1 Analysis of spectral clustering

Spectral clustering is actually a special case of optimal graph partition problem. Recall that we discuss two versions of spectral clusterings, based on either ratio cut or normalized cut. They are corresponding to graph Laplacian and normalized Laplacian respectively.

To warm up, we apply the spectral clustering to an undirected graph with \( k \) connected components. Suppose we have an undirected graph with weight \( w_{ij} \) and \( k \) connected components, denoted by \( \Gamma_1, \cdots, \Gamma_k \).

Based on the spectral clustering, we first construct its graph Laplacian. After rearranging data points, the graph Laplacian \( L \) can be written as a block-diagonal matrix,

\[
L = \begin{bmatrix}
L_1 & 0 & \cdots & 0 \\
0 & L_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & L_k
\end{bmatrix}
\]

where each \( L_i \in \mathbb{R}^{\mid \Gamma_i \mid \times \mid \Gamma_i \mid} \) is the graph Laplacian for each connected component.

Since \( L_i 1_{\mid \Gamma_i \mid} = 0 \) for all \( 1 \leq i \leq k \) where \( 1_{\mid \Gamma_i \mid} \) is a \( \mid \Gamma_i \mid \times 1 \) vector with all entries equal to 1. Therefore, the \( k \) smallest eigenvalues of \( L \) equal 0. Moreover, the null space of this matrix is spanned by \( k \) indicator functions such as

\[
\varphi_i = \frac{1}{\sqrt{\mid \Gamma_i \mid}} \begin{bmatrix} 1_{\mid \Gamma_i \mid} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \varphi_k = \frac{1}{\sqrt{\mid \Gamma_k \mid}} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1_{\mid \Gamma_k \mid} \end{bmatrix}
\]

where \( \langle \varphi_i, \varphi_j \rangle = 0 \) for \( i \neq j \) and \( \| \varphi_i \| = 1 \). This representation is unique up to an orthogonal transformation.
Hence the $\Phi$ is in the form of

$$
\Phi = \begin{pmatrix}
\frac{1}{\sqrt{|\Gamma_1|}}1_{|\Gamma_1|} & 0 & \cdots & 0 \\
0 & \frac{1}{\sqrt{|\Gamma_2|}}1_{|\Gamma_2|} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \frac{1}{\sqrt{|\Gamma_k|}}1_{|\Gamma_k|}
\end{pmatrix} \in \mathbb{R}^{n \times k}.
$$

Each row of $\Phi$ is a vector with only one nonzero entry and it is equal to $\frac{1}{\sqrt{|\Gamma_i|}}$. In other words, this spectral mapping transforms all points in the $i$th connected component $\Gamma_i$ into a single point $\frac{1}{\sqrt{|\Gamma_i|}}e_i$ where $\{e_i\}_{i=1}^k$ is the canonical basis in $\mathbb{R}^k$.

### 1.1 Graph cut point of view on spectral clustering

The intuition of clustering is to separate points in different groups according to their similarities. For data given in an undirected graph with weight $W$, we want to find a partition such that the connection between each group is small. Such a partition corresponds to a cut on the graph. Details can be found in [?, ?, ?, ?].

![Figure 1: Two concentric circles and its $\sigma$-nearest neighbor graphs](image)

In particular, as discussed before, for a disconnected graph with weight $W$, we can find partition such that the connection between two components is zero. Spectral clustering can be explained and derived from the notion of a cut on graph.

### 1.2 Graph cut, ratio cut, and normalized cut

**Cut:** First we define what a cut means here for two sets. Suppose we have two disjoint sets of vertices on the graph $\Gamma$ and $\Gamma^c$ s.t. $\Gamma \cap \Gamma^c = \emptyset$ and $\Gamma \cup \Gamma^c = \{1, \cdots, n\}$. We define a cut w.r.t. $\Gamma \subseteq V$ as

$$
cut(\Gamma, \Gamma^c) = \sum_{i \in \Gamma, j \in \Gamma^c} w_{ij} = \cut(\Gamma^c, \Gamma)
$$

is the total summation of the edge weights whose two vertices are in different sets. In particular, $\cut(V) = 0$. 

2
Figure 2: A partition of data set corresponds to a partition of network vertices.

Here, cut(Γ) measures how much Γ and Γ^c are associated/connected. A smaller value for cut(Γ) means fewer connections between Γ and Γ^c. But would minimizing cut(Γ) over all Γ ⊆ V give us a good partition of the graph? It does not always work.

