2, \ldots, \mathbf{v}_n$ be a set of orthonormal eigenvectors of the Laplacian matrix L corresponding to the eigenvalues $0 = \lambda_1 \le \lambda_2 \le \ldots \lambda_n$. We finally show that $\mu = \lambda_2$ so that $\lambda_2 \le \phi_G$.

Lemma 3.6. *If* $\mathbf{1}^T \mathbf{x} = 0$, *then* $\mathbf{x} = \alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3 \ldots \alpha_n \mathbf{v}_n = 0$ *where* $\alpha_i \in \mathbb{R}$ *for every* $i = 1, \ldots, n$.

Proof. Let $\mathbf{1}^T \mathbf{x} = 0$. Let $\mathbf{v}_1, \ldots \mathbf{v}_n$ be a set of orthonormal eigenvectors of L corresponding to the eigenvalue $\lambda_1 = 0, \lambda_2, \ldots, \lambda_n$.
Then ${\mathbf{v}_1}^T \mathbf{v}_2 = 0, {\mathbf{v}_1}^T \mathbf{v}_3 = 0, \ldots, {\mathbf{v}_1}^T \mathbf{v}_n = 0$.
Then ${\mathbf{v}_1}^T (\alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3 \cdots + \alpha_n \mathbf{v}_n) = 0$ where $\alpha_i \in \mathbb{R}$ for every $i = 1, \ldots, n$.
Since $\mathbf{v_1} = \mathbf{1}$, we have $\mathbf{1}^T (\alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3 \cdots + \alpha_n \mathbf{v}_n) = 0$.
Thus, $\mathbf{1}^T \mathbf{x} = 0$ implies that $\mathbf{x} = \alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3 \cdots + \alpha_n \mathbf{v}_n$. □

Lemma 3.7. *Let* $\mathbf{1}^T \mathbf{x} = 0$ *and* $\|\mathbf{x}\| = 1$. *Then* $\mathbf{x} = \sum\limits_{i=2}^{n} \alpha_i \mathbf{v}_i$ *with* $\sum\limits_{i=2}^{n} \alpha_i^2 = 1$.

Proof. By Lemma 3.6, since $\mathbf{1}^T \mathbf{x} = 0$, we have $\mathbf{x} = (\alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3 \ldots \alpha_n \mathbf{v}_n) = \sum\limits_{i=2}^{n} \alpha_i \mathbf{v}_i$, $\alpha_i \in \mathbb{R}$ for every $i = 1, \ldots, n$. Then

$$\begin{aligned}
\mathbf{x} \cdot \mathbf{x} &= \|x\|^2 \\
&= (\alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3 + \cdots + \alpha_n \mathbf{v}_n)^T (\alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3 + \cdots + \alpha_n \mathbf{v}_n) \\
&= \alpha_2^2 + \cdots + \alpha_n^2 \\
&= \sum_{i=2}^{n} \alpha_i^2 \\
&= 1
\end{aligned}$$

Hence, $\sum\limits_{i=2}^{n} \alpha_i^2 = 1$.
Therefore, $\mathbf{x} = \sum\limits_{i=2}^{n} \alpha_i \mathbf{v}_i$ with $\sum\limits_{i=2}^{n} \alpha_i^2 = 1$. □

Lemma 3.8. $\mathbf{x}^T L \mathbf{x} = \alpha^2 \lambda_2 + \cdots + \alpha_n^2 \lambda_n$ *where* $\mathbf{x} = \sum\limits_{i=2}^{n} \alpha_i \mathbf{v}_i$.

Proof. Let $\mathbf{x} = \sum_{i=2}^{n} \alpha_i \mathbf{v}_i$. Note that $L\mathbf{v}_i = \lambda_i \mathbf{v}_i$ for $i = 2, 3, \ldots, n$ where $\lambda_i \in \mathbb{R}$. Also, ${\mathbf{v}_i}^T \mathbf{v}_i = 1$. Then we have:

$$\begin{aligned}
\mathbf{x}^T L\mathbf{x} &= (\alpha_2 \mathbf{v}_2^T + \alpha_3 \mathbf{v}_3^T + \ldots \alpha_n \mathbf{v}_n^T) L(\alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3 + \ldots \alpha_n \mathbf{v}_n) \\
&= \alpha_2^2 \mathbf{v}_2^T L\mathbf{v_2} + \alpha_3^2 \mathbf{v}_3^T L\mathbf{v_3} + \cdots + \alpha_n^2 \mathbf{v}_n^T L\mathbf{v_n} \\
&= \alpha_2^2 \lambda_2 \mathbf{v}_2^T \mathbf{v}_2 + \cdots + \alpha_n^2 \lambda_2 \mathbf{v}_n^T \mathbf{v}_n \\
&= \alpha^2 \lambda_2 + \cdots + \alpha_n^2 \lambda_n
\end{aligned}$$

□

Theorem 3.1. *$\mu = \lambda_2 < \phi_G$ where λ_2 is the Fielder value of G.*

Proof. Recall that by Lemma 3.5, $\mu = \min\{\mathbf{x}^T L\mathbf{x} \mid \|\mathbf{x}\| = 1, \mathbf{1}^T \mathbf{x} = 0\}$.
By Lemma 3.1, we have $\mu \leq \phi_G$, so we only need to show that $\mu = \lambda_2$.

- $\mu \geq \lambda_2$

  By Lemma 3.7, we see that $\mathbf{x} = \sum_{i=2}^{n} \alpha_i \mathbf{v}_i$ with $\sum_{i=2}^{n} \alpha_i^2 = 1$.

  By Lemma 3.8, we have $\mathbf{x}^T L\mathbf{x} = \alpha^2 \lambda_2 + \cdots + \alpha_n^2 \lambda_n$ where $\mathbf{x} = \sum_{i=2}^{n} \alpha_i \mathbf{v}_i$.

  Combining the above, since $\lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_n$, we have

  $$\mathbf{x}^T L\mathbf{x} = \alpha_2^2 \lambda_2 + \cdots + \alpha_n^2 \lambda_n \geq \lambda_2 \cdot \sum_{i=2}^{n} \alpha_i^2 = \lambda_2$$

  Thus, $\mu = \min\{\mathbf{x}^T L\mathbf{x} \mid \|\mathbf{x}\| = 1, \mathbf{1}^T \mathbf{x} = 0\} \geq \lambda_2$.

- $\mu \leq \lambda_2$

  By Lemma 3.8, we have $\mathbf{x}^T L\mathbf{x} = \alpha^2 \lambda_2 + \cdots + \alpha_n^2 \lambda_n$ where $\mathbf{x} = \sum_{i=2}^{n} \alpha_i \mathbf{v}_i$.
  We can choose $\alpha_2 = 1, \alpha_3 = 0, \ldots, \alpha_n = 0$.
  Then $\mathbf{x}^T L\mathbf{x} = \alpha_2^2 \lambda_2 = \lambda_2$.
  Since $\lambda_2 \in \{\mathbf{x}^T L\mathbf{x} | \, \|\mathbf{x}\| = 1, \mathbf{1}^T \mathbf{x} = 0\}$, we have $\mu = \min\{\mathbf{x}^T L\mathbf{x} \mid \|\mathbf{x}\| = 1, \mathbf{1}^T \mathbf{x} = 0\} \leq \lambda_2$.

Since $\mu \geq \lambda_2$ and $\mu \leq \lambda_2$, we conclude that $\mu = \lambda_2$.
Therefore, $\lambda_2 \leq \phi_G$. □

Algorithm. We introduce the algorithm to break a graph into 2 main clusters using the eigenvector corresponding to the Fielder value of G.

Algorithm 3 Clustering using the eigenvector corresponding to the Fielder value of G

Input: Graph G
Output: 2 clusters of G

1: Compute the Laplacian matrix L.
2: Compute the Fielder value of G and the corresponding eigenvector $\mathbf{w}$.
3: Sort the components of $\mathbf{w}$ in descending order so that $w_{i_1} \geq w_{i_2} \cdots \geq w_{i_n}$
4: Partition G with the cuts: $A_k = \{v_{i_1}, v_{i_2} \ldots, v_{i_k}\}$ for $k = 1, \ldots, n-1$.
5: Output A_i that induces smallest density among $A_k, k = 1, \ldots, n-1$

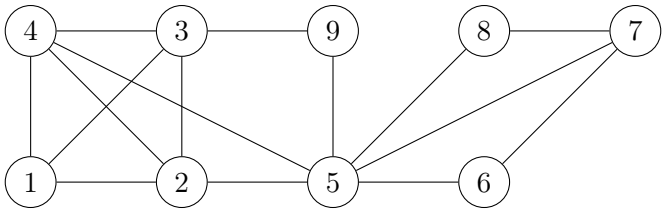

Figure 5: Graph $G = (V, E)$

Example 3.2. We use $G = (V, E)$ in Figure 5 as an example to demonstrate the algorithm. The MATLAB code

```
format short
A = [0 1 1 1 0 0 0 0 0; 1 0 1 1 1 0 0 0 0;1 1 0 1 0 0 0 0 1;
    1 1 1 0 1 0 0 0 0; 0 1 0 1 0 1 1 1 1; 0 0 0 0 1 0 1 0 0;
    0 0 0 0 1 1 0 1 0; 0 0 0 0 1 0 1 0 1;0 0 1 0 1 0 0 1 0];
D = diag(sum(A,2));
L = D - A;
[v,e] = eig(L);
w = v(:,2); %the eigenvector corresponding to the Fielder value of G
[out,index] = sort(w)
```

printed the following

```
index =

     1
     3
     4
     2
     9
     5
     8
     7
     6
```

This means that $w_6 \geq w_7 \geq w_8 \geq w_5 \geq w_9 \geq w_2 \geq w_4 \geq w_3 \geq w_1$.
Then $A_1 = \{v_6\}, A_2 = \{v_6, v_7\}, A_3 = \{v_6, v_7, v_8\}, A_4 = \{v_6, v_7, v_8, v_5\}, A_5 = \{v_6, v_7, v_8, v_5, v_9\}, A_6 = \{v_6, v_7, v_8, v_5, v_9, v_2\}, A_7 = \{v_6, v_7, v_8, v_5, v_9, v_2, v_4\}, A_8 = \{v_6, v_7, v_8, v_5, v_9, v_2, v_4, v_3\}$.
We calculate the density of every A_i and compare. Then we pick one A_i such that the density of the cut it induces is the minimum among all A_i's.

$$\phi(A_1, \bar{A}_1) = 9 \cdot \frac{2}{1 \cdot 8} = \frac{9}{4}, \phi(A_2, \bar{A}_2) = 9 \cdot \frac{3}{2 \cdot 7} = \frac{27}{14}$$

$$\phi(A_3, \bar{A}_3) = 9 \cdot \frac{3}{3 \cdot 6} = \frac{3}{2}, \phi(A_4, \bar{A}_4) = 9 \cdot \frac{3}{4 \cdot 5} = \frac{27}{20}$$

$$\phi(A_5, \bar{A}_5) = 9 \cdot \frac{3}{5 \cdot 4} = \frac{27}{20}, \phi(A_6, \bar{A}_6) = 9 \cdot \frac{5}{6 \cdot 3} = \frac{5}{2}$$

$$\phi(A_7, \bar{A}_7) = 9 \cdot \frac{4}{7 \cdot 2} = \frac{18}{7}, \phi(A_8, \bar{A}_8) = 9 \cdot \frac{3}{8 \cdot 1} = \frac{27}{8}$$

Thus, we can choose a cut between $A_4 = \{v_5, v_6, v_7, v_8\}$ and $\bar{A}_4 = \{v_1, v_2, v_3, v_4, v_9\}$ to get the relatively small density.

Part II
Spectral Clustering

Given a set of vertices in an undirected graph, we can classify them via certain features to put relatively well-connected vertices within a group and those that are not so well-connected into the rest. For instance, assume we have a social network graph with people as vertices and an edge between people who have the same living habits. Then we may cluster these people by their nationality or gender identity. The actions of such classifications are called clustering. If we use the eigenpairs of the Laplacian matrix of the graph to cluster, then the process is called spectral clustering.

There are two kinds of spectral clustering, corresponding to different types of Laplacian matrices: unnormalized spectral clustering and normalized spectral clustering. Additionally, we use ratio cut and normalized cut to convert different spectral clustering methods into standard minimization problems. We further figure out that that normalized spectral clustering using the random walk normalized Laplacian matrix is better than using the other Laplacian matrices.

4 Weighted Laplacian Matrix

At this stage, we need to extend our knowledge on the graph theory to have the prerequisite for further studying spectral clustering.

In the section, we will study concepts related to the weighted graph as well as the definition and the properties of the Laplacian matrix.

Recognition: The definitions and results on the weighted graph and the Laplacian matrix mainly come from chapter 2 of [4].

Definition 4.1. *An undirected graph $G = (V, E, W)$ is a **weighted graph** if V is the set of vertices with $|V| = n$, W is a symmetric matrix such that $w_{ij} \geq 0$ for every $i = 1, \ldots, n$ and $j = 1, \ldots, n$ and $w_{ii} = 0$ for every $i = 1, \ldots, n$. Also, $(v_i, v_j) \in E$ if and only if $w_{ij} > 0$.*

Remark 4.1. *We can consider W as a weighted adjacency matrix.*

Definition 4.2. *For every vertex $v_i \in V$ in graph $G = (V, E, W)$, the **degree** of v_i is defined by*

$$d_i = \sum_{j=1}^{n} w_{ij}$$

*The **degree matrix** $D \in \mathbb{R}^{n \times n}$ of $G = (V, E, W)$ is:*

$$D = diag(d_1, \ldots, d_n)$$

Definition 4.3. *The **Laplacian matrix** of $G = (V, E, W)$ is:*

$$L = D - W \in \mathbb{R}^{n \times n}$$

Definition 4.4. *Given any subset of nodes $A \subseteq V$, the **volume** of A is:*

$$vol(A) = \sum_{v_i \in A} d_i$$

Definition 4.5. *Given any two subset $A, B \subseteq V$, we define **links** between A and B as*

$$links(A, B) = \sum_{v_i \in A, v_j \in B} w_{ij}$$

Remark 4.2. *Since the matrix W is symmetric, $links(A, B) = links(B, A)$.*

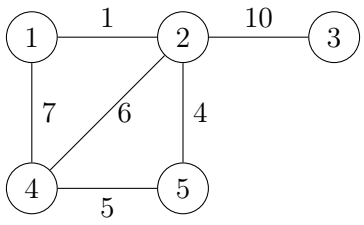

Figure 6: Graph $G = (V, E, W)$

Example 4.1. *We can find out W, D and L of the undirected but weighted graph $G = (V, E, W)$ in Figure 6. Note that if all the weights of the edges are 1, then the graph is unweighted, which is Figure 1. The weighted adjacency matrix W is*

$$W = \begin{bmatrix} 0 & 1 & 0 & 7 & 0 \\ 1 & 0 & 10 & 6 & 4 \\ 0 & 10 & 0 & 0 & 0 \\ 7 & 6 & 0 & 0 & 5 \\ 0 & 4 & 0 & 5 & 0 \end{bmatrix}$$

The degree matrix D is

$$D = \begin{bmatrix} 8 & 0 & 0 & 0 & 0 \\ 0 & 21 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 18 & 0 \\ 0 & 0 & 0 & 0 & 9 \end{bmatrix}$$

The Laplacian matrix $L = D - W$ is

$$L = \begin{bmatrix} 8 & -1 & 0 & -7 & 0 \\ -1 & 21 & -10 & -6 & -4 \\ 0 & -10 & 10 & 0 & 0 \\ -7 & -6 & 0 & 18 & -5 \\ 0 & -4 & 0 & -5 & 9 \end{bmatrix}$$

Let $A = \{v_1, v_2\}$.
Then $vol(A) = d_1 + d_2 = 8 + 21 = 29$.
Then $links(A, \bar{A}) = w_{14} + w_{24} + w_{25} + w_{23} = 7 + 6 + 4 + 10 = 21$.

Theorem 4.1. *[4] The Laplacian matrix $L = D - W$ is symmetric positive semi-definite. Specifically,*

$$\mathbf{x}^T L \mathbf{x} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (\mathbf{x}_i - \mathbf{x}_j)^2$$

Proof. Since both D and W are symmetric, $L = D - W$ is symmetric.
For all $\mathbf{x} \in \mathbb{R}^n$, we have

$$\begin{aligned}
\mathbf{x}^T L\mathbf{x} &= \mathbf{x}^T D\mathbf{x} - \mathbf{x}^T W\mathbf{x} \\
&= \sum_{i=1}^{n} d_i x_i^2 - \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j w_{ij} \\
&= \frac{1}{2}(\sum_{i=1}^{n} d_i x_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j w_{ij} + \sum_{j=1}^{n} d_j x_j^2) \\
&= \frac{1}{2}(\sum_{i=1}^{n}\sum_{j=1}^{n} x_i^2 w_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n} 2x_i x_j w_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{n} x_j^2 w_{ij}) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i^2 w_{ij} + 2x_i x_j w_{ij} + x_j^2 w_{ij}) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(x_i - x_j)^2 \geq 0
\end{aligned}$$

Therefore, L is symmetric positive semidefinite. □

5 Unnormalized Spectral Clustering

We have introduced the basics of the Laplacian matrix. However, there are two questions we may ask:

1. If the data set is so large enough that manually drawing the graph and computing the Laplacian matrix becomes complicating, then what do we do?

2. In Algorithm 3, we use the Laplacian matrix to bi-partition the graph. Is it possible to partition the graph into more clusters with the Laplacian matrix?