Figure 3: An example in which minimizing the cut does not yield a satisfactory partition.

For example, the cut can return unwanted and meaningless results: declaring a single vertex as a cluster. Figure 3 shows that

\[ \Gamma = \{1, \cdots, 15\}, \quad \Gamma^c = \{16\} \implies \text{cut}(\Gamma, \Gamma^c) = 1, \]
\[ \Gamma = \{1, \cdots, 8\}, \quad \Gamma^c = \{9, \cdots, 16\} \implies \text{cut}(\Gamma, \Gamma^c) = 2. \]
In other words, minimizing $\text{cut}(\Gamma)$ is likely to give us unbalanced clusters/partitions.

**Definition 1.1.** Define the ratio cut and normalized cut:

\[
\begin{align*}
\text{Rcut}(\Gamma) &= \frac{\text{cut}(\Gamma, \Gamma^c)}{|\Gamma|} + \frac{\text{cut}(\Gamma, \Gamma^c)}{|\Gamma^c|}, \\
\text{Ncut}(\Gamma) &= \frac{\text{cut}(\Gamma, \Gamma^c)}{\text{Vol}(\Gamma)} + \frac{\text{cut}(\Gamma, \Gamma^c)}{\text{Vol}(\Gamma^c)}
\end{align*}
\]

where $\text{Vol}(\Gamma)$ is the volume of $\Gamma$ and is defined as $\text{Vol}(\Gamma) = \sum_{i \in \Gamma} d_i = \sum_{i \in \Gamma} \sum_{j=1}^n w_{ij}$.

Now let us consider the strength and weakness of $\text{Rcut}(\Gamma)$ and $\text{Ncut}(\Gamma)$:

1. Pros: given the same value of $\text{cut}(\Gamma)$, we will have smaller $\text{Rcut}(\Gamma)$ and $\text{Ncut}(\Gamma)$ if $\Gamma$ and $\Gamma^c$ are more or less of the same size. Due to this property, we can avoid unbalanced clusters by using $\text{Rcut}(\Gamma)$ and $\text{Ncut}(\Gamma)$ as new criteria.

2. Cons: both two criteria are difficult to minimize, i.e.,

\[
\begin{align*}
\min_{\Gamma \subseteq V} \text{Rcut}(\Gamma) \\
\min_{\Gamma \subseteq V} \text{Ncut}(\Gamma)
\end{align*}
\]

are NP-hard in general since they are discrete/combinatorial optimization. Using brute force requires us to search over $2^{|V|}$ (the total number of subsets in $V$) choices.

### 1.3 Relaxation of the ratio cut

We find it difficult to minimize this function over all subsets of $\Gamma$. However, it does not mean that we cannot approximate (1.1) and (1.2) by other optimization programs which are easier to solve. We hope the solutions to the alternative programs are able to give us good approximations of the original ones.

In this section, we will show that the ratio cut (1.1) can be approximated by a continuous optimization program (eigenvalue/vector problem of $L$) which exactly matches the first step of spectral clustering. We start with relating the ratio cut to the quadratic form of $L$.

First note that the cut is directly related to quadratic form associated to Laplacian and normalized Laplacian.

**Lemma 1.1.** Let $1_{\Gamma} \in \mathbb{R}^n$ and $1_{\Gamma^c} \in \mathbb{R}^n$ be the indicator vectors of $\Gamma$ and $\Gamma^c$ respectively. Then

\[
\text{cut}(\Gamma, \Gamma^c) = 1_\Gamma^T L 1_\Gamma = 1_{\Gamma^c}^T L 1_{\Gamma^c}.
\]

**Proof:** Note that the quadratic form of $L$ satisfies: $z^T L z = \frac{1}{2} \sum_{i,j} w_{ij} (z_i - z_j)^2$ for all
\[ z \in \mathbb{R}^n \text{ and thus} \]

\[
1_{\Gamma}^\top L 1_{\Gamma} = \frac{1}{2} \sum_{i,j} w_{i,j} (1_{\Gamma}(i) - 1_{\Gamma}(j))^2
\]

\[
= \frac{1}{2} \left( \sum_{i \in \Gamma, j \in \Gamma^c} + 2 \sum_{i \in \Gamma, j \in \Gamma^c} + \sum_{i \in \Gamma^c, j \in \Gamma^c} \right) w_{i,j} (1_{\Gamma}(i) - 1_{\Gamma}(j))^2
\]

\[
= \sum_{i \in \Gamma, j \in \Gamma^c} w_{i,j} (1_{\Gamma}(i) - 1_{\Gamma}(j))^2
\]

\[
= \sum_{i \in \Gamma, j \in \Gamma^c} w_{i,j} = \text{cut}(\Gamma, \Gamma^c) = 1_{\Gamma^c}^\top L 1_{\Gamma^c}
\]

where \(|1_{\Gamma}(i) - 1_{\Gamma}(j)| = 1\) for \(i \in \Gamma, j \in \Gamma_j\).

Note that

\[ z^\top L z = \langle L, zz^\top \rangle = \text{Tr}(Lzz^\top) \]

As a result, the ratio cut can be written into

\[
\text{Rcut}(\Gamma, \Gamma^c) = \frac{\text{cut}(\Gamma, \Gamma^c)}{|\Gamma|} + \frac{\text{cut}(\Gamma, \Gamma^c)}{|\Gamma^c|}
\]

\[
= \frac{1_{\Gamma}^\top L 1_{\Gamma}}{|\Gamma|} + \frac{1_{\Gamma^c}^\top L 1_{\Gamma^c}}{|\Gamma^c|}
\]

\[
= \left\langle L, \underbrace{1_{\Gamma} 1_{\Gamma}^\top + \frac{1}{|\Gamma^c|} 1_{\Gamma^c} 1_{\Gamma^c}^\top}_{Z} \right\rangle = \langle L, Z \rangle
\]

where

\[ Z := \frac{1}{|\Gamma|} 1_{\Gamma} 1_{\Gamma}^\top + \frac{1}{|\Gamma^c|} 1_{\Gamma^c} 1_{\Gamma^c}^\top. \quad (1.5) \]

In other words, minimizing the ratio cut is equivalent to minimizing \(\langle L, Z \rangle\) over all \(Z\) in the form of (1.5). This is still an NP-hard problem. Thus we want to relax this hard problem and see how it works.

**Relaxation:** What is relaxation? The idea of relaxation is to enlarge the search space which yields a computationally feasible problem (solvable with an algorithm of polynomial time complexity).

We can see that \(Z\) is positive semidefinite and a projection matrix, i.e., \(Z = UU^\top\) where \(U \in \mathbb{R}^{n \times 2}\)

\[
U = \begin{bmatrix}
\frac{1}{\sqrt{|\Gamma|}} 1_{\Gamma}, & \frac{1}{\sqrt{|\Gamma^c|}} 1_{\Gamma^c}
\end{bmatrix}, \quad U^\top U = I_2.
\]

Therefore, one way to relax the ratio cut is to replace \(Z\) by a symmetric PSD projection matrix:

\[
\min \langle L, VV^\top \rangle \quad \text{s.t.} \quad V^\top V = I_2.
\]

The solution is easy to obtain by computing the eigenspace w.r.t. the smallest two eigenvalue. Remember that the smallest eigenvalue is also \(1_n\). Therefore, we only need to compute the second smallest eigenvector \(\varphi_2\) and \(V = [1_n, \varphi_2]\). Note that the second smallest eigenvector is also the minimizer to the following optimization program (recall the min-max characterization of eigenvalues/eigenvectors of symmetric matrices in PCA):

\[
\min \langle v^\top L v \rangle \quad \text{s.t.} \quad \|v\| = 1, \quad v^\top 1_n = 0.
\]
In other words, only the second smallest eigenvector matters for two clusters. Also this spectral relaxation is exactly the first step in spectral clustering.