The goal of this section is to find ways to address the two problems. To answer the first question, we use the similarity matrix to represent the graph and get the Laplacian matrix. To answer the latter, we use a method called the spectral clustering.

Similarity Graph. Given data points $x_1, \ldots, x_n$, we want to have some measurement of similarity between every pairs of data points x_i and x_j so that we can divide the graph into several clusters so that points in the same group are similar in some sense, and dissimilar points in different groups are dissimilar.

While implementing the similarity graph, we have to make sure that the points considered as similar by the similarity function are closely related in the application of the actual data points.

There are several ways to reformulate data points into similarity graphs. Here we introduce the concept of the ϵ-neighborhood graph.

Definition 5.1. *([9]) The* ***similarity matrix*** $S \in \mathbb{R}^{n\times n}$ *induced by the ϵ-neighborhood graph is:*

$$s_{ij} = \begin{cases} 1, & \text{if } \|x_i - x_j\| < \epsilon, v_i \neq v_j \\ 0, & \text{otherwise} \end{cases}$$

Remark 5.1. *One practical problem with the ϵ-neighborhood graph is that it is sometimes difficult to find a proper $\epsilon > 0$ in the figure if different parts of the figure contains different distances between the data points.*

Remark 5.2. *If we use the similarity matrix S induced by the ϵ-neighborhood graph, then the corresponding Laplacian matrix is usually sparse.*

Algorithm We now state the algorithm of the unnormalized spectral clustering.

Algorithm 4 Unnormalized Spectral Clustering

Input: similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters we want
Output: k clusters of the data points
1: Compute the Laplacian matrix L
2: Compute the first k eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ of L
3: Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ as columns.
4: Let $\mathbf{y}_i \in \mathbb{R}^k$ be the vectors correspoinding to the ith row of U, $i = 1, \ldots, n$.
5: Cluster the points $\mathbf{y_i}_i, i = 1, \ldots, n$ in $\mathbb{R}^k$ with the k-means clustering into clusters $C_1, \ldots, C_k$.

How do we choose k? Ideally, we want to have k clusters such that the first k eigenvalues of the Laplacian matrix $\lambda_1, \lambda_2, \ldots, \lambda_k$ are relatively small, but λ_{k+1} are relatively large. The multiplicity of the eigenvalue $\lambda = 0$ may help give us an estimation of k.

Remark 5.3. *([9]) The k-means step is not forced. We can also use other techniques to construct the final solution from the real-valued representation.*

Remark 5.4. *([9]) The speed of convergence of spectral clustering depends on the size of eigengap, $\Delta_k = |\lambda_k - \lambda_{k+1}|$. The larger the eigengap is, the faster the algorithm computing the first k eigenvectors converge.*

Example 5.1. *We now generate data from two half-moon shapes. We want to see how we can use the unnormalized spectral clustering to correctly identify the two clusters.*

Recognition: We use the MATLAB code for the two-moons function provided on the website: ***https://www.mathworks.com/help/stats/label-data-using-semi-supervised-learning-techniques.html****. The MATLAB code*

```
rng("default")
[X,label] = twomoons(500);
epsilon = 0.3;k = 2;

S = similarity(X,epsilon);
idx1 = my_unnormalized_sc(S,k);

figure
gscatter(X(:,1),X(:,2),idx1,"rb");
legend("cluster 1","cluster 2","cluster 3")
title("unnormalized spectral clustering")
```

printed the following

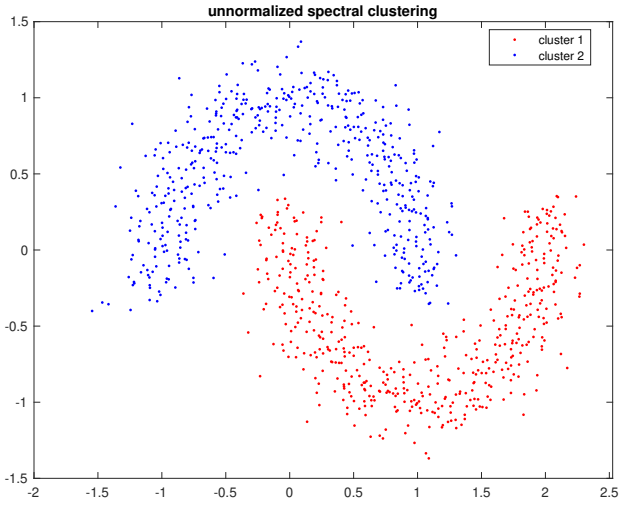

Figure 7: Unnormalized spectral clustering

6 Ratio cut and Unnormalized Spectral Clustering

In this section, we introduce the concept of the ratio cut and see how it is related to unnormalized spectral clustering.

Definition 6.1. *Given a clustering $V = C_1 \cup \cdots \cup C_k$, the **ratio cut** function is:*

$$RatioCut(C_1, \ldots, C_k) = \sum_{l=1}^{k} \frac{cut(C_l, \bar{C}_l)}{|C_l|}$$

where $|C_l|$ is the number of vertices in C_l and

$$cut(C_l, \bar{C}_l) = \sum_{v_i \in C_l, v_j \in \bar{C}_l} w_{ij}$$

Remark 6.1. *For a clustering $V = C_1 \cup \cdots \cup C_k$, we explicitly request that the clusters $C_1, C_2, \ldots, C_k$ are relatively large so that they do not contain only a few data points.*

Definition 6.2. *([9]) Given a partition of V into k clusters $C_1, \ldots, C_k$, we define the **unnormalized cluster indicating vectors** $\mathbf{h}_l \in \mathbb{R}^n, l = 1, 2, \ldots, k$ such that*

$$h_{il} = \begin{cases} \dfrac{1}{\sqrt{|C_l|}}, & \text{if } v_i \in C_l \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

Let $H \in \mathbb{R}^{n \times k}$ be the matrix where $H = \big[\mathbf{h}_1, \ldots, \mathbf{h}_k\big]$.

Proposition 6.1. *([9]) H is an orthogonal matrix, i.e., $H^T H = I \in \mathbb{R}^{k \times k}$.*

Proof. Let $(H^T H)_{uv}$ denote the (uv)-entry of $(H^T H)$.

$$(H^T H)_{uv} = \sum_{i=1}^{n} h_{iu} h_{iv}$$

- Suppose $u = v$.
  Then $h_{iu}h_{iv} = (h_{iu})^2 = (\frac{1}{\sqrt{|C_u|}})^2 = \frac{1}{|C_u|}$ if $v_i \in C_u$ and 0 otherwise.
  Then $(H^TH)_{uu} = \sum_{i=1}^{n} h_{iu}h_{iv} = \sum_{v_i \in C_u} \frac{1}{|C_u|} + \sum_{v_i \notin C_u} 0 = 1$.
- Suppose $u \neq v$.
  Then $h_{iu}h_{iv} = (0)(\frac{1}{\sqrt{|C_v|}}) = 0$ if $v_i \in C_v$, and $h_{iu}h_{iv} = (\frac{1}{\sqrt{|C_u|}})(0) = 0$ if $v_i \in C_u$.
  Then $(H^TH)_{uv} = \sum_{v_i \in C_u} 0 + \sum_{v_i \in C_v} 0 = 0$.

Therefore, $H^TH = I \in \mathbb{R}^{k \times k}$. □

Proposition 6.2. *([9])*

$$(H^TLH)_{ii} = \mathbf{h}_i^T L \mathbf{h}_i = \frac{cut(C_i, \bar{C}_i)}{|C_i|}$$

Proof. By Theorem 4.1 and Equation (6.1),

$$\begin{aligned}
\mathbf{h}_i^T L \mathbf{h}_i &= \frac{1}{2}\sum_{x=1}^{n}\sum_{y=1}^{n} w_{xy}(h_{xi} - h_{yi})^2 \\
&= \frac{1}{2}\sum_{x \in C_i, y \in \bar{C}_i} w_{xy}(h_{xi} - h_{yi})^2 + \frac{1}{2}\sum_{x \in \bar{C}_i, y \in C_i} w_{xy}(h_{xi} - h_{yi})^2 \\
&= \frac{1}{2}\sum_{x \in C_i, y \in \bar{C}_i} w_{xy}(\frac{1}{\sqrt{|C_i|}} - 0)^2 + \frac{1}{2}\sum_{x \in \bar{C}_i, y \in C_i} w_{xy}(0 - \frac{1}{\sqrt{|C_i|}})^2 \\
&= \frac{1}{2}\sum_{x \in C_i, y \in \bar{C}_i} w_{xy}\frac{1}{|C_i|} + \frac{1}{2}\sum_{x \in \bar{C}_i, y \in C_i} w_{xy}\frac{1}{|C_i|} \\
&= \sum_{x \in C_i, y \in \bar{C}_i} w_{xy}\frac{1}{|C_i|} \\
&= \frac{cut(C_i, \bar{C}_i)}{|C_i|}
\end{aligned}$$

□

Hence, we have

$$RatioCut(C_1, \ldots, C_k) = \sum_{i=1}^{k} \mathbf{h}_i^T L \mathbf{h}_i = \sum_{i=1}^{k} (H^TLH)_{ii} = Tr(H^TLH)$$

Thus, the problem of minimizing $RatioCut(C_1, \ldots, C_k)$ is the same as:

$$\min_{H \text{ is of form (6.1)}} Tr(H^TLH) \text{ subject to } H^TH = I \in \mathbb{R}^{k \times k} \tag{6.2}$$

According to section 5.1 and section 5.2 of [9], this is NP-hard. Hence, we relax the problem by allowing the entries of the matrix H to take arbitrary real values. Then the relaxed problem becomes:

$$\min_{H \in \mathbb{R}^{n \times k}} Tr(H^TLH) \text{ subject to } H^TH = I \in \mathbb{R}^{k \times k} \tag{6.3}$$

By section 5.2 of [9], it turns out that the trace minimization problem can be solved if we choose H that contains the first k eigenvectors of L as columns, which is the same as the matrix U in Algorithm 4. Thus, we first compute H by computing the first k eigenvectors of L and then apply k-means clustering to the rows of H. This leads to the unnormalized spectral clustering.

7 Normalized Spectral Clustering

We have introduced the idea of the unnormalized spectral clustering. Naturally, we would think that there is a method called the normalized spectral clustering. To understand the normalized spectral clustering, we have to first understand some basic concepts on the normalized Laplacian matrix.

In this section, we focus on the the normalized Laplacian matrix and the normalized spectral clustering.

7.1 Normalized Laplacian Matrix

Let $D^{-\alpha} = diag(d_1^{-\alpha}, \ldots, d_n^{-\alpha})$ where $\alpha \in \mathbb{R}$.

Definition 7.1. *([9]) Given a graph $G = (V, E, W)$ with no isolated vertex, the **symmetric normalized Laplacian matrix** L_{sym} and the **random walk normalized Laplacian matrix** L_{rw} are defined by*

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$
$$L_{rw} = D^{-1} L = I - D^{-1} W$$

Remark 7.1. *L_{sym} is a symmetric matrix because*

$$(L_{sym})^T = (D^{-\frac{1}{2}} L D^{-\frac{1}{2}})^T = (D^{-\frac{1}{2}})^T L^T (D^{-\frac{1}{2}})^T = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = L_{sym}$$

Remark 7.2.

$$\begin{aligned} L_{rw} &= D^{-1} L \\ &= D^{-\frac{1}{2}} (D^{-\frac{1}{2}} L D^{-\frac{1}{2}}) D^{\frac{1}{2}} \\ &= D^{-\frac{1}{2}} L_{sym} D^{\frac{1}{2}} \end{aligned}$$

Example 7.1. *In Example 1.1 and Example 1.2, we figure out $D, A,$ and L of the graph G in Figure 1. We have*

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}, L = \begin{bmatrix} 2 & -1 & 0 & -1 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

Then

$$D^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}, D^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

Therefore,

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = \begin{bmatrix} 1 & -\frac{1}{2\sqrt{2}} & 0 & -\frac{1}{\sqrt{6}} & 0 \\ -\frac{1}{2\sqrt{2}} & 1 & -\frac{1}{2} & -\frac{1}{2\sqrt{3}} & -\frac{1}{2\sqrt{2}} \\ 0 & -\frac{1}{2} & 1 & 0 & 0 \\ -\frac{1}{\sqrt{6}} & -\frac{1}{2\sqrt{3}} & 0 & 1 & -\frac{1}{\sqrt{6}} \\ 0 & -\frac{1}{2\sqrt{2}} & 0 & -\frac{1}{\sqrt{6}} & 1 \end{bmatrix}$$

$$L_{rw} = D^{-1} L = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & -\frac{1}{2} & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ 0 & -1 & 1 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & 0 & 1 & -\frac{1}{3} \\ 0 & -\frac{1}{2} & 0 & -\frac{1}{2} & 1 \end{bmatrix}$$

Theorem 7.1. *([9]) For every $\mathbf{x} \in \mathbb{R}^n$, we have*

$$\mathbf{x}^T L_{sym} \mathbf{x} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})^2$$

Proof.

$$\begin{aligned} \mathbf{x}^T L_{sym} \mathbf{x} &= \mathbf{x}^T (I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}) \mathbf{x} \\ &= \sum_{i=1}^{n} x_i^2 - \sum_{i,j=1}^{n} \frac{x_i}{\sqrt{d_i}} w_{ij} \frac{x_j}{\sqrt{d_j}} \\ &= \frac{1}{2} (\sum_{i=1}^{n} x_i^2 - 2 \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{x_i w_{ij} x_j}{\sqrt{d_i \sqrt{d_j}}} + \sum j = 1^n x_j^2) \\ &= \frac{1}{2} (\sum_{i=1}^{n} \sum_{j=1}^{n} \frac{w_{ij}}{d_i} x_i^2 - 2 \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{x_i w_{ij} x_j}{\sqrt{d_i} \sqrt{d_j}} + \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{w_{ij}}{d_j} x_j^2) \\ &= \frac{1}{2} (\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (\frac{x_i}{\sqrt{d_i}})^2 - 2 \frac{x_i w_{ij} x_j}{\sqrt{d_i} \sqrt{d_j}} + w_{ij} (\frac{x_j}{\sqrt{d_j}})^2) \\ &= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})^2 \end{aligned}$$

□

Theorem 7.2. *([9]) λ is an eigenvalue of L_{rw} with eigenvector $\mathbf{v}$ if an only if λ is an eigenvalue of L_{sym} with eigenvector $\mathbf{w} = D^{\frac{1}{2}} \mathbf{v}$.*

Proof.

$$\begin{gathered} L_{sym}(D^{\frac{1}{2}} \mathbf{v}) = \lambda (D^{\frac{1}{2}} \mathbf{v}) \\ D^{-\frac{1}{2}} L D^{-\frac{1}{2}} D^{\frac{1}{2}} \mathbf{v} = D^{\frac{1}{2}} \lambda \mathbf{v} \\ D^{-\frac{1}{2}} (D^{-\frac{1}{2}} L D^{-\frac{1}{2}}) D^{\frac{1}{2}} D^{\frac{1}{2}} \mathbf{v} = \lambda \mathbf{v} \\ D^{-1} L I \mathbf{v} = \lambda \mathbf{v} \\ (D^{-1} L) \mathbf{v} = \lambda \mathbf{v} \\ L_{rw} \mathbf{v} = \lambda \mathbf{v} \end{gathered}$$

□

Theorem 7.3. *([9]) λ is an eigenvalue of L_{rw} with eigenvector $\mathbf{v}$ if and only if λ and $\mathbf{v}$ solve the generalized eigenvalue problem $L\mathbf{v} = \lambda D\mathbf{v}$.*

Proof. We show that $L_{rw}\mathbf{v} = \lambda\mathbf{v}$.
Then $D^{-1}L\mathbf{v} = \lambda\mathbf{v}$.
Therefore, $L\mathbf{v} = \lambda D\mathbf{v}$. □

Theorem 7.4. *([9]) 0 is an eigenvalue of L_{rw} with the constant one vector $\mathbf{1}$ as eigenvector. 0 is an eigenvalue of L_{sym} with eigenvector $D^{\frac{1}{2}}\mathbf{1}$.*

Proof. The first statement true is because

$$L_{rw}\mathbf{1} = I\mathbf{1} - D^{-1}W\mathbf{1} = \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} - \begin{bmatrix}\frac{1}{d_1}\sum_{j=1}^{n} w_{1j}\\\frac{1}{d_2}\sum_{j=1}^{n} w_{2j}\\\vdots\\\frac{1}{d_n}\sum_{j=1}^{n} w_{nj}\end{bmatrix} = \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} - \begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} = \mathbf{0}$$

Therefore, L_{sym} has an eigenvalue $\lambda = 0$ with the corresponding eigenvector $\mathbf{v} = \mathbf{1}$.
The second statement is true because of Theorem 7.2.
Therefore, L_{rw} has an eigenvalue $\lambda = 0$ with the corresponding eigenvector $\mathbf{v} = D^{\frac{1}{2}}\mathbf{1}$. □

Theorem 7.5. *([9]) L_{sym} and L_{rw} are positive semidefinite and have n nonnegative real-valued eigenvalues $0 = \lambda_1 \leq \cdots \leq \lambda_n$.*

Proof. By Theorem 7.1, we have

$$\mathbf{x}^T L_{sym}\mathbf{x} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})^2$$

for every $\mathbf{x} \in \mathbb{R}^n$.
Then $\mathbf{x}^T L_{sym}\mathbf{x} \geq 0$ for every $\mathbf{x} \in \mathbb{R}^n$.
Hence, L_{sym} is positive semidefinite.
Therefore, $L_{sym} \in \mathbb{R}^{n\times n}$ has n nonnegative real-valued eigenvalues $0 = \lambda_1 \leq \cdots \leq \lambda_n$.
By Theorem 7.2, since L_{rw} has the same eigenvalues as L_{sym}, L_{rw} also has eigenvalues $0 = \lambda_1 \leq \cdots \leq \lambda_n$.
Therefore, L_{rw} is positive semidefinite. □

Proposition 7.1. *([9]) The multiplicity k of the eigenvalue 0 of both L_{rw} and L_{sym} equals the number of connected components $A_1, \ldots, A_k$ in the graph.*

Proof. The proof is similar to that of Theorem 2.1. □

7.2 Algorithms

Algorithm. We have shown some basic properties of the normalized Laplacian matrix L_{rw} and L_{sym}. Now we can introduce the normalized spectral clustering. The algorithm is called normalized spectral clustering because we use the eigenvectors of the normalized Laplacian matrices.

Algorithm 5 Normalized Spectral Clustering using L_{rw}

Input: similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters we want
Output: k clusters of the data points

1: Compute the unnormalized Laplacian matrix L.
2: Compute the first k eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ of the eigenproblem $L\mathbf{u} = \lambda D\mathbf{u}$.
3: Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ as columns.
4: Let $\mathbf{y}_i \in \mathbb{R}^k$ be the vectors corresponding to the ith row of U, $i = 1, \ldots, n$.
5: Cluster the points $\mathbf{y_i}_i, i = 1, \ldots, n$ in $\mathbb{R}^k$ with the k-means clustering into clusters $C_1, \ldots, C_k$.

Algorithm 6 Normalized Spectral Clustering using L_{sym}

Input: similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters we want
Output: k clusters of the data points

1: Compute the unnormalized Laplacian matrix L
2: Compute the normalized Laplacian matrix L_{sym}
3: Compute the first k eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ of L_{sym}
4: Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ as columns.
5: Let $T \in \mathbb{R}^{n \times k}$ be the row-normalized matrix such that $t_{ij} = \frac{u_{ij}}{(\sum\limits_k u_{ik}^2)^{\frac{1}{2}}}$
6: Let $\mathbf{y}_i \in \mathbb{R}^k$ be the vectors corresponding to the ith row of T, $i = 1, \ldots, n$.
7: Cluster the points $\mathbf{y_i}_i, i = 1, \ldots, n$ in $\mathbb{R}^k$ with the k-means clustering into clusters $C_1, \ldots, C_k$.

Practical Method for solving U in Algorithm 5 We have to be very careful about how to program so that we take the first k eigenvectors of L_{rw}. If we directly compute L_{rw} with $L_{rw} = D^{-1}L$, then L_{rw} may not be symmetric, so the eigenvalue of L_{rw} may not all be real values. Fortunately, in Theorem 7.3, we find out that the first k eigenvalues and eigenvectors of L_{rw} are indeed the solutions to the eigenproblem $L\mathbf{u} = \lambda D\mathbf{u}$. Therefore, we can solve the eigenproblem instead.

Example 7.2. *We use the two-moons data again. This time we want to confirm that the normalized spectral clustering using L_{rw} to correctly identify the two clusters.*

The MATLAB code

```
[X,label] = twomoons(500);

epsilon = 0.3;
k = 2;

S = similarity(X,epsilon);
idx1 = my_normalized_sc_rw(S,k)

figure
gscatter(X(:,1),X(:,2),idx1,"rb");
legend("cluster 1","cluster 2")
title("spectral clustering using L_{rw}")
```

printed the following

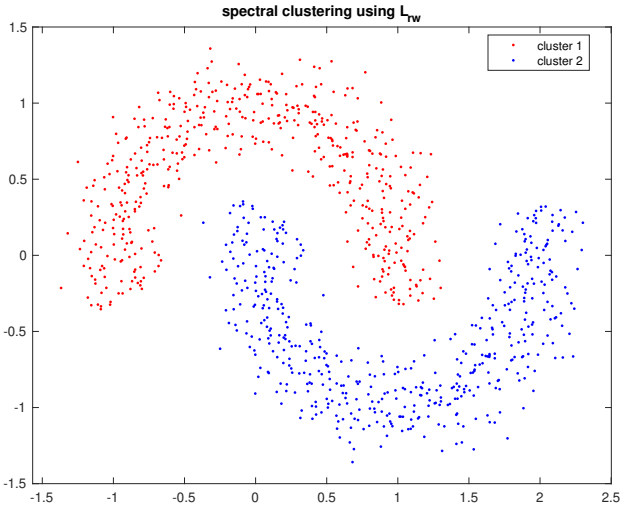

Figure 8: Normalized spectral clustering using L_{rw}

Example 7.3. *We use the normalized spectral clustering with L_{sym} to identity the two clusters in the data from two-moons shapes.*
The MATLAB code

```
[X,label] = twomoons(500);

epsilon = 0.3; k = 2;

S = similarity(X,epsilon);
idx1 = my_normalized_sc_sym(S,k)

figure
gscatter(X(:,1),X(:,2),idx1,"rb");
legend("cluster 1","cluster 2")
title("spectral clustering using L_{sym}")
```

printed the following

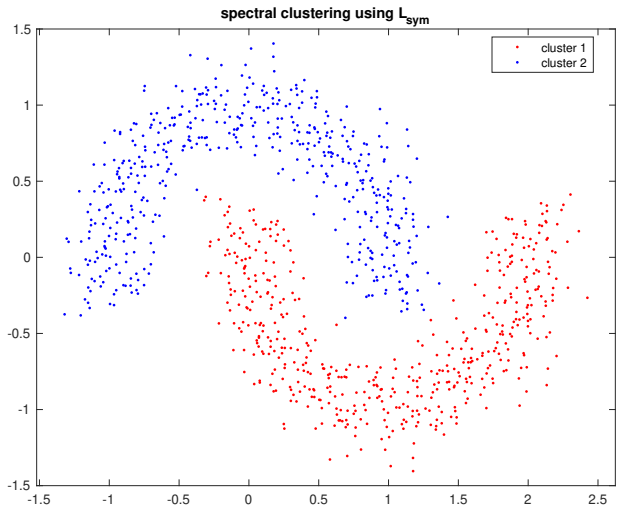

Figure 9: Normalized spectral clustering using L_{sym}

8 Normalized Cut and Normalized Spectral Clustering

In this section, we introduce the concept of the normalized cut and see how it is related to normalized spectral clustering.

Definition 8.1. *Given a clustering $V = C_1 \cup \cdots \cup C_k$, the* ***normalized cut*** *function is:*

$$NCut(C_1, \ldots, C_k) = \sum_{l=1}^{k} \frac{cut(C_l, \bar{C}_l)}{vol(C_l)}$$

where

$$vol(C_l) = \sum_{v_i \in C_l} d_i$$

Definition 8.2. *([9]) Given a partition of V into k clusters $C_1, \ldots, C_k$, we define k* ***normalized cluster indicating vectors*** *$\mathbf{h}_l \in \mathbb{R}^n, l = 1, \ldots, k$ such that*

$$h_{il} = \begin{cases} \dfrac{1}{\sqrt{vol(C_l)}}, & \text{if } v_i \in C_l \\ 0, & \text{otherwise} \end{cases} \tag{8.1}$$

Then we set $H \in \mathbb{R}^{n \times k}$ as the matrix where $H = [\mathbf{h_1}, \ldots, \mathbf{h}_k]$.

Proposition 8.1. *([9]) Let H be of form (8.1) and D be the degree matrix of the graph $G = (V, E, W)$. Then*

$$H^T D H = I \in \mathbb{R}^{k \times k}$$

Proof. Let $(H^T D H)_{uv}$ denote the (u, v)-entry of $(H^T D H)$.

$$(H^T D H)_{uv} = \sum_{i=1}^{n} h_{iu} \cdot d_i \cdot h_{iv}$$

- Suppose $u = v$.
  Then $h_{iu} \cdot d_i \cdot h_{iv} = (h_{iu})^2 \cdot d_i = (\frac{1}{\sqrt{vol(C_l)}})^2 \cdot d_i = \frac{d_i}{vol(C_l)}$ if $v_i \in C_u$ and 0 otherwise.
  Note that $vol(C_u) = \sum_{v_i \in C_u} d_i$.
  Then $(H^T DH)_{uv} = \sum_{i=1}^{n} h_{iu} d_i h_{iv} = \sum_{v_i \in C_u} \frac{1}{vol(C_u)} d_i + \sum_{i \notin C_u} 0 = 1$.
- Suppose $u \neq v$.
  Then $h_{iu} \cdot d_i \cdot h_{iv} = (0) d_i (\frac{1}{\sqrt{vol(C_l)}}) = 0$ if $v_i \in C_v$, and $h_{iu} h_{iv} = (\frac{1}{\sqrt{vol(C_l)}}) d_i (0) = 0$ if $v_i \in C_u$.
  Then $(H^T DH)_{uv} = \sum_{v_i \in C_u} 0 + \sum_{v_i \in C_v} 0 = 0$.

Therefore, $H^T DH = I \in \mathbb{R}^{k \times k}$. □

Proposition 8.2. *([9])*

$$(H^T LH)_{ii} = \mathbf{h}_i^T L \mathbf{h}_i = \frac{cut(C_i, \bar{C}_i)}{vol(C_i)}$$

Proof. By Theorem 4.1 and Equation (8.1),

$$\begin{aligned}
\mathbf{h}_i^T L \mathbf{h}_i &= \frac{1}{2} \sum_{x=1}^{n} \sum_{y=1}^{n} w_{xy} (h_{xi} - h_{yi})^2 \\
&= \frac{1}{2} \sum_{x \in C_i, y \in \bar{C}_i} w_{xy} (h_{xi} - h_{yi})^2 + \frac{1}{2} \sum_{x \in \bar{C}_i, y \in C_i} w_{xy} (h_{xi} - h_{yi})^2 \\
&= \frac{1}{2} \sum_{x \in C_i, y \in \bar{C}_i} w_{xy} (\frac{1}{\sqrt{vol(C_i)}} - 0)^2 + \frac{1}{2} \sum_{x \in \bar{C}_i, y \in C_i} w_{xy} (0 - \frac{1}{\sqrt{vol(C_i)}})^2 \\
&= \frac{1}{2} \sum_{x \in C_i, y \in \bar{C}_i} w_{xy} \frac{1}{vol(C_i)} + \frac{1}{2} \sum_{x \in \bar{C}_i, y \in C_i} w_{xy} \frac{1}{vol(C_i)} \\
&= \sum_{x \in C_i, y \in \bar{C}_i} w_{xy} \frac{1}{vol(C_i)} \\
&= \frac{cut(C_i, \bar{C}_i)}{vol(C_i)}
\end{aligned}$$

□

Hence, we have

$$NCut(C_1, \ldots, C_k) = \sum_{i=1}^{k} \frac{cut(C_i, \bar{C}_i)}{vol(C_i)} = \sum_{i=1}^{k} \mathbf{h}_i^T L \mathbf{h}_i = \sum_{i=1}^{k} (H^T LH)_{ii} = Tr(H^T LH)$$

Thus, the problem of minimizing $NCut(C_1, \ldots, C_k)$ is the same as:

$$\min_{H \text{ is of form (8.1)}} Tr(H^T LH) \text{ subject to } H^T DH = I \in \mathbb{R}^{k \times k} \tag{8.2}$$

Relaxing the discreteness condition, we allow H to take arbitrary real values. Then the relaxed problem becomes:

$$\min_{H \in \mathbb{R}^{n \times k}} Tr(H^T LH) \text{ subject to } H^T DH = I \in \mathbb{R}^{k \times k} \tag{8.3}$$

Substituting $H = D^{-1/2}T$ for $T \in \mathbb{R}^{n \times k}$, we obtain the relaxed problem

$$\min_{T \in \mathbb{R}^{n \times k}} Tr(T^T D^{-1/2} L D^{-1/2} T) \text{ subject to } T^T T = I \in \mathbb{R}^{k \times k} \tag{8.4}$$

By section 5.3 of [9], this is the standard trace minimization problem which is solved by the matrix T which contains the first k eigenvectors of $L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ as columns. Re-substituting $H = D^{-1/2}T$ and using Theorem 7.2, we see that the solution H consists of the first k eigenvectors of L_{rw}, or the first k generalized eigenvectors of $L\mathbf{u} = \lambda D\mathbf{u}$. This yields the normalized spectral clustering algorithm according to Algorithms 5.

9 Advantages of Normalized Spectral Clustering over Unnormalized Spectral Clustering

We have learned the concepts of unnormalized spectral clustering and normalized spectral clustering. One might wonder: which graph Laplacian matrix should we use to do the spectral clustering and why?

In this section, we compare the spectral clustering algorithms and realize the superiority of the normalized spectral clustering over the unnormalized spectral clustering.

Advantages of normalized spectral clustering over unnormalized spectral clustering ([9]) If the vertices of a given graph $G = (V, E)$ have approximately the same degree, then all the Laplacian matrices L, L_{rw}, L_{sym} will be similar to each other and work equally well for spectral clustering. However, if the graph are very broadly distributed, then there are two reasons to advocate for using normalized spectral clustering rather than unnormalzied spectral clustering.

1. By the graph partition point of view, the objectives of spectral clustering are to put dissimilar points into different clusters and similar points into the same cluster. The normalized spectral clustering can successfully implement both of the objectives while unnormalized can only implement the first objective.

2. By the statistical analysis of both algorithms, if we draw more and more data points, then normalized spectral clustering will converge to a useful partition in most real world applications. In contrast, unnormalized spectral clustering can fail to converge, or that it can converge to trivial solutions which construct clusters consisting of one single point. We have to make sure that the eigenvalues of L corresponding to the eigenvectors used in unnormalized spectral clustering are significantly smaller than the minimal degree in the graph.

Therefore, from the graph partition point of view and the statistical point of view, we should always use normalized spectral clustering rather than unnormalized spectral clustering.

Advantages of using L_{rw} over L_{sym} ([9]) The eigenvectors of L_{rw} are cluser indicator vectors while the eigenvectors of L_{sym} are additionally multiplied with $D^{\frac{1}{2}}$, which might lead to undesired results. Using L_{sym} does not have any computational advantages as well. Therefore, we should always use L_{rw}, i.e., it is preferable to use Algorithm 5 for spectral clustering.

Example 9.1. *This is an example to verify the superiority of normalized spectral clustering. It has 3000 randomly generated data points. We see that normalized spectral clustering provides* 3 *useful clusters. On the contrary, unnormalized spectral clustering gives a trivial solution.*
The MATLAB code

```
rng("default");
X = [randn(1000,2)*0.75+ones(1000,2);randn(1000,2)*0.5-ones(1000,2);randn(1000,2)*0.75];
epsilon = 1;k = 3;
S = similarity(X,epsilon);
D = diag(sum(S,2));
L = D - S;
[v1, e1] = eig(L,D);
U1 = v1(:,1:k);idx1 = kmeans(U1,k);

[v2, e2] = eig(L);
U2 = v2(:,1:k);
idx2 = kmeans(U2,k);

ax1 = subplot(2,1,1);
gscatter(X(:,1),X(:,2),idx1,"rgb");
legend("cluster 1","cluster 2","cluster 3")
title(ax1,"Normalized spectral clustering using")

ax2 = subplot(2,1,2);
gscatter(X(:,1),X(:,2),idx2,"rgb");
legend("cluster 1","cluster 2","cluster 3")
title(ax2,"Unnormalized spectral clustering")
```

printed the following

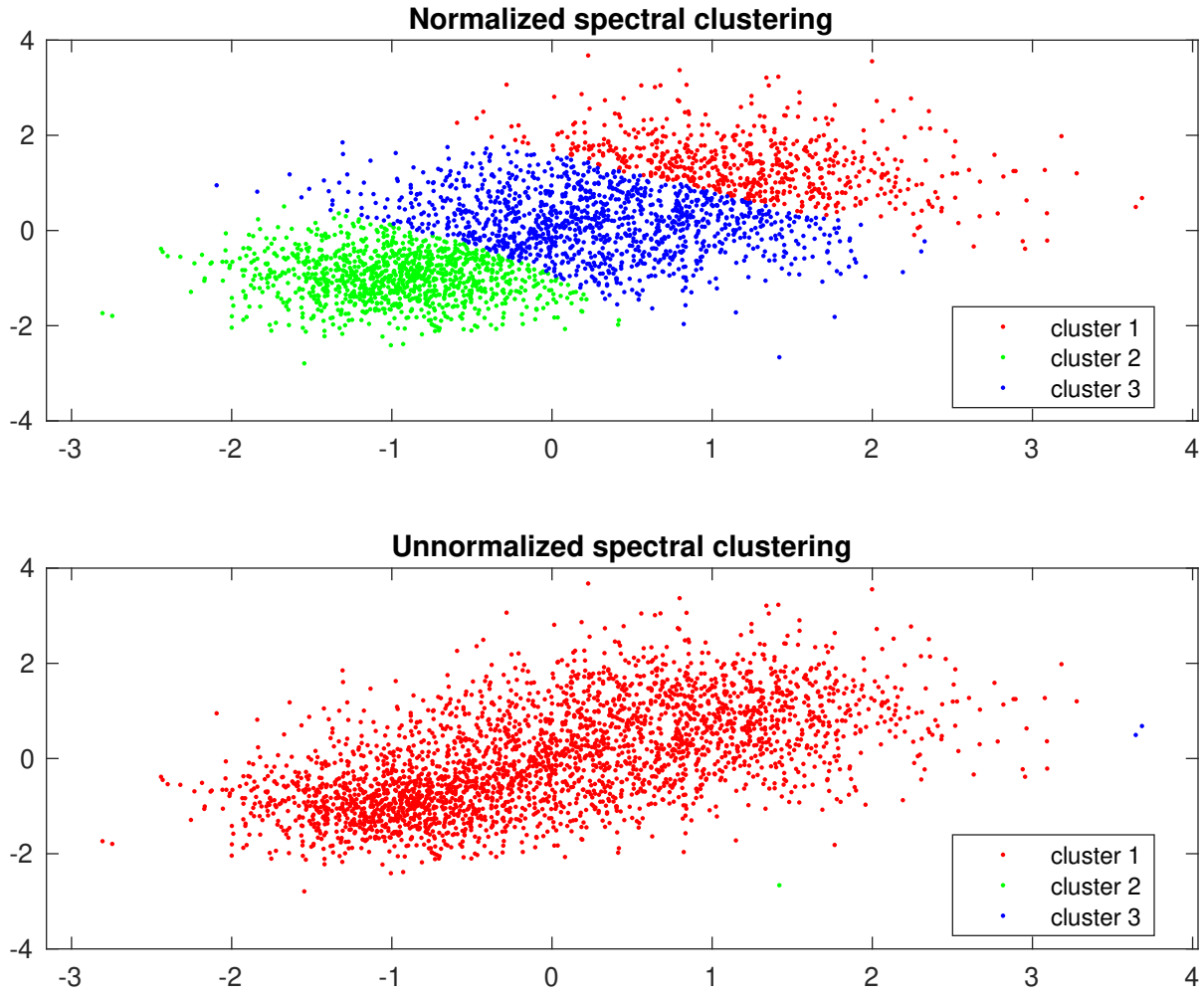

Figure 10: Comparison between normalized and unnormalized spectral clustering

Part III
Fairness and Fair Spectral Clustering

In this part, we introduce two fairness notions: the group fairness criteria and the individual fairness criteria. By satisfying the group fairness criteria, we will have a clustering where each data group has equal representation in every cluster. In this way, we can partition the data in a way that "does not discriminate based on some protected attribute."([1])

The individual fairness criteria further extends the group fairness criteria. It tends to find clusters that represent the interests of every individual. That is, the clustering asks that the adjacent vertices of every vertex in the graph are proportionally represented in each cluster.