However, \( \varphi_2 \) is not necessarily integral. Rounding is needed: using k-means for example, which is equivalent to find \( r \) such that

\[
\Gamma := \{ i : \varphi_2(i) > r \}, \quad \Gamma^c := \{ i : \varphi_2(i) \leq r \}
\]

From the discussion above, we can above that

\[
\lambda_2(L) \leq \text{OPT-Rcut}.
\]

### 1.4 Relaxation of the normalized cut

Normalized cut has a similar result:

\[
\text{Ncut}(\Gamma, \Gamma^c) = \frac{\text{cut}(\Gamma, \Gamma^c)}{\text{Vol}(\Gamma)} + \frac{\text{cut}(\Gamma, \Gamma^c)}{\text{Vol}(\Gamma^c)} = \frac{1^{\top}_\Gamma L 1_\Gamma}{1^{\top}_\Gamma D 1_\Gamma} + \frac{1^{\top}_{\Gamma^c} L 1_{\Gamma^c}}{1^{\top}_{\Gamma^c} D 1_{\Gamma^c}}
\]

where

\[
\text{Vol}(\Gamma) = 1^{\top}_\Gamma D 1_\Gamma, \quad \text{Vol}(\Gamma^c) = 1^{\top}_{\Gamma^c} D 1_{\Gamma^c}.
\]

Now it holds that

\[
\text{Ncut}(\Gamma, \Gamma^c) = \frac{1^{\top}_\Gamma L 1_\Gamma}{1^{\top}_\Gamma D 1_\Gamma} + \frac{1^{\top}_{\Gamma^c} L 1_{\Gamma^c}}{1^{\top}_{\Gamma^c} D 1_{\Gamma^c}}
\]

\[
= \left( D^{-1/2} LD^{-1/2}, \frac{D^{1/2} 1_\Gamma 1^{\top}_\Gamma D^{1/2}}{1^{\top}_\Gamma D 1_\Gamma} \right) + \left( D^{-1/2} LD^{-1/2}, \frac{D^{1/2} 1_{\Gamma^c} 1^{\top}_{\Gamma^c} D^{1/2}}{1^{\top}_{\Gamma^c} D 1_{\Gamma^c}} \right)
\]

\[
= \langle \mathcal{L}, Z \rangle
\]

where

\[
Z = \frac{D^{1/2} 1_\Gamma 1^{\top}_\Gamma D^{1/2}}{1^{\top}_\Gamma D 1_\Gamma} + \frac{D^{1/2} 1_{\Gamma^c} 1^{\top}_{\Gamma^c} D^{1/2}}{1^{\top}_{\Gamma^c} D 1_{\Gamma^c}}
\]

In fact, \( Z \) is still a symmetric PSD projection matrix. The spectral relaxation is

\[
\text{min} \ \langle \mathcal{L}, VV^\top \rangle \quad \text{s.t.} \quad V^\top V = I_2.
\]

Note that the smallest eigenvalue and eigenvector is 0 and \( D^{1/2} 1_n \). Thus only the second smallest eigenvector matters, which is given by

\[
\text{min} \ \langle \mathcal{L}, vv^\top \rangle \quad \text{s.t.} \quad ||v|| = 1, v^\top D^{1/2} 1_n = 0.
\]

The rounding procedure is exactly the same: applying kmeans to \( \varphi_2 \). We also have

\[
\lambda_2(\mathcal{L}) \leq \text{OPT-Ncut}.
\]

### 1.5 Cheeger’s inequality

The conductance of \( \Gamma \) is given by

\[
\phi(\Gamma) = \frac{\text{cut}(\Gamma, \Gamma^c)}{\min\{\text{Vol}(\Gamma), \text{Vol}(\Gamma^c)\}}
\]

and the conductance of a graph is

\[
\phi_G := \min_{\Gamma \subseteq V} \phi(\Gamma).
\]

Remember that the graph conductance is closely related to optimal normalized cut:

\[
\frac{1}{2} \text{OPT-Ncut} \leq \phi_G \leq \text{OPT-Ncut}.
\]
**Theorem 1.2** (Cheeger’s inequality). It holds that

\[ \frac{\lambda_2(L)}{2} \leq \phi_G \leq \sqrt{2\lambda_2(L)}. \]

Note that the conductance is bounded by the Fiedler eigenvalue of normalized Laplacian. The proof can be found in [?, ?]. In fact, the lower bound of \( \phi_G \) is easy to obtain, simply following from spectral relaxation.

In other words,

\[ \text{OPT-Ncut} \leq 2\sqrt{2\lambda_2(L)}. \]

Is this bound tight? Yes, the bound is tight. The examples are

(a) Complete graph: the first inequality \( \frac{\lambda_2(L)}{2} \leq \phi_G \) is tight up to a constant.

(b) \( n \)-cycle graph: \( \phi_G \leq \sqrt{2\lambda_2(L)} \) is tight.

**References**