We can incorporate both group fairness criteria and individual fairness criteria into spectral clustering and build the corresponding models and then do examples to see how these constraints alter the behaviors of spectral clustering.

10 Group Fairness

In this section, we give the definition of group fairness along with examples. Then we study how to achieve the group fairness criteria in terms of H where H is of form (8.1) so that we translate it from the set notation into the linear algebra notation.

10.1 Definition

Definition 10.1. *([7]) Given $V = V_1 \cup \cdots \cup V_h$ where the groups $V_1, \ldots, V_h$ are disjoint, we define the **group fairness criteria** as follows:*

- *a cluster C_l is statistically fair if all the groups have proportional representation in it. That is,*

$$\frac{|V_s \cap C_l|}{|C_l|} = \frac{|V_s|}{|V|} \text{ for every } s \in \{1, \ldots, h\} \tag{10.1}$$

- *a clustering $C = C_1 \cup \cdots \cup C_k$ is statistically fair if:*

$$\text{for every } l \in \{1, \ldots, k\}, \ \frac{|V_s \cap C_l|}{|C_l|} = \frac{|V_s|}{|V|}, \ \text{for every } s \in \{1, \ldots, h\} \tag{10.2}$$

Example 10.1. *Consider the topics of college admission. Suppose that there are 720 applicants. Among them, there are 360 Whites, 240 Asians, 60 Blacks, and 60 Hispanics. Let $V_1 = \{Whites\}, V_2 = \{Asians\}, V_3 = \{Blacks\}$, and $V_4 = \{Hispanics\}$.*

Now suppose that the admission committee have made their decisions so that the applicants are clustered into two according to the admission decisions. Let $C_1 = \{$successful applicants$\}$ and $C_2 = \{$unsuccessful applicants$\}$.Suppose that one-third of applicant from each ethnic groups are admitted. That is, 120 Whites, 80 Asians, 20 Blacks, and 20 Hispanics are admitted.

We want to check that the admission satisfies that group fairness criteria so that no race is discriminated in the admission process.

We have $|V| = 720, |V_1| = 360, |V_2| = 240, |V_3| = 60, |V_4| = 60$. Then

$$\frac{|V_1|}{|V|} = \frac{1}{2}, \frac{|V_2|}{|V|} = \frac{1}{3}, \frac{|V_3|}{|V|} = \frac{1}{12}, \frac{|V_1|}{|V|} = \frac{1}{12}$$

For C_1,

$$\frac{|V_1 \cap C_1|}{|C_1|} = \frac{120}{240} = \frac{|V_1|}{|V|}, \frac{|V_2 \cap C_1|}{|C_1|} = \frac{80}{240} = \frac{|V_2|}{|V|},$$
$$\frac{|V_3 \cap C_1|}{|C_1|} = \frac{20}{240} = \frac{|V_3|}{|V|}, \frac{|V_4 \cap C_1|}{|C_1|} = \frac{20}{240} = \frac{|V_4|}{|V|}$$

For C_2,

$$\frac{|V_1 \cap C_2|}{|C_2|} = \frac{240}{480} = \frac{|V_1|}{|V|}, \frac{|V_2 \cap C_2|}{|C_2|} = \frac{160}{480} = \frac{|V_2|}{|V|},$$
$$\frac{|V_3 \cap C_2|}{|C_2|} = \frac{40}{480} = \frac{|V_3|}{|V|}, \frac{|V_4 \cap C_2|}{|C_2|} = \frac{40}{480} = \frac{|V_4|}{|V|}$$

Hence, for every $l \in \{1, 2\}$, $\frac{|V_s \cap C_l|}{|C_l|} = \frac{|V_s|}{|V|}$ where $s \in \{1, 2, 3, 4\}$.
Thus, the group fairness criteria is satisfied, and we have a statistically fair admission.

Example 10.2. *Suppose we are given the following data points. V_1 are the 4 rectangle data points, and V_2 are the 4 circle data points. To have 2 statistically fair clusters, we want to find clusters C_1 and C_2 such that each cluster contains 2 data points from V_1 and 2 data points from V_2.*

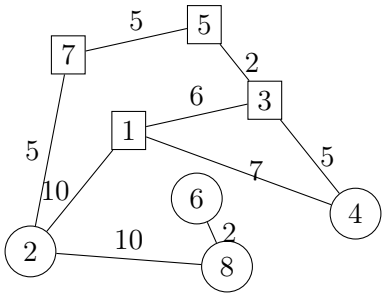

Figure 11: Graph G

Then one way to do the find a clustering that satisfies the group fairness criteria is the following. We can check that the group fairness criteria is satisfied.

We have: $V_1 = \{v_1, v_3, v_5, v_7\}$, $V_2 = \{v_2, , v_4, v_6, v_8\}$, $C_1 = \{v_1, v_2, v_6, v_7\}$, and $C_2 = \{v_3, v_4, v_5, v_8\}$. Then

$$\frac{|V_1|}{|V|} = \frac{4}{8}, \frac{|V_2|}{|V|} = \frac{4}{8}$$

For C_1,

$$\frac{|V_1 \cap C_1|}{|C_1|} = \frac{2}{4} = \frac{|V_1|}{|V|}, \frac{|V_1 \cap C_1|}{|C_1|} = \frac{2}{4} = \frac{|V_2|}{|V|}$$

For C_2,

$$\frac{|V_1 \cap C_2|}{|C_2|} = \frac{2}{4} = \frac{|V_1|}{|V|}, \frac{|V_1 \cap C_2|}{|C_2|} = \frac{2}{4} = \frac{|V_2|}{|V|}$$

Hence, we have a statistically fair clustering.

Example 10.3. *We continue to study Example 10.2. This time to give a counterexample. Now if we change the clusters to $C_1 = \{v_1, v_3, v_5, v_7\}$ and $C_2 = \{v_2, v_4, v_6, v_8\}$, we will see that the clustering is not statistically fair.*

$$\frac{|V_1|}{|V|} = \frac{6}{8}, \frac{|V_2|}{|V|} = \frac{2}{8}$$

For C_1,

$$\frac{|V_1 \cap C_1|}{|C_1|} = \frac{4}{4} \neq \frac{|V_1|}{|V|}, \frac{|V_1 \cap C_1|}{|C_1|} = \frac{0}{4} \neq \frac{|V_2|}{|V|}$$

For C_2,

$$\frac{|V_1 \cap C_2|}{|C_2|} = \frac{0}{4} \neq \frac{|V_1|}{|V|}, \frac{|V_1 \cap C_2|}{|C_2|} = \frac{4}{4} \neq \frac{|V_2|}{|V|}$$

Therefore, the clustering is not statistically fair.

10.2 Matrix Representation

Definition 10.2. *Given $V = V_1 \cup \cdots \cup V_h$ where $V_1, \ldots, V_h$ are disjoint, we define the **group-membership vectors** $f^{(s)} \in \{0,1\}^n, s \in \{1, \ldots, h\}, n = |V|$ such that*

$$f_i^{(s)} = \begin{cases} 1, \text{ if } v_i \in V_s \\ 0, \text{ otherwise} \end{cases} \tag{10.3}$$

Proposition 10.1. *Let $\hat{F} \in \mathbb{R}^{n \times h}$ be the following matrix:*

$$\hat{F} = \begin{bmatrix} f^{(1)} - \frac{|V_1|}{|V|} \cdot \mathbf{1}_n & \ldots & f^{(h)} - \frac{|V_h|}{|V|} \cdot \mathbf{1}_n \end{bmatrix}$$

Then $rank(\hat{F}) = h - 1$.

Proof. • First, we want to show that $\hat{F}\mathbf{1}_h = \mathbf{0}_n$.

$$\begin{aligned} \hat{F}\mathbf{1}_h &= \sum_{s=1}^{h} (f^{(s)} - \frac{|V_s|}{|V|} \cdot \mathbf{1}_n) \\ &= \sum_{s=1}^{h} f^{(s)} - \sum_{s=1}^{h} \frac{|V_s|}{|V|} \cdot \mathbf{1}_n \\ &= \mathbf{1}_n - \mathbf{1}_n \\ &= \mathbf{0}_n \end{aligned}$$

Therefore, $\hat{F}\mathbf{1}_h = \mathbf{0}_n$ and $nullity(\hat{F}) \geq 1$.

• Next, we want to show that the matrix

$$F = \begin{bmatrix} f^{(1)} - \frac{|V_1|}{|V|} \cdot \mathbf{1}_n & \ldots & f^{(h-1)} - \frac{|V_{h-1}|}{|V|} \cdot \mathbf{1}_n \end{bmatrix}$$

has full column rank. That is, $F\mathbf{x} = \mathbf{0}_n \iff \mathbf{x} = \mathbf{0}_{h-1}$.
Suppose that for some vector $\mathbf{x} = \begin{bmatrix} x_1 & \ldots & x_{h-1} \end{bmatrix}^T \in \mathbb{R}^{(h-1)}$, we have $F\mathbf{x} = \mathbf{0}_n$.

Without loss of generality, let $f^{(s)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ where $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ corresponds to V_s.

Then

$$x_1(f^{(1)} - \frac{|V_1|}{|V|} \cdot \mathbf{1}_n) + \cdots + x_{h-1}(f^{(h-1)} - \frac{|V_{h-1}|}{|V|} \cdot \mathbf{1}_n) = \mathbf{0}_n$$

This is equivalent to

$$x_1 f^{(1)} + \cdots + x_{h-1} f^{(h-1)} = x_1 \frac{|V_1|}{|V|} \cdot \mathbf{1}_n + \cdots + x_{h-1} \frac{|V_{h-1}|}{|V|} \cdot \mathbf{1}_n$$

Looking at the ith row of both sides, we have:
for $i = 1, \ldots, |V_1|$:

$$x_1 = x_1 \frac{|V_1|}{|V|} + \cdots + x_{h-1} \frac{|V_{h-1}|}{|V|}$$

for $i = (|V_1| + 1), \ldots, (|V_1| + |V_2|)$:

$$x_2 = x_1 \frac{|V_1|}{|V|} + \cdots + x_{h-1} \frac{|V_{h-1}|}{|V|}$$

$$\vdots$$

for $i = (|V_1| + \cdots + |V_{h-2}| + 1), \ldots, (|V_1| + \cdots + |V_{h-1}|)$:

$$x_{h-1} = x_1 \frac{|V_1|}{|V|} + \cdots + x_{h-1} \frac{|V_{h-1}|}{|V|}$$

for $i = (|V_1| + \cdots + |V_{h-1}| + 1), \ldots, n$:

$$0 = x_1 \frac{|V_1|}{|V|} + \cdots + x_{h-1} \frac{|V_{h-1}|}{|V|}$$

From the last block, $x_1 \frac{|V_1|}{|V|} + \cdots + x_{h-1} \frac{|V_{h-1}|}{|V|} = 0$.
We can plug this result into the other blocks and recursively get $x_i = 0$ where $i = 1, \ldots, h-1$.
Thus, $F\mathbf{x} = 0_n \implies \mathbf{x} = \mathbf{0}_{h-1}$.
Conversely, if $\mathbf{x} = \mathbf{0}_{h-1}$, then $F\mathbf{x} = \mathbf{0}_n$.
Therefore, $F\mathbf{x} = \mathbf{0}_n \iff \mathbf{x} = \mathbf{0}_{h-1}$.
Therefore, F has full column rank.
Therefore, $\hat{F}$ has rank $h - 1$.

□

Lemma 10.1. *([7]) Let $F \in \mathbb{R}^{n\times(h-1)}$ consists of $h-1$ group-membership vectors such that*

$$F = \left[f^{(1)} - \tfrac{|V_1|}{|V|} \cdot \mathbf{1}_n \quad \dots \quad f^{(h-1)} - \tfrac{|V_{h-1}|}{|V|} \cdot \mathbf{1}_n \right] \tag{10.4}$$

Let $V = C_1 \cup \cdots \cup C_k$ be a clustering encoded by the matrix $H \in \mathbb{R}^{n\times k}$ where H consists of normalized cluster indicating vectors as in Equation (8.1). Then

$$F^T H = 0 \in \mathbb{R}^{(h-1)\times k} \tag{10.5}$$

is equivalent to the group fairness criteria (Equation (10.2)).

Proof. Let $l \in \{1, \dots, k\}$ be fixed. For every $s \in \{1, \dots, h\}$, by Equations 10.3 and Equations 8.1,

$$f_i^{(s)} h_{il} = \begin{cases} (1)(\frac{1}{\sqrt{vol(C_l)}}) = \frac{1}{\sqrt{vol(C_l)}}, \text{ if } v_i \in V_s \cap C_l \\ 0, \text{ otherwise} \end{cases}$$

Hence,

$$\sum_{i=1}^{n} f_i^{(s)} h_{il} = \sum_{v_i \in V_s \cap C_l} \frac{1}{\sqrt{vol(C_l)}} + \sum_{i \notin V_s \cap C_l} 0 = \frac{|V_s \cap C_l|}{\sqrt{vol(C_l)}}$$

By Equations 8.1,

$$\sum_{i=1}^{n} h_{il} = \sum_{v_i \in C_l} \frac{1}{\sqrt{vol(C_l)}} + \sum_{v_i \notin C_l} 0 = \sum_{i \in C_l} \frac{1}{\sqrt{vol(C_l)}}$$

Hence,

$$\sum_{i=1}^{n} \frac{|V_s|}{|V|} h_{il} = \frac{|V_s|}{|V|} \sum_{i=1}^{n} h_{il} = \frac{|V_s|}{|V|} \sum_{v_i \in C_l} \frac{1}{\sqrt{vol(C_l)}} = \frac{|V_s|}{|V|} \cdot \frac{|C_l|}{\sqrt{vol(C_l)}}$$

Combining the above, we get

$$\sum_{i=1}^{n} (f_i^{(s)} - \frac{|V_s|}{|V|}) h_{il} = \frac{|V_s \cap C_l|}{\sqrt{vol(C_l)}} - \frac{|V_s|}{|V|} \cdot \frac{|C_l|}{\sqrt{vol(C_l)}}$$

Therefore, for every $s \in \{1, \dots, h\}$

$$\sum_{i=1}^{n} (f_i^{(s)} - \frac{|V_s|}{|V|}) h_{il} = 0 \iff \frac{|V_s \cap C_l|}{C_l} = \frac{|V_s|}{|V|}$$

We already show that $\hat{F}$ has rank $h-1$.
Thus, to satisfy the fairness criteria, we only need to take the first $h-1$ columns of $\hat{F}$, which is equivalent to F, so that $F^T H = 0 \in \mathbb{R}^{(h-1)\times k}$

Therefore,

$$
\begin{aligned}
F^T H &= \begin{bmatrix} f^{(1)} - \frac{|V_1|}{|V|} \cdot \mathbf{1}_n \\ f^{(2)} - \frac{|V_2|}{|V|} \cdot \mathbf{1}_n \\ \vdots \\ f^{(h-1)} - \frac{|V_{h-1}|}{|V|} \cdot \mathbf{1}_n \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \dots & \mathbf{h}_k \end{bmatrix} \\
&= \begin{bmatrix} \sum_{i=1}^{n} (f_i^{(1)} - \frac{|V_1|}{|V|}) h_{i1} & \dots & \sum_{i=1}^{n} (f_i^{(1)} - \frac{|V_1|}{|V|}) h_{ik} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} (f_i^{(h-1)} - \frac{|V_{h-1}|}{|V|}) h_{i1} & \dots & \sum_{i=1}^{n} (f_i^{(h-1)} - \frac{|V_{h-1}|}{|V|}) h_{ik} \end{bmatrix} \\
&= 0 \in \mathbb{R}^{(h-1) \times k}
\end{aligned}
$$

is equivalent to the group fairness criteria. □

Example 10.4. *In Example 10.2, we show that $C_1 = \{v_1, v_2, v_6, v_7\}$ and $C_2 = \{v_3, v_4, v_5, v_8\}$ are statistically fair by the group fairness criteria. We want to show that the backward direction of Lemma 10.1 holds. The degree of each vertex v_i is:*

$$
\begin{aligned}
d_1 = 23, d_2 = 25, d_3 = 13, d_4 = 12 \\
d_5 = 7, d_6 = 2, d_7 = 10, d_8 = 12
\end{aligned}
$$

Then the volumes of cluster C_1 and C_2 are:

$$
\begin{aligned}
vol(C_l) = d_1 + d_2 + d_6 + d_7 = 60 \\
vol(C_2) = d_3 + d_4 + d_5 + d_8 = 44
\end{aligned}
$$

Now we can construct the vectors $f^{(s)}, s = 1, 2$ and $\mathbf{h}_l, l = 1, 2$

$$
f^{(1)} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, f^{(2)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \mathbf{h}_1 = \begin{bmatrix} 1/\sqrt{60} \\ 1/\sqrt{60} \\ 0 \\ 0 \\ 0 \\ 1/\sqrt{60} \\ 1/\sqrt{60} \\ 0 \end{bmatrix}, \mathbf{h}_2 = \begin{bmatrix} 0 \\ 0 \\ 1/\sqrt{29} \\ 1/\sqrt{29} \\ 1/\sqrt{29} \\ 0 \\ 0 \\ 1/\sqrt{29} \end{bmatrix}
$$

Then

$$
F = \begin{bmatrix} 1/2 \\ -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \\ -1/2 \end{bmatrix}, H = \begin{bmatrix} 1/\sqrt{60} & 0 \\ 1/\sqrt{60} & 0 \\ 0 & 1/\sqrt{29} \\ 0 & 1/\sqrt{29} \\ 0 & 1/\sqrt{29} \\ 1/\sqrt{60} & 0 \\ 1/\sqrt{60} & 0 \\ 0 & 1/\sqrt{29} \end{bmatrix}
$$

We have

$$F^T H = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Therefore, the result is verified.

Example 10.5. *We continue to study Lemma 10.1 from Example 10.3. We see that* $C_1 = \{v_1, v_3, v_5, v_7\}$ *and* $C_2 = \{v_2, v_4, v_6, v_8\}$ *do not form a fair clustering by the group fairness criteria.*
In this case, the volumes of C_1 *and* C_2 *are:*

$$vol(C_1) = d_1 + d_3 + d_5 + d_7 = 53$$
$$vol(C_2) = d_2 + d_4 + d_6 + d_8 = 51$$

Then we can form H*:*

$$H = \begin{bmatrix} 1/\sqrt{53} & 0 \\ 0 & 1/\sqrt{29} \\ 1/\sqrt{53} & 0 \\ 0 & 1/\sqrt{29} \\ 1/\sqrt{53} & 0 \\ 0 & 1/\sqrt{29} \\ 1/\sqrt{53} & 0 \\ 0 & 1/\sqrt{29} \end{bmatrix}$$

Then

$$F^T H = \begin{bmatrix} 0.2747 & -0.3714 \end{bmatrix} \neq \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Therefore, the result is verified.

10.3 Consistency Analysis

$F^T H = 0 \in \mathbb{R}^{(h-1)\times k}$ implies that the columns of H are in $null(F^T)$. We would like to find out the relationship between n, k, and h so that $null(F^T)$ is sufficiently large to provide a non-trivial H, i.e., all the columns of H are non-zero.

Proposition 10.2. *We have a non-trivial* $H \in \mathbb{R}^{n\times k}$ *if and only if* $n - h + 1 \geq k$.

Proof. ($\Longrightarrow$) Suppose that we have a non-trivial $H \in \mathbb{R}^{n\times k}$.
Then $nullity(F^T) \geq k$.
Recall that by the rank-nullity theorem, for any $A \in \mathbb{R}^{m\times n}$, $rank(A) + nullity(A) = rank(A^T) + nullity(A) = n$.
By Proposition 10.1, F has full rank and $rank(F) = h - 1$. Then $rank(F^T) = rank(F) = h - 1$.
Then $nullity(F^T) = n - h + 1$.
Thus, $n - h + 1 \geq k$.

($\Longleftarrow$) Suppose that $n - h + 1 \geq k$.
Then $nullity(F^T) \geq k$.
Thus, we have a non-trivial H. □

Example 10.6. *In Example 10.4, we find* H *such that* $F^T H = 0 \in \mathbb{R}^{(h-1)\times k}$. *We want to verify that* $n - h + 1 \geq k$.
Here $n = 8$, $h = 2$, $k = 2$.
Then $n - h + 1 = 7 \geq k = 2$.
Therefore, the result is verified.

11 Normalized Spectral Clustering with Group Fairness Constraints

In this section, we incorporate the group fairness criteria into the normalized spectral clustering, do the consistency analysis, and provide a numerical experiment.

Recognition: This section is based on the Appendix of [7].

11.1 Model

Let $F \in \mathbb{R}^{n\times(h-1)}$ consists of $h-1$ group-membership vectors defined by Equation (10.4) and $H \in \mathbb{R}^{n\times k}$ consists of k normalized cluster indicating vectors defined by Equation (8.1). By Lemma 10.1, $F^TH = 0 \in \mathbb{R}^{(h-1)\times k}$ implies the fairness criteria.

Therefore, to get the normalized spectral clustering that achieves group fairness criteria, we have to solve

$$\min_{H \text{ is of form (8.1)}} Tr(H^TLH) \text{ subject to } H^TDH = I \text{ and } F^TH = 0 \tag{11.1}$$

By relaxation, we have

$$\min_{H\in\mathbb{R}^{n\times k}} Tr(H^TLH) \text{ subject to } H^TDH = I \text{ and } F^TH = 0 \tag{11.2}$$

11.2 Numerical Experiment

Example 11.1. *We continue Example 10.4 by doing its numerical version. The coordinates of the data points and the data groups are the same as those in the example. We see that the normalized spectral clustering with group fairness constraints successfully finds* 2 *statistically fair clusters:* $C_1 = \{v_1, v_2, v_6, v_7\}$ *and* $C_2 = \{v_3, v_4, v_5, v_8\}$. *On the other hand, if we use the normalized spectral clustering without group fairness constraints, then each cluster finds exactly* V_1 *and* V_2*, which results in unfair clustering by the group fairness criteria.*

The MATLAB code

```
clear;

X = [0.6 0.4; 0 -2; 2.4 0; 3.0 -1.5;
    1.6 1; 1.5 -1.3; -0.2 0.6; 1.9 -2.2];

v = [1 2 1 2 1 2 1 2];

epsilon = 2.8; k = 2;
adj = similarity(X,epsilon);

f1 = [1 0 1 0 1 0 1 0]';
F = [f1];
n = size(F,1); l = size(F,2);
for i = 1:l
    F(:,i) =  F(:,i) - sum(F(:,i))/n  * ones(n,1);
end

clusterLabels1 = Fair_SC_normalized(adj,k,F);
clusterLabels2 = spectralcluster(X,2);
```

```
figure
gscatter(X(:,1),X(:,2),v,"k","x*",[20 20])
X1 = X(:,1); X2 = X(:,2);
hold on
scatter(X1(clusterLabels1 == 1), X2(clusterLabels1 == 1),400,"r","s",'LineWidth',4);
scatter(X1(clusterLabels1 == 2), X2(clusterLabels1 == 2),400,"b","o",'LineWidth',4);
lgd = legend("group 1","group 2","cluster 1","cluster 2");
lgd.FontSize = 12;
title1 = title("Normalized spectral clustering with group fairness constraints");
title1.FontSize = 18;
hold off


gscatter(X(:,1),X(:,2),v,"k","x*",[20 20])
hold on
X1 = X(:,1); X2 = X(:,2);
scatter(X1(clusterLabels2 == 1), X2(clusterLabels2 == 1),400,"r","s",'LineWidth',4);
scatter(X1(clusterLabels2 == 2), X2(clusterLabels2 == 2),400,"b","o",'LineWidth',4);
lgd = legend("group 1","group 2","cluster 1","cluster 2");
lgd.FontSize = 12;
title2 = title("Normalized spectral clustering without group fairness constraints");
title2.FontSize = 17;
hold off

print -depsc newfigure1
```

prints the following

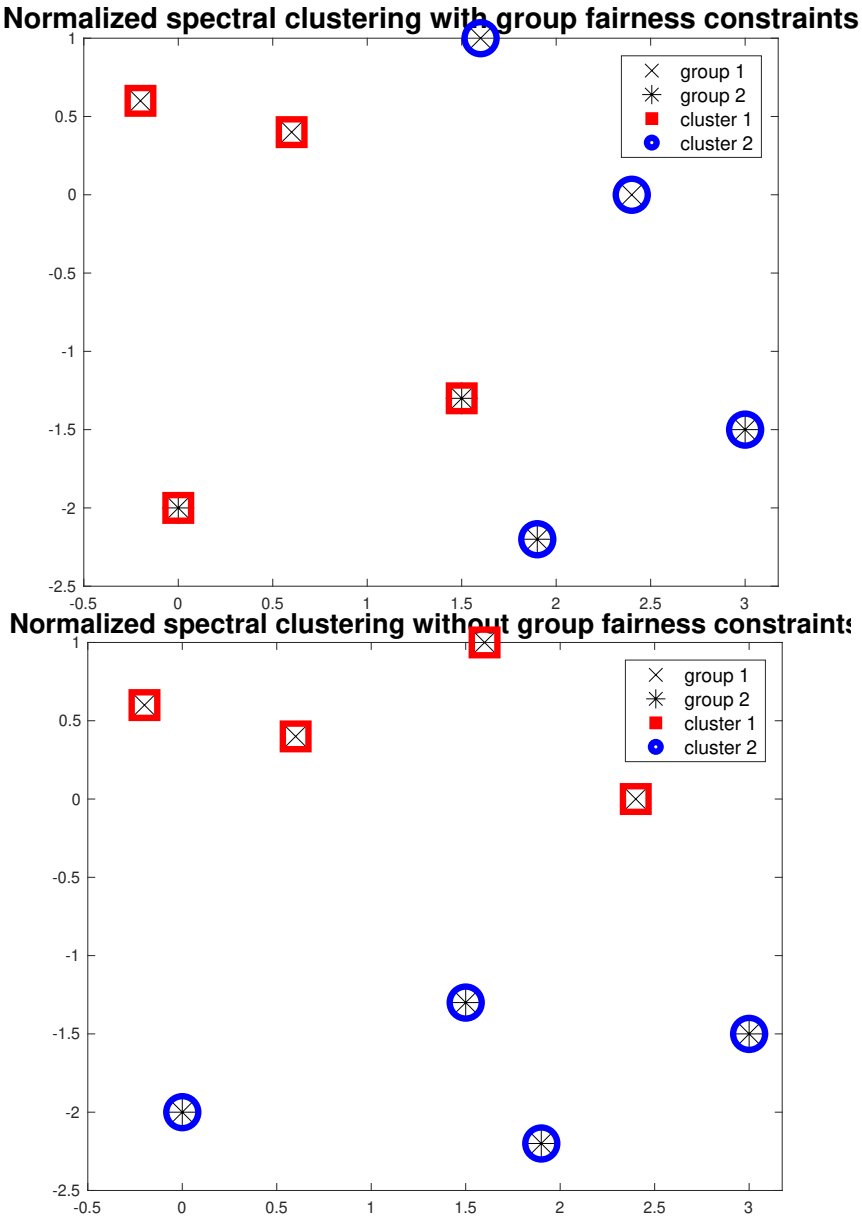

Figure 12: Normalized spectral clustering with/without group fairness constraints

12 Individual Fairness

By [5], fairness should be dealt with not only at the group level but also at the individual level. It states that group fairness criteria fails to consider individuals' interests. Consequently, it proposes another fairness criteria using the concept of the representation graph instead of using the group-membership vectors. Thus, we need to study the individual fairness criteria, in both the set notation and the matrix notation.

12.1 Definition

Definition 12.1. *([5]) A **representation graph** $\mathcal{R} = (V, E_R)$ is a graph that contains a node for each individual in the data set and two nodes are connected if they are similar with respect to sensitive attributes or represent each other's interests.*

Remark 12.1. *([5]) "The representation graph represents how similar the individuals are with respect to some sensitive attributes."*

We use $R \in \{0,1\} \in \mathbb{R}^{n \times n}$ to denote the **adjacency matrix** of $\mathcal{R}$ where $n = |V|$.

Relationship between $\mathcal{R} = (V, E_R)$ and $G = (V, E_G)$ ([5] & [6]) G denotes a similarity graph based on which clusters to be discovered, and $\mathcal{R}$ is defined on the same set of vertices as G and has edges specifying the relationship between the vertices. For example, G may be an ϵ-neighborhood similarity graph defined by Definition 5.1, and $\mathcal{R}$ may be a relationship graph based on the the interactions between individuals.

We assume that G and $\mathcal{R}$ are undirected, and G additionally does not have self-loops. Thus, both A and R are symmetric, and $a_{ii} = 0$ for every $i = 1, \ldots, n$.

Definition 12.2. *([5]) Given a representation graph $\mathcal{R} = (V, E_R)$ and $G = (V, E_G)$, we define the **individual fairness criteria** as follows:*

- *a vertex v_i finds clusters $C_1, \ldots, C_k$ individually fair if all its adjacent vertices in $\mathcal{R}$ are represented proportionally in each cluster. That is,*

$$\frac{|\{v_j \mid r_{ij} = 1 \wedge v_j \in C_l\}|}{|C_l|} = \frac{|\{v_j \mid r_{ij} = 1\}|}{|V|}, \text{ for every } l \in \{1, \ldots, k\} \tag{12.1}$$

- *a clustering $C = C_1 \cup \ldots C_k$ is individually fair with respect to a representation graph $\mathcal{R}$ if:*

$$\text{for every } i \in \{1, \ldots, n\}, \frac{|\{v_j \mid r_{ij} = 1 \wedge v_j \in C_l\}|}{|C_l|} = \frac{|\{v_j \mid r_{ij} = 1\}|}{|V|}, \text{ for every } l \in \{1, \ldots, k\} \tag{12.2}$$

Example 12.1. *Suppose that there are 6 students called Alex, Ben, Carl, Daniel, Ethan, and Felix. They want to build two 3-player soccer teams to play together. Alex and Felix are friends of nobody. Ben is a friend of Carl and Daniel. Carl is a friend of Ben and Ethan. Daniel is a friend of Ben and Ethan. Ethan is a friend of Carl and Daniel. Let $C_1 = \{team\ 1\}$ and $C_2 = \{team\ 2\}$.*

Then, to build the team fairly from the perspective of individuals, we want that for every student, half of his friends are in team 1, and the other half are in team 2.

$$C_1 = \{Ben, Carl, Felix\}, C_2 = \{Alex, Daniel, Ethan\}$$

A possible graph $G = (V, E_G)$ can be the following:

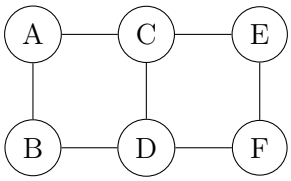

Figure 13: Graph G

An unweighted representation graph $\mathcal{R} = (V, E_R)$ with respect to the relationship of the students is shown below:

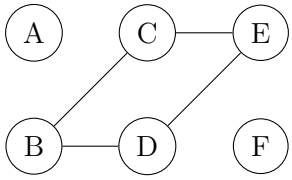

Figure 14: Team 1 and Team 2 given the friend relationship graph

We can check that:
For Alex,

$$\frac{|\{v_j \mid r_{1j} = 1 \wedge v_j \in C_1\}|}{|C_1|} = 0 = \frac{|\{v_j \mid r_{1j} = 1\}|}{|V|}, \frac{|\{v_j \mid r_{1j} \wedge v_j \in C_2\}|}{|C_2|} = 0 = \frac{|\{v_j \mid r_{1j} = 1\}|}{|V|}$$

For Ben,

$$\frac{|\{v_j \mid r_{2j} = 1 \wedge v_j \in C_1\}|}{|C_1|} = \frac{1}{3} = \frac{|\{v_j \mid r_{2j} = 1\}|}{|V|}, \frac{|\{v_j \mid r_{2j} \wedge v_j \in C_2\}|}{|C_2|} = \frac{1}{3} = \frac{|\{v_j \mid r_{2j} = 1\}|}{|V|}$$

For Carl,

$$\frac{|\{v_j \mid r_{3j} = 1 \wedge v_j \in C_1\}|}{|C_1|} = \frac{1}{3} = \frac{|\{v_j \mid r_{3j} = 1\}|}{|V|}, \frac{|\{v_j \mid r_{3j} \wedge v_j \in C_2\}|}{|C_2|} = \frac{1}{3} = \frac{|\{v_j \mid r_{3j} = 1\}|}{|V|}$$

For David,

$$\frac{|\{v_j \mid r_{4j} = 1 \wedge v_j \in C_1\}|}{|C_1|} = \frac{1}{3} = \frac{|\{v_j \mid r_{4j} = 1\}|}{|V|}, \frac{|\{v_j \mid r_{4j} \wedge v_j \in C_2\}|}{|C_2|} = \frac{1}{3} = \frac{|\{v_j \mid r_{4j} = 1\}|}{|V|}$$

For Ethan,

$$\frac{|\{v_j \mid r_{5j} = 1 \wedge v_j \in C_1\}|}{|C_1|} = \frac{1}{3} = \frac{|\{v_j \mid r_{5j} = 1\}|}{|V|}, \frac{|\{v_j \mid r_{5j} \wedge v_j \in C_2\}|}{|C_2|} = \frac{1}{3} = \frac{|\{v_j \mid r_{5j} = 1\}|}{|V|}$$

For Felix,

$$\frac{|\{v_j \mid r_{6j} = 1 \wedge v_j \in C_1\}|}{|C_1|} = 0 = \frac{|\{v_j \mid r_{6j} = 1\}|}{|V|}, \frac{|\{v_j \mid r_{6j} \wedge v_j \in C_2\}|}{|C_2|} = 0 = \frac{|\{v_j \mid r_{6j} = 1\}|}{|V|}$$

Therefore, team 1 and team 2 are individually fair.

Example 12.2. *We continue to study Example 12.1. This time we give a counterexample of individual fairness. Suppose that we build the two teams with the following:*

$$C_1 = \{Alex, Ben, Felix\}, C_2 = \{Carl, Daniel, Ethan\}$$

Then these two teams are not individually fair for everyone.
For example, both Ben's friends are in team 2, and he has no friend in team 1, which is not fair to him. That is,

$$\frac{|\{v_j \mid r_{2j} = 1 \wedge v_j \in C_1\}|}{|C_1|} = 0 \neq \frac{|\{v_j \mid r_{1j} = 1\}|}{|V|} = \frac{2}{6}, \frac{|\{v_j \mid r_{1j} \wedge v_j \in C_2\}|}{|C_2|} = \frac{2}{3} \neq \frac{|\{v_j \mid r_{1j} = 1\}|}{|V|} = \frac{2}{6}$$

Therefore, team 1 and team 2 are not individually fair.

12.2 Relationship between Individual Fairness and Group Fairness

Individual fairness vs. group fairness ([5]) A representation graph $\mathcal{R} = (V, E_R)$ where $r_{ij} = 1$ if and only if v_i and v_j belong to the same data group reduces the fairness criteria from individual fairness to group fairness.

Example 12.3. *The purpose of this example is to identify the difference between group fairness and individual fairness. Suppose that we are given the following representation graph $\mathcal{R}$. We have the following:*

$$V_1 = \{v_1, v_3, v_5, v_7\}, V_2 = \{v_2, v_4, v_6, v_8\}$$

A possible graph $G = (V, E_G)$ can be the following:

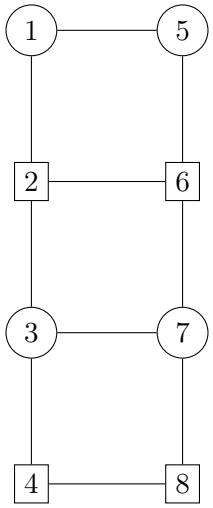

Figure 15: Graph $G = (V, E_G)$

The representation graph $\mathcal{R} = (V, E_R)$ is the following:

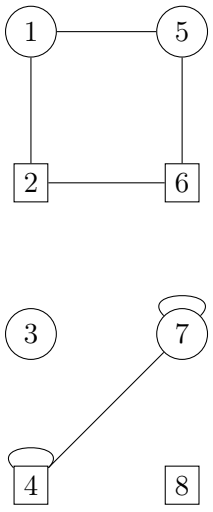

Figure 16: Representation graph $\mathcal{R}$

- *Following the group fairness criteria, we can choose* $C_1 = \{v_1, v_2, v_5, v_6\}$ *and* $C_2 = \{v_3, v_4, v_7, v_8\}$.
  We first check that the clustering satisfies Equation (10.2).

$$\frac{|V_1|}{|V|} = \frac{4}{8}, \frac{|V_2|}{|V|} = \frac{4}{8}$$

  For C_1,

$$\frac{|V_1 \cap C_1|}{|C_1|} = \frac{2}{4} = \frac{|V_1|}{|V|}, \frac{|V_2 \cap C_1|}{|C_1|} = \frac{2}{4} = \frac{|V_2|}{|V|}$$

  For C_2,

$$\frac{|V_1 \cap C_2|}{|C_2|} = \frac{2}{4} = \frac{|V_1|}{|V|}, \frac{|V_2 \cap C_2|}{|C_2|} = \frac{2}{4} = \frac{|V_2|}{|V|}$$

  Therefore, it is a statistically fair clustering.
  However, according to individual fairness, such a clustering $C = C_1 \cup C_2$ *is not fair from the perspective of vertices.*
  For example, both of adjacent vertices of v_1 *are in* C_1*, and neither of them is in* C_2*. That is,*

$$\frac{|\{v_j \mid r_1 j \wedge v_j \in C_1\}|}{|C_1|} = \frac{2}{4} \neq \frac{|\{v_j \mid r_{1j} = 1\}|}{|V|} = \frac{2}{8}$$

  Therefore, v_1 *does not have enough representation in* C_2.
  Therefore, it is not a fair clustering by individual fairness.

- *We now change our clustering so that* $C_1 = \{v_1, v_4, v_5, v_8\}$ *and* $C_2 = \{v_2, v_3, v_6, v_7\}$.
  We first check

$$\frac{|V_1|}{|V|} = \frac{4}{8}, \frac{|V_2|}{|V|} = \frac{4}{8}$$

  For C_1,

$$\frac{|V_1 \cap C_1|}{|C_1|} = \frac{2}{4} = \frac{|V_1|}{|V|}, \frac{|V_1 \cap C_1|}{|C_1|} = \frac{2}{4} = \frac{|V_2|}{|V|}$$

  For C_2,

$$\frac{|V_1 \cap C_2|}{|C_2|} = \frac{2}{4} = \frac{|V_1|}{|V|}, \frac{|V_1 \cap C_2|}{|C_2|} = \frac{2}{4} = \frac{|V_2|}{|V|}$$

  Hence, it is a fair clustering by the group fairness criteria.
  It further satisfies the individual fairness criteria.
  For v_1,

$$\frac{|\{v_j \mid r_{1j} = 1 \wedge v_j \in C_l\}|}{|C_l|} = \frac{1}{4} = \frac{|\{v_j \mid r_{1j} = 1\}|}{|V|}, l \in \{1, 2\}$$

  For v_2,

$$\frac{|\{v_j \mid r_{2j} = 1 \wedge v_j \in C_l\}|}{|C_l|} = \frac{1}{4} = \frac{|\{v_j \mid r_{2j} = 1\}|}{|V|}, l \in \{1, 2\}$$

  It is easy to see that $v_3, \ldots, v_8$ *all finds clusters* C_1 *and* C_2 *individually fair.*
  Therefore, clusters C_1, C_2 *are individually fair with respect to this representation graph* $\mathcal{R}$.
  Therefore, such a clustering is fair by both the group fairness criteria and the individual fairness criteria.

12.3 Matrix Representation

Lemma 12.1. *([5]) Let $n = |V|$ and $H \in \mathbb{R}^{n\times k}$ be of form (8.1). Let $J = I - \frac{1}{|V|}\mathbf{1}\mathbf{1}^T$ where $\mathbf{1} \in \mathbb{R}^n$ is an all-ones vector. Let $R \in \mathbb{R}^{n\times n}$ be the adjacency matrix of the given representation graph $\mathcal{R}$. Then*

$$RJH = 0 \in \mathbb{R}^{n\times k} \tag{12.3}$$

is equivalent to the individual fairness criteria (Equation (12.2)).

Proof. Fix any arbitrary node $v_i \in V$ and $l \in \{1, \ldots, k\}$. Recall that H contains the normalized cluster that indicates vectors such that

$$h_{il} = \begin{cases} \frac{1}{\sqrt{|vol(C_l)|}}, \text{ if } i \in C_l \\ 0, \text{ if } i \notin C_l \end{cases}$$

Note that $RH = \sum_{j=1}^{n} r_{ij}h_{jl}$ and $R(I - \mathbf{1}\mathbf{1}^T/n)H = \frac{1}{n}(\sum_{j=1}^{n} r_{ij})(\sum_{j=1}^{n} h_{jl})$.
By Definition 12.2 and Equation (8.1), $r_{ij}h_{jl} = \frac{1}{\sqrt{|vol(C_l)|}}$ if $v_j \in \{v_j \mid r_{ij} = 1 \wedge v_j \in C_l\}$ and 0 otherwise.
By Equation (8.1), $\sum_{j=1}^{n} h_{jl} = \sum_{v_i \in C_l} \frac{1}{\sqrt{|vol(C_l)|}} = \frac{|C_l|}{\sqrt{|vol(C_l)|}}$.
Combining the above, we have

$$\sum_{j=1}^{n} r_{ij}h_{jl} = \frac{1}{n}(\sum_{j=1}^{n} r_{ij})(\sum_{j=1}^{n} h_{jl})$$

is equivalent to

$$\frac{1}{\sqrt{|vol(C_l)|}} |\{v_j \mid r_{ij} = 1 \wedge v_j \in C_k\}| = \frac{1}{n} |\{v_j \mid r_{ij} = 1\}| \frac{|C_l|}{\sqrt{|vol(C_l)|}}$$

Since this holds for any arbitrary $v_i \in V$ and $l \in \{1, \ldots, k\}$,

$$R(I - \frac{1}{|V|}\mathbf{1}\mathbf{1}^T)H = 0 \in \mathbb{R}^{n\times k} \iff \forall v_i \in V, \frac{|\{v_j \mid r_{ij} = 1 \wedge v_j \in C_l\}|}{|C_l|} = \frac{|\{v_j \mid r_{ij} = 1\}|}{|V|}, l \in \{1, \ldots, k\}$$

□

Example 12.4. *In Example 12.1, we show that $C_1 = \{Ben, Carl, Felix\}, C_2 = \{Alex, Daniel, Ethan\}$ are individually fair. We verify that the clusters satisfy the backward direction of Lemma 12.1. The adjacency matrix R of the representation graph $\mathcal{R}$ is*

$$R = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The degree of each vertex v_i is:

$$d_1 = 0, d_2 = 2, d_3 = 2, d_4 = 2, d_5 = 2, d_6 = 0$$

Then the volumes of cluster C_1 and C_2 are:

$$vol(C_1) = d_2 + d_3 + d_6 = 4, vol(C_2) = d_1 + d_4 + d_5 = 4$$

Then the matrix H containing the normalized cluster indicating vectors is

$$H = \begin{bmatrix} 0 & 1/2 \\ 1/2 & 0 \\ 1/2 & 0 \\ 0 & 1/2 \\ 0 & 1/2 \\ 1/2 & 0 \end{bmatrix}$$

We have

$$RJH = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Therefore, the result is verified.

Example 12.5. *We continue to verify Lemma 12.1 from Example 12.2. We show that $C_1 = \{Alex, Ben, Felix\}, C_2 = \{Carl, Daniel, Ethan\}$ does not provide an individually fair clustering. That is, suppose*

$$H = \begin{bmatrix} 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 0 \\ 0 & 1/\sqrt{6} \\ 0 & 1/\sqrt{6} \\ 0 & 1/\sqrt{6} \\ 1/\sqrt{2} & 0 \end{bmatrix}$$

Then

$$RJH = \begin{bmatrix} 0 & 0 \\ -0.7071 & 0.4082 \\ 0 & 0 \\ 0 & 0 \\ -0.7071 & 0.4082 \\ 0 & 0 \end{bmatrix} \neq 0 \in \mathbb{R}^{6\times 2}$$

Therefore, the result is verified.

Example 12.6. *We now verify that the clusters satisfy the backward direction of Lemma 12.1 using Example 12.3.*

The adjacency matrix R of the representation graph $\mathcal{R}$ is

$$R = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The degree of each vertex is:

$$d_1 = 2, d_2 = 2, d_3 = 0, d_4 = 2,$$
$$d_5 = 2, d_6 = 2, d_7 = 2, d_8 = 0$$

- *We show that $C_1 = \{v_1, v_2, v_5, v_6\}$ and $C_2 = \{v_3, v_4, v_7, v_8\}$ does not give an individually fair clustering.*
  The volumes of C_1 and C_2 are:

  $$vol(C_1) = d_1 + d_2 + d_5 + d_6 = 8, vol(C_2) = d_3 + d_4 + d_7 + d_8 = 4$$

  Then the matrix H containing the normalized cluster indicating vectors

  $$H = \begin{bmatrix} 1/\sqrt{8} & 0 \\ 1/\sqrt{8} & 0 \\ 0 & 1/2 \\ 0 & 1/2 \\ 1/\sqrt{8} & 0 \\ 1/\sqrt{8} & 0 \\ 0 & 1/2 \\ 0 & 1/2 \end{bmatrix}$$

  Then

  $$RJH = \begin{bmatrix} 0.3536 & -0.5 \\ 0.3536 & -0.5 \\ 0 & 0 \\ -0.3536 & 0.5 \\ 0.3536 & -0.5 \\ 0.3536 & -0.5 \\ -0.3536 & 0.5 \\ 0 & 0 \end{bmatrix} \neq 0 \in \mathbb{R}^{8\times 2}$$

- *In contrast, we show that $C_1 = \{v_1, v_4, v_5, v_8\}$ and $C_2 = \{v_2, v_3, v_6, v_7\}$ form an individually fair clustering.*
  The volumes of C_1 and C_2 are:

  $$vol(C_1) = d_1 + d_4 + d_5 + d_8 = 6,$$
  $$vol(C_2) = d_2 + d_3 + d_6 + d_7 = 6$$

Then the matrix H is

$$H = \begin{bmatrix} 1/\sqrt{6} & 0 \\ 0 & 1/\sqrt{6} \\ 0 & 1/\sqrt{6} \\ 1/\sqrt{6} & 0 \\ 1/\sqrt{6} & 0 \\ 0 & 1/\sqrt{6} \\ 0 & 1/\sqrt{6} \\ 1/\sqrt{6} & 0 \end{bmatrix}$$

We have

$$RJH = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Therefore, the result is verified.

12.4 Consistency Analysis

$RJH = 0 \in \mathbb{R}^{n\times k}$ implies that the columns of H are in $null(RJ)$. We want that $null(RJ)$ to be sufficient large so that we get a non-trivial $H \in \mathbb{R}^{n\times k}$, i.e., all the columns of H are non-zero. Hence, we need to figure out the rank condition of R.

Proposition 12.1. *$rank(J) = n - 1$*

Proof.

$$J = \begin{bmatrix} 1-\frac{1}{n} & -\frac{1}{n} & \dots & -\frac{1}{n} & -\frac{1}{n} \\ -\frac{1}{n} & 1-\frac{1}{n} & \dots & -\frac{1}{n} & -\frac{1}{n} \\ \vdots & & & & \\ -\frac{1}{n} & -\frac{1}{n} & \dots & 1-\frac{1}{n} & -\frac{1}{n} \\ -\frac{1}{n} & -\frac{1}{n} & \dots & -\frac{1}{n} & 1-\frac{1}{n} \end{bmatrix}$$

Then

$$J\mathbf{1}_n = I\mathbf{1} - \frac{1}{n}\mathbf{1}(\mathbf{1}^T\mathbf{1}) = \mathbf{1} - \frac{n}{n}\mathbf{1} = \mathbf{0}_n$$

Therefore, $J\mathbf{1}_n = \mathbf{0}_n$.
Therefore, $nullity(J) \geq n - 1$.
Let M represent the first $n - 1$ columns of J.
We want to show that M has full column rank. That is, $M\mathbf{x} = \mathbf{0}_n \iff \mathbf{x} = \mathbf{0}_{n-1}$.
Suppose that for some $\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_{n-1} \end{bmatrix}^T \in \mathbb{R}^{(n-1)}$, $M\mathbf{x} = \mathbf{0}_n$.
Then looking at the ith row of both sides, we have:
for $i = 1, \dots, n-1$:

$$x_i - \frac{1}{n}\sum_{s=1}^{n-1} x_s = 0$$

for $i = n$:

$$-\frac{1}{n}\sum_{s=1}^{n-1} x_s = 0$$

From the last block, $-\frac{1}{n}\sum_{s=1}^{n-1} x_s = 0$. We can plug it into the other blocks and recursively get $x_1 = \cdots = x_{n-1} = 0$.
Thus, $\mathbf{x} = \mathbf{0}_{n-1}$.
Conversely, if $\mathbf{x} = \mathbf{0}_{n-1}$, then $M\mathbf{x} = \mathbf{0}_n$.
Therefore, $M\mathbf{x} = \mathbf{0}_n \iff \mathbf{x} = \mathbf{0}_{n-1}$.
Therefore, M has full rank.
Therefore, $rank(J) = n - 1$. □

Recall that:

- By the rank-nullity theorem, if $A \in \mathbb{R}^{m\times n}$, then $rank(A) + nullity(A) = rank(A^T) + nullity(A) = n$.
- By the upper bound on the rank of matrix multiplication, if $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times k}$, then $rank(AB) \leq \min\{rank(A), rank(B)\}$.
- By the Sylvester's rank inequality, if $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times k}$, then $rank(A)+rank(B)-n \leq rank(AB)$.

Proposition 12.2. *Suppose that we get a non-trivial $H \in \mathbb{R}^{n\times k}$. Then $rank(R) \leq n - k + 1$.*

Proof. Let $H \in \mathbb{R}^{n\times k}$ be non-trivial.
Since the columns of $H \in \mathbb{R}^{n\times k}$ are in $null(RJ)$ where $RJ \in \mathbb{R}^{n\times n}$, we have $nullity(RJ) \geq k$.
By the rank-nullity theorem, $rank(RJ) + nullity(RJ) = rank((RJ)^T) + nullity(RJ) = n$, we have $rank(RJ) \leq n - k$.
By the upper bound on the rank of matrix multiplication, we have:

$$rank(RJ) \leq \min\{rank(R), rank(J\} = \min\{rank(R), n - 1\}$$

By the Sylvester's rank inequality, the lower bound on $rank(RJ)$ is

$$rank(R) + rank(J) - n = rank(R) - 1 \leq rank(RJ)$$

Combining the above, to have a non-trivial $H \in \mathbb{R}^{n\times k}$, we want:

$$rank(R) - 1 \leq rank(RJ) \leq \min\{rank(R), n - 1\}$$

Therefore, $rank(R) \leq n - k + 1$. □

Example 12.7. *In both Example 12.4 and Example 12.6, we can find $H \in \mathbb{R}^{n\times k}$ subject to $RJH = 0 \in \mathbb{R}^{n\times k}$. We want to check that in both cases, $rank(R) \leq n - k + 1$.*

In Example 12.4, since we have $n = 6$ data points and $k = 2$ individually fair clusters, we should expect that $rank(R) \leq n - k + 1 = 5$.
Indeed, $rank(R) = 2$.
Thus, it satisfies Proposition 12.2.

In Example 12.6, since we have $n = 8$ data points and $k = 2$ individually fair clusters, we should expect that $rank(R) \leq n - k + 1 = 7$.
Indeed, $rank(R) = 3$.
Thus, it also satisfies Proposition 12.2.

Proposition 12.3. *$rank(R) \leq n - k + 1$ does not guarantee that we have a non-trivial H.*

Example 12.8. *Suppose that we are given the adjacency matrix R of a representation graph $\mathcal{R}$ as the following:*

$$R = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Since $R = R^T$, R is a reasonable adjacency matrix of some representation graph.
We would like to find $k = 2$ individually clusters.
This is equivalent to finding a non-trivial $H \in \mathbb{R}^{4\times 2}$.
Since $rank(R) = 3$, $rank(R) \leq n - k + 1 = 4 - 2 + 1 = 3$.
However,

$$RJ = \begin{bmatrix} 0.5 & 0.5 & -0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 & -0.5 \\ -0.5 & 0.5 & 0.5 & -0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Thus, $rank(RJ) = 3$, which implies that $nullity(RJ) = 1 < k = 2$.
Therefore, there does not exist a non-trivial $H \in \mathbb{R}^{4\times 2}$.

Remark 12.2. *In practice, we want R to be sparse and to have low rank in order to get a non-trivial H.*

13 Normalized Spectral Clustering with Individual Fairness Constraints

In this section, we incorporate the individual fairness criteria into the normalized spectral clustering and do the consistency analysis.

Recognition: This section is based on section 3 of [5].

13.1 Model

Let $R \in \mathbb{R}^{n\times n}$ be the adjacency matrix of $\mathcal{R} = (G, E_R)$ and $H \in \mathbb{R}^{n\times k}$ consists of k normalized cluster indicating vectors defined by Equation (8.1). By Lemma 12.1, $RJH = 0 \in \mathbb{R}^{n\times k}$ where $J = I - \frac{1}{|V|}\mathbf{1}\mathbf{1}^T$ implies the individual fairness criteria.

Therefore, to get the normalized spectral clustering with individual fairness constraints, we have to solve

$$\min_{H \text{ is of form (8.1)}} Tr(H^T LH) \text{ subject to } H^T DH = I \text{ and } RJH = 0 \tag{13.1}$$

By relaxation, we have

$$\min_{H\in\mathbb{R}^{n\times k}} Tr(H^T LH) \text{ subject to } H^T DH = I \text{ and } RJH = 0 \tag{13.2}$$

13.2 Numerical Experiment

Example 13.1. *We continue Example 12.4 by doing a numerical experiment. We use the same representation graph $\mathcal{R}$ as in the example. We see that the normalized spectral clustering with individual fairness constraints provides the same clustering as in the example.*

The MATLAB code

```
clear;

X = [-1.5 1.5; -1.5 0; 0 1.5;
    0 0; 1.5 1.5; 1.5 0];

epsilon = 2; k = 2;
adj = similarity(X,epsilon);

R = [0 0 0 0 0 0;
    0 0 1 1 0 0;
    0 1 0 0 1 0;
    0 1 0 0 1 0;
    0 0 1 1 0 0;
    0 0 0 0 0 0];
J = eye(6) - 1/6 * ones(6,1) * ones(1,6);

clusterLabels1 = Fair_SC_normalized_individual(adj,k,R* J);

figure
hold on
X1 = X(:,1); X2 = X(:,2);
scatter(X1(clusterLabels1 == 1), X2(clusterLabels1 == 1),400,"r","s",'LineWidth',4);
scatter(X1(clusterLabels1 == 2), X2(clusterLabels1 == 2),400,"b","o",'LineWidth',4);
lgd = legend("cluster 1","cluster 2");
lgd.FontSize = 12;
title2 = title("Normalized spectral clustering with individual fairness constraints");
title2.FontSize = 15;
hold off
print -depsc newfigure
```

prints the following

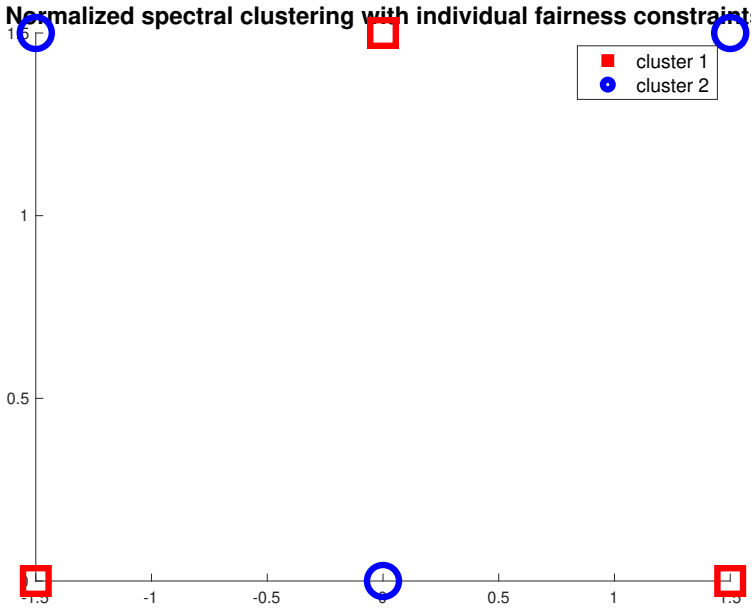

Figure 17: Normalized spectral clustering with individual fairness constraints

Example 13.2. *We can do the numerical version of Example 12.6. We use the same representation graph $\mathcal{R}$ as in the example. The normalized spectral clustering with individual fairness constraints provides the same clustering as in the example.*

The MATLAB code

```
clear;

X = [-2 2; -2 0; -2 -2; -2 -4 ;0 2; 0 0; 0 -2; 0 -4];

epsilon = 5; k = 2;
adj = similarity(X,epsilon)

R = [0 1 0 0 1 0 0 0;
    1 0 0 0 0 1 0 0;
    0 0 0 0 0 0 0 0;
    0 0 0 1 0 0 1 0;
    1 0 0 0 0 1 0 0;
    0 1 0 0 1 0 0 0;
    0 0 0 1 0 0 1 0;
    0 0 0 0 0 0 0 0];
J = eye(8) - 1/8 * ones(8,1) * ones(1,8);
RJ = R * J;

clusterLabels1 = Fair_SC_normalized_individual(adj,k, RJ);

figure
hold on
X1 = X(:,1); X2 = X(:,2);
```

```
scatter(X1(clusterLabels1 == 1), X2(clusterLabels1 == 1),400,"r","s",'LineWidth',4);
scatter(X1(clusterLabels1 == 2), X2(clusterLabels1 == 2),400,"b","o",'LineWidth',4);
lgd = legend("cluster 1","cluster 2");
lgd.FontSize = 12;
title2 = title("Normalized spectral clustering with individual fairness constraints");
title2.FontSize = 15;
hold off

print -depsc newfigure
```

prints the following

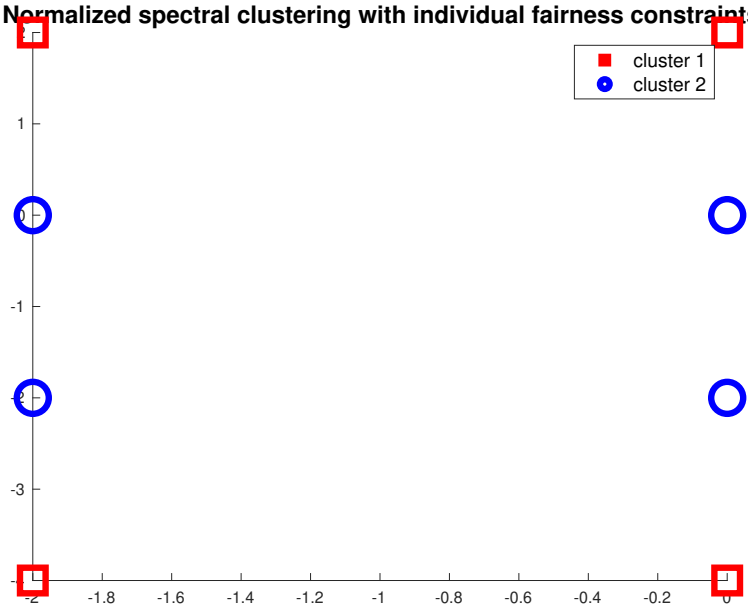

Figure 18: Normalized spectral clustering with individual fairness constraints

References

[1] Anagnostopoulos, Aris, et al. *Principal Fairness: Removing Bias via Projections.* 2021. https://arxiv.org/abs/1905.13651.

[2] Boyd, Stephen, and Lieven Vandenberghe. *Introduction to Applied Linear Algebra.* 1 ed. Cambridge University Press, 2018, pp. 69-87.

[3] Chierichetti, Flavio, et al. *Fair Clustering Through Fairlets.* Conference on Neural Information Processing Systems, 2017. https://arxiv.org/abs/1802.05733.

[4] Gallier, Jean. *Notes on Elementary Spectral Graph Theory. Applications to Graph Clustering Using Normalized Cuts.* 2013. https://arxiv.org/abs/1311.2492.

[5] Gupta, Shubham, and Ambedkar Dukkipati. *Protecting Individual Interests across Clusters: Spectral Clustering with Guarantees.* 2021. https://arxiv.org/abs/2105.03714.

[6] Gupta, Shubham, and Ambedkar Dukkipati. *On consistency of constrained spectral clustering under representation-aware stochastic block model.* 2022. https://arxiv.org/abs/2203.02005.

[7] Kleindessner, Matthäus, et al. *Guarantees for Spectral Clustering with Fairness Constraints.* Proceedings of the 36th International Conference on Machine Learning, 2019. https://arxiv.org/abs/1901.08668.

[8] Rosen, Kenneth. *Discrete Mathematics and Its Applications.* 8th ed., McGraw Hill, 2018, pp. 641-89.

[9] von Luxburg, Ulrike. *A Tutorial on Spectral Clustering.* Statistics and Computing, 2007. https://arxiv.org/abs/0711.0189.

[10] Prof. Bai's Lecture Notes on Spectral Clustering.

[11] Prof. Bai's Lecture Notes for ECS 122A.

Part IV
Appendix

A Acknowledgement

I would like to express my gratitude to Prof. Zhaojun Bai for guiding me upon this undergraduate thesis. This is my first independent project, in which I not only learn how to do research but also find my research interests. Prof. Bai always provides me with earnest and useful advice. He has also supported me greatly on my academic career path. I appreciate my time working with him.

B K-means Clustering

Now we introduce k-means clustering. This is for two reasons. First, k-means clustering shows another idea of how we can cluster data points. Second, we need k-means clustering to do the spectral clustering.

In this section, we introduce the mathematical foundation of k-means clustering. We also give an algorithm for the k-means clustering.

Recognition: the definitions and results come mainly from Chapter 4 of [2].

Clustering Assignment. Suppose we have N n-vector $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, and we want to cluster them into k groups. We label the groups $1, 2, \ldots, k$ and specify the assignment of the given vectors using $\mathbf{c}_i$ where $\mathbf{c}_i$ is the group to which the vector $\mathbf{x}_i$ is assigned. Let G_j denote the set of indices corresponding to group j such that $G_j = \{i \mid \mathbf{c}_i = j\}$.

Example B.1. Let $N = 7$ and $k = 3$. Then $c = (3, 2, 1, 1, 2, 3, 1)$ implies that group 1 includes $\mathbf{x}_3$, $\mathbf{x}_4$, $\mathbf{x}_7$, group 2 includes $\mathbf{x}_2, \mathbf{x}_5$, and group 3 includes $\mathbf{x}_1, \mathbf{x}_6$. Then $G_1 = \{3, 4, 7\}$, $G_2 = \{2, 4\}$, $G_3 = \{1, 6\}$

Clustering Objective. For each group G_i, we call the group representative $\mathbf{z}_i$, $i = 1, \ldots, k$. If $\mathbf{x}_i$ is in group $G_j, j = \mathbf{c}_i$, then the representative vector $\mathbf{z}_{\mathbf{c}_i}$ should be close to the vectors in the related group, that is, $\|\mathbf{x}_i - \mathbf{z}_{\mathbf{c}_i}\|$ should be small.

We define

$$J^{\text{clust}} = \frac{\|\mathbf{x}_1 - \mathbf{z}_{\mathbf{c}_1}\|^2 + \cdots + \|\mathbf{x}_N - \mathbf{z}_{\mathbf{c}_N}\|^2}{N}$$

which is the mean square distance from the vectors to the representative vector in the same group.

The smaller J^{clust} is, the better the clustering. If $J^{\text{clust}} = 0$, then $\|\mathbf{x_i} - \mathbf{z}_{\mathbf{c}_i}\| = 0$ for every i, which implies that the original vector only takes k distinct values, and every vector $\mathbf{x}_i$ is assigned to the representative with the same norm.

Definition B.1. *A clustering is **optimal** if the given $\mathbf{c}_1, \ldots \mathbf{c}_N$ and $\mathbf{z}_1, \ldots, \mathbf{z}_k$ induces minimal J^{clust}.*

Remark B.1. *The clustering choices found by the k-means clustering are suboptimal which means that they might not give the lowest possible value of J^{clust}.*

Theorem B.1. *Suppose that the representative vectors are fixed. Then assigning each vector to its nearest representative induces the minimal J^{clust}.*

Proof. Let the representative vectors $\mathbf{z}_1, \ldots, \mathbf{z}_k$ be fixed. We want to find the group assignments $\mathbf{c}_1, \ldots, \mathbf{c}_N$ to minimize J^{clust}.

Note that J^{clust} is a sum of N terms. Each choice of $\mathbf{c}_i$ only affects the term $\frac{\left\|\mathbf{x}_i - \mathbf{z}_{\mathbf{c}_i}\right\|^2}{N}$. Then we only need to choose $\mathbf{c}_i$ so that $\|\mathbf{x}_i - \mathbf{z}_{\mathbf{c}_i}\|$ is minimal. Then over $j = 1, \ldots k$, we choose the proper j so that we minimize the $\|\mathbf{x}_i - \mathbf{z}_j\|$.
The mathematical expression of $\|\mathbf{x}_i - \mathbf{z}_{\mathbf{c}_i}\|$ becomes

$$\|\mathbf{x}_i - \mathbf{z}_{\mathbf{c}_i}\| = \min_{j=1,\ldots,k} \|\mathbf{x}_i - \mathbf{z}_j\|$$

The value of J^{clust} becomes the mean of the squared distance from the data vectors to the closest representative.

$$J^{clust} = \frac{\min\limits_{j=1,\ldots,k} \|\mathbf{x}_1 - \mathbf{z}_j\|^2 + \ldots \min\limits_{j=1,\ldots,k} \|\mathbf{x}_N - \mathbf{z}_j\|^2}{N}$$

□

Definition B.2. *The **cluster centroid** is the average of the vectors $\mathbf{x}_i$ assigned to its group.*

$$\mathbf{z}_j = \frac{\sum\limits_{i \in G_j} \mathbf{x}_i}{|G_j|}$$

where $|G_j|$ denotes the number of elements in group G_j.

Example B.2. Suppose group K contains these vectors.

$$\mathbf{x}_1 = \begin{bmatrix} 80 \\ 56 \\ 35 \end{bmatrix}, \mathbf{x}_2 = \begin{bmatrix} 75 \\ 65 \\ 68 \end{bmatrix}, \mathbf{x}_3 = \begin{bmatrix} 93 \\ 45 \\ 66 \end{bmatrix}, \mathbf{x}_4 = \begin{bmatrix} 68 \\ 54 \\ 27 \end{bmatrix}$$

Then $\sum\limits_{i \in G_j} \mathbf{x}_i = \begin{bmatrix} 316 \\ 220 \\ 196 \end{bmatrix}$ and $|G_j| = 4$.

Thus, $\mathbf{z}_1 = \begin{bmatrix} 79 \\ 55 \\ 49 \end{bmatrix}$

Theorem B.2. *Suppose that the group assignments are fixed. Then choosing $\mathbf{z}_j$ to be the centroid of the cluster induces minimal J^{clust}.*

Proof. Let the group assignments $G_i, i = k, \ldots, N$ be fixed. We can consider J^{clust} as a sum of k terms, where each term is the average of the summation of the form $\|\mathbf{x}_i - \mathbf{z}_j\|^2$ for any $i \in G_j$. Then

$$J^{clust} = J_1 + \cdots + J_k$$

where

$$J_j = \frac{\sum\limits_{i \in G_j} \|\mathbf{x}_i - \mathbf{z}_j\|^2}{N}$$

The choice of $\mathbf{z}_j$ only affects the term J_j. We can choose each $\mathbf{z}_j$ to minimize J_j. To do so, we choose $\mathbf{z}_j$ to be the cluster centroid. Thus, if the group assignments are fixed, we choose the representative vector to be the average of the vectors assigned to the group to minimize J^{clust}. □

Now, suppose that neither the group assignment nor the group representatives are fixed. How do we find the minimal J^{clust}? The answer is that we can use k-means clustering.

Algorithm. The k-means clustering requires us to repeatedly alternate between updating the assignment of groups and updating the representatives so that J^{clust} becomes smaller at each step. We stop if the choices of the group representatives and group assignments do not change.

Algorithm 7 k-means clustering

Input: N vectors $\mathbf{x}_1, \ldots \mathbf{x}_N$ and k group representative vectors $\mathbf{z}_1, \ldots, \mathbf{z}_k$

Output: k clusters of the vectors

1: Randomly pick the initial group representatives from the original vectors.
2: For each vector $\mathbf{x}_i$, assign $\mathbf{x}_i$ to the nearest representative vector, $i = 1, \ldots, N$. If there is a tie, we we assign $\mathbf{x}_i$ to the group associated with one of the closest representative with the smallest value of j.
3: For each group G_j, set $\mathbf{z}_j$ to the new cluster centroid, $j = 1, \ldots, k$.
4: Repeat the 2 steps above until the changes of J^{clust} in successive iterations are small.

Remark B.2. *If any of the group assignments becomes empty, we simply drop this group and finish with a partition with fewer than k groups.*

We choose k by this rule: The value of J^{clust} with k is significantly lower than that with $k-1$ but not significantly higher than that with $k+1$. For example, suppose that for some $\mathbf{x}_1, \ldots, \mathbf{x}_N$, we have J^{clust} with $k = 5$ is 10, J^{clust} with $k = 6$ is 9, J^{clust} with $k = 7$ is 5, and J^{clust} with $k = 8$ is 4.5. Then we should choose $k = 7$.

Program. The following codes present how we can apply the k-means algorithm in MATLAB. Each sub-function corresponds to one step of the k-means algorithm.
Here is the MATLAB code.

```
function [idx,C] = mykmeans(X,k,n)
init_centroid = pick_initial_centroids(X,k);
index = find_centroids(X,init_centroid,k);
centroids = compute_centroids(X,index,k);
[idx,C]=repeat_step(X,centroids,n,k);

%step 1: we randomly pick the representatives from the original vectors
function init_centroid = pick_initial_centroids(X,k)
random_number = randperm(size(X,1));
init_centroid = X(random_number(1:k),:);
end

%step 2: compute j where j is the group label to which x_i is assigned
%j = 1,...,k
function index = find_centroids(X,init_centroid,k)
index = [];
for i = 1:length(X)
    for j = 1:k
    dist(j) = norm(X(i,:)-init_centroid(j,:),2);
    end
    newindex = find(dist == min(dist));
    index = [index,newindex];
end
```

```
index = index';
end

%step 3:compute the group representative z_j for each group j = 1,...,k
%z_j is the mean of the vectors x_i in its group
function centroids = compute_centroids(X,index,k)
for i = 1:k
    newindex = find(index == i);
    num = length(newindex);
    centroids(i,:) = sum(X(newindex, :)) / num;
end
end

%step 4: repeat step 1 and step 2 until J^{clust} gets stable
function [newidx,newcentroids1]=repeat_step(X,centroids,n,k)
centroids1 =  centroids;
for i = 1: length(n)
    newidx = find_centroids(X,centroids1,k);
    centroids1 = compute_centroids(X,newidx,k);
    newcentroids1 = centroids1;
end
end
end
```

Lemma B.1. *Suppose that the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are clustered using the k-means algorithm with group representatives $\mathbf{z}_1, \ldots, \mathbf{z}_k$. If the entries of the vectors $\mathbf{x}_i$ are nonnegative and sum to one, then the entries of the representatives $\mathbf{z}_j$ are also nonnegative and sum to one.*

Proof.

$$\mathbf{z}_j = \frac{\sum\limits_{i \in G_j} \mathbf{x}_i}{|G_j|}$$

Since the entries of every $\mathbf{x}_i$ are nonnegative, $\sum\limits_{i \in G_j} \mathbf{x}_i$ is nonnegative. Also, $|G_j|$ is positive. Therefore, the the entries of the representatives $\mathbf{z}_j$ are also nonnegative.

The entries of $\mathbf{x}_i$ sum up to one. Then $\mathbf{1}^T \mathbf{x}_i = 1$.

$$\begin{aligned}
\mathbf{1}^T \mathbf{z}_j &= \mathbf{1}^T \frac{\sum\limits_{i \in G_j} \mathbf{x}_i}{|G_j|} \\
&= \frac{1}{|G_j|} \sum_{i \in G_j} \mathbf{1}^T \mathbf{x}_i \\
&= \frac{1}{|G_j|} \sum_{i \in G_j} 1 \\
&= \frac{1}{|G_j|} |G_j| \\
&= 1
\end{aligned}$$

Therefore, the entries in $\mathbf{z}_j$ also sum to one. □

Lemma B.2. *Suppose that the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are clustered using the k-means algorithm with group representatives $\mathbf{z}_1, \ldots, \mathbf{z}_k$. If the vectors $\mathbf{x}_i$ are Boolean, then the ith entry of the j group representative, which is $(\mathbf{z}_j)_i$, is 1, 0, or a fraction.*

Proof. If all vectors in a group G_j are $\mathbf{1}$, then the group representative $\mathbf{z}_j$ is $\mathbf{1}$.

$$\mathbf{z}_j = \frac{\sum\limits_{i \in G_j} \mathbf{x}_i}{|G_j|} = \frac{1}{|G_j|}(|G_j|\mathbf{1}) = \mathbf{1}$$

If all the vectors in a group G_j are $\mathbf{0}$, then the group representative $\mathbf{z}_j$ is $\mathbf{0}$.

$$\mathbf{z}_j = \frac{\sum\limits_{i \in G_j} \mathbf{x}_i}{|G_j|} = \frac{1}{|G_j|}\mathbf{0} = \mathbf{0}$$

If some vectors in G_j are $\mathbf{1}$ and others are $\mathbf{0}$, then $\mathbf{z}_j$ is a fraction.
Therefore, the ith entry of z_j is 0, 1, or a fraction. □

Compare the k-means Clustering and spectral Clustering. The k-means clustering and spectral clustering represent two approaches for clustering.

The k-means clustering focuses on the compactness of the graph. Therefore, points close to each other are grouped together and are compact around the group representatives.

The advantages of k-means clustering include the following:

- It is simple to implement.
- It is relatively computationally cheap as its time complexity is $O(kNn)$ where k is the number of groups, N is the number of vectors, and n is the size of the vectors.

The disadvantages of k-means clustering include the following:

- With different initial representatives, the algorithm may result in different final group assignments and group representatives.
- It cannot guarantee that it finds the partition that induces the minimal J^{clust}.
- Calculating the Euclidean norm in high dimensions can be difficult.

Spectral clustering focuses on the connectivity of the graph. Therefore, only connected points are grouped together. Even if two points are close to each other, as long as they are disconnected, they can be put into different clusters.

The advantages of spectral clustering include:

- It gives relatively good clustering results and can correctly cluster points that actually belong to the same cluster.
- It does not make any assumptions on the cluster shapes.

- It does not necessarily require the data set.

The disadvantages of spectral clustering include:

- It is very sensitive to the change of eigenvectors in the Laplacian matrix. So even if we shift the Laplacian matrix by a small constant, the result may not be perfect.
- It is computationally expensive. It is shown that its complexity is $O(n^3)$ in general, where n is the number of data points. For large and dense data sets, computing the eigenvalues and eigenvectors of a large matrix takes a lot of time.

Example B.3. Here is an example of how the k-means algorithm clusters 1500 2-vectors into 3 groups. The MATLAB code

```
rng("default")
k = 3;n =10000;
X = [randn(500,2)*0.75;randn(500,2)*0.5;randn(500,2)*0.65];

[idx,C] = mykmeans(X,k,n);

figure
gscatter(X(:,1),X(:,2),idx,"cgr");
hold on
plot(C(:,1),C(:,2),"kx")
legend("cluster 1","cluster 2","cluster 3","cluster centroid")
title("my k-means clustering")
hold off
```

printed the following

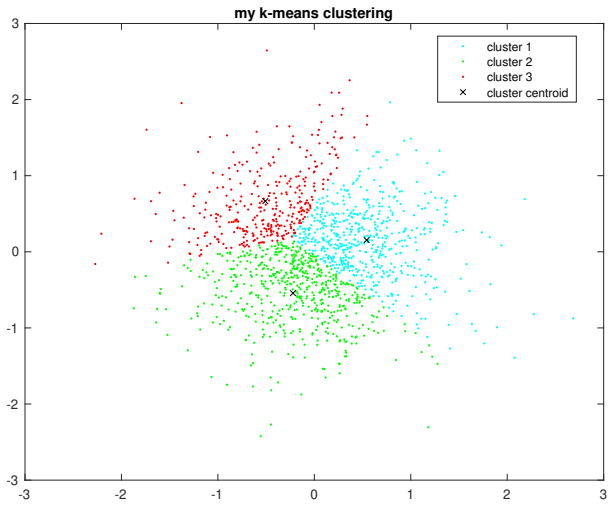

Figure 19: K-means clustering

Example B.4. Here is an example of two concentric circles. The k-means algorithm does not distinguish different circles and provides a bad solution . In comparison, spectral clustering gives a

good solution, which is consistent with the example given above.
The MATLAB code

```
N = 800;
r1 = 2; r2 = 4;
theta = linspace(0,2*pi,N)';

X1 = r1*[cos(theta),sin(theta)]+ rand(N,1);
X2 = r2*[cos(theta),sin(theta)]+ rand(N,1);
X = [X1;X2];

%run the k-means algorithm
k=2; n = 1000;
[idx,C] = mykmeans(X,k,n);

ax1 = subplot(2,1,1);
gscatter(X(:,1),X(:,2),idx,"cy");
hold on
plot(C(:,1),C(:,2),"kx")
legend("cluster 1","cluster 2", "cluster centroid")
title(ax1,"k-means clustering")
idx1 = spectralcluster(X,2);
%we use spectral clustering to cluster
ax2 = subplot(2,1,2);
gscatter(X(:,1),X(:,2),idx1,"cy");
legend("cluster 1","cluster 2")
title(ax2, "spectral clustering")
```

printed the following

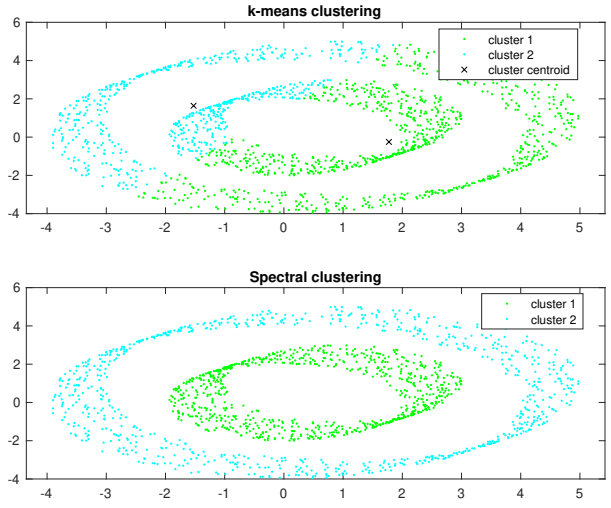

Figure 20: Comparison between k-means clustering and spectral clustering

C MATLAB Codes for Different Functions

C.1 Similarity Graph: Epsilon-Neighborhood Graph

The following codes present how we can use the similarity matrix to get the ϵ-neighborhood graph.

Firstly, we use the MATLAB function pdist(X) where X contains n data points to calculate the pairwise distance of the data points in set X so that we get a vector $D \in \mathbb{R}^{\frac{n(n-1)}{2}}$ such that

$$D = \begin{bmatrix} \|v_1 - v_2\| & \|v_1 - v_3\| & \dots & \|v_{n-1} - v_n\| \end{bmatrix}$$

Then we use the MATLAB function squareform(D) to transform D into a $n \times n$ symmetric matrix S. The function will first create a $n \times n$ zero matrix. In the first row of the upper triangular part of the matrix, we place the first $n-1$ entries of D in order. In the second row of the upper triangular part of the matrix, we place the next $n-2$ entries of D. We repeat the process until we put the last entry of D in the last row of the upper triangular part of the matrix. Then we create the lower triangular part of the matrix to build the corresponding symmetric matrix S using the upper triangular part of the matrix.

After that, we translate every entry of S into 1 or 0. If it is greater than the threshold $\epsilon > 0$, then it is 1. Otherwise, it is 0.

Here is the MATLAB code.

```
function S = similarity(X,epsilon)
D = pdist(X);
S = squareform(D);
S_row = size(S,1);
S_col = size(S,2);
for i = 1: S_row
    for j = 1:S_col
        if S(i,j) < epsilon && i ~=j
            S(i,j) = 1;
        else
            S(i,j)=0;
        end
    end
end
end
```

C.2 Unnormalized Spectral Clustering

```
function idx1 = my_unnormalized_sc(S,k)
D = diag(sum(S,2));
L = D - S;
[v,e] = eig(L);
U = v(:,1:k);
idx1 = kmeans(U,k);
end
```

C.3 Normalized Spectral Clustering using L_{rw}

```
function idx1 = my_normalized_sc_rw(S,k)
D = diag(sum(S,2));
L = D - S;
[v, e] = eig(L,D);
U = v(:,1:k);
idx1 = kmeans(U,k);
end
```

C.4 Normalized Spectral Clustering with L_{sym}

```
function idx1 = my_normalized_sc_sym(S,k)
n = size(S,1);
D = diag(sum(S,2));
D1 = diag(sum(S,2).^(-1/2));  %D^(-1/2)
L = D - S;
L_sym = D1 * L * D1;
[v,~] = eig(L_sym);
U = v(:,1:k);
T = zeros(n,k);
for i = 1: n
    for j = 1:k
        sum1 = sum((U(i,k)).^2);
        T(i,j) = U(i,j)/((sum1).^(1/2));
    end
end
idx1 = kmeans(T,k);
end
```

C.5 Normalized Spectral Clustering with Group Fairness Constraints

Appendix A of [7] provides the algorithm for normalized spectral clustering. In addition, [7] provides the corresponding MATLAB code in the file Fair_SC_normalized.m.

We build our MATLAB code for the fair version of normalized spectral clustering based on the algorithm and code [7] provides. The main change between our code and [7]'s is that instead of their input "sensitive", we require an input $F \in \mathbb{R}^{n\times(h-1)}$ defined by Equation (10.4).

```
function clusterLabels = Fair_SC_normalized(adj,k,F)
%adjis the adjacency matrix of size n x n
%k is number of clusters
%F = [f^(1) - |v_1| / |v| * 1_n ... f^(h-1) - |v_(h-1)| / |v| * 1_n]
%compute the Laplacian matrix L
D = diag(sum(adj,2));
L = D - adj;

%compute a matrix Z whose columns form an orthogonal basis of null(F^T)
Z = null(F');
```

```
%compute the principal square root Q of Z' D Z
Q=sqrtm(Z'*D*Z);

%compute the first k eigenvectors of Q^(-1)*Z^T*L*Z*Q^(-1)
Qinv=inv(Q);
Msymm=Qinv'*Z'*L*Z*Qinv;
Msymm=(Msymm+Msymm')/2;

% "smallestabs" and "smallestreal" both find the first k eigenvalues but with
%different methods related to the Krylov method
% "maxiterations" gives the maximum number of iterations
% "SubspaceDimension" gives the number of basis vectors

try
    [Y, eigValues] = eigs(Msymm,k,'smallestabs','MaxIterations', ...
        500,'SubspaceDimension',min(size(Msymm,1),max(2*k,25)));
catch
    [Y, eigValues] = eigs(Msymm,k,'smallestreal','MaxIterations',...
        1000,'SubspaceDimension',min(size(Msymm,1),max(2*k,25)));
end

%apply k-means clusteirng to the rows of H = Z Q^(-1) Y
%Y is a matrix containing these eigenvectors
H = Z*Qinv*Y;
clusterLabels = kmeans(H,k,'Replicates',10);
end
```

C.6 Normalized Spectral Clustering with Individual Fairness Constraints

```
function clusterLabels = Fair_SC_normalized_individual(adj,k,RJ)
% RJ = R * J
D = diag(sum(adj,2));
L = D - adj;

[U, S, V] = svd(RJ);
r = rank(RJ);
F = sqrt(S) * V'; F = F(1:r, :)';

Z = null(F');

Q=sqrtm(Z'*D*Z);

Qinv=inv(Q);
Msymm=Qinv'*Z'*L*Z*Qinv;
Msymm=(Msymm+Msymm')/2;

[Y, eigValues] = eigs(Msymm,k);
```

```
H = Z*Qinv*Y;
clusterLabels = kmeans(H,k,'Replicates',10);
end
```