Spectral Clustering and Community Detection

Arnab Sen

June 2023

Course Outline

Part I.

- k-mean clustering
- Spectral clustering
- Worst case analysis ($k = 2$): Cheeger's inequality

Part II.

- Random case analysis: Stochastic Block Model ($k = 2$)
- Dense case: analysis via spectral perturbation
- Sparse case: phase transition, spectral redemption.

Part I

k-mean clustering

Data: $x_1, x_2, \dots, x_n$ are points in $\mathbb{R}^d$.

Goal: Partition the data points into k disjoint groups $\mathcal{S} = \{S_1, S_2, \dots, S_k\}$ with centers $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_k)$ such that the following energy is minimized

$$H(\boldsymbol{\mu}, \mathcal{S}) = \sum_{j=1}^{k} \sum_{i \in S_j} \|x_i - \mu_j\|_2^2. \tag{1}$$

Here k = the number of clusters, which is assumed to be known. The above minimization is done with respect to all partitions $\mathcal{S}$ of $[n]$ and all centers $\mu_1, \dots, \mu_k \in \mathbb{R}^d$.

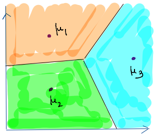
Minimization of k-mean energy

- The minimization of the k-mean-energy is NP-hard.
- Given the partition $\mathcal{S}$, the optimal centers are given by the respective means of each partition blocks

$$\mu_j = \frac{1}{|S_j|} \sum_{i \in S_j} x_i.$$

- Given the centers $(\mu_j)_j$, the optimal partition is given by Voronoi partition of $\mathbb{R}^d$, that is,

$$i \in S_j \Leftrightarrow \|x_i - \mu_j\|_2 \leq \|x_i - \mu_\ell\|_2 \text{ for all } \ell = 1, \dots, k.$$



Lloyd's algorithm

Initialize centers $\mu_1^0, \mu_2^0, \ldots, \mu_k^0$.

It is an iterative algorithm that alternates between

Step I. (assignment step) Given the centers $\mu_1^t, \ldots, \mu_k^t$ and the cluster assignments $(a_1^t, \ldots, a_n^t) \in \{1, 2, \ldots, k\}^n$, update the partition from $\mathcal{S}^t$ to $\mathcal{S}^{t+1}$ as follows.

For $i = 1, 2, \ldots, n$

$$\text{if } \operatorname{argmin}_{1 \le l \le k} \|x_i - \mu_l^t\|_2 < \|x_i - \mu_{a_i^t}^t\|_2,$$

then assign x_i to the new cluster $a_i^{t+1} := \operatorname{argmin}_{1 \le l \le k} \|x_i - \mu_l^t\|_2$. Otherwise (in case of equality), keep the previous assignment $a_i^{t+1} = a_i^t$.

Step II. (refitting step) Update the centers.

$$\mu_j^{t+1} = \frac{1}{|S_j^{t+1}|} \sum_{i \in S_j^{t+1}} x_i$$

The algorithm terminates during step I if we have $\mathcal{S}^{t+1} = \mathcal{S}^t$ for all j.

Convergence of Lloyd's algorithm

Recall

$$H(\boldsymbol{\mu}, \mathcal{S}) = \sum_{j=1}^{k} \sum_{i \in S_j} \|x_i - \mu_j\|_2^2.$$

Lemma

The Lloyd's algorithm decreases the energy in each step, i.e.,

- *Step I:*

$$H(\boldsymbol{\mu}, \mathcal{S}^{t+1}) \leq H(\boldsymbol{\mu}, \mathcal{S}^{t}) \text{ for any } \boldsymbol{\mu}.$$

  Furthermore, the equality holds if and only if $\mathcal{S}^{t+1} = \mathcal{S}^{t}$.

- *Step II:*

$$H(\boldsymbol{\mu}^{t+1}, \mathcal{S}) \leq H(\boldsymbol{\mu}^{t}, \mathcal{S}) \text{ for any } \mathcal{S}.$$

Hence, the algorithm converges in a finite number of iterations.

- Lloyd's algorithm may get stuck in local optima and is not guaranteed to converge to the global optima.

Deficiencies of Lloyd's algorithm

- The final partition of Lloyd's algorithm heavily depends on the initial choice of the centers.

- Lloyd's algorithm always yields partition with convex regions. Hence, it often struggles with non-convex clusters.

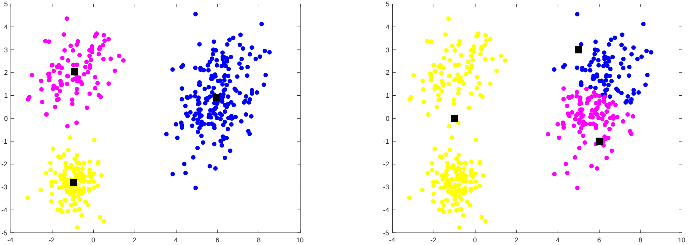
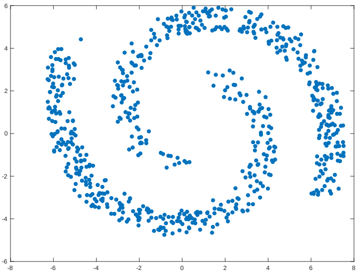
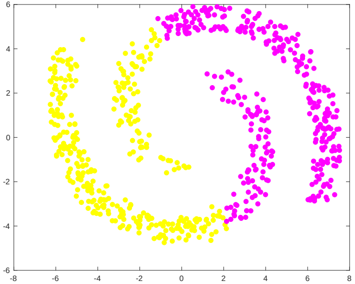


Figure: The k-mean clustering is sensitive to initial choice of centers.

k-mean clusters for double-moon shaped data





clustering on graph

- Sometimes the data points do not live naturally in a metric space, but it comes with a graphical (similarity) structure. For example, Facebook network.

- Given the data $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, we can naturally represent the data using a graph (undirected, possibly weighted) - examples in the next slide - such that "close" points are connected by an edge.

- Partition the graph into disjoint components such that

  - a lot of connections (large weights) within a components

  - very few connections (small weights) across different components.

How to construct such graphs?

Construct a weighted adjacency matrix $W \in \mathbb{R}^{n\times n}$, which is symmetric, where

$$0 \leq W(i,j) = \text{encodes the how similar or close are the points } x_i \text{ and } x_j.$$

$W(i,j) > 0$ iff $i \sim j$.

- ε-neighborhood graph: Connect all points whose pairwise distances are smaller than ε.

  Take W to be the adjacency matrix of the graph, i.e., $W(i,j) = 1$ if $x_i \sim x_j$ and $W(i,j) = 0$ otherwise.

- k-nearest neighbor graph: Connect vertex x_i with vertex x_j if x_i is among the k-nearest neighbors of x_j and vice-versa.

- Gaussian weights. Complete graph with weights

$$W(i,j) = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\right).$$

Graph Laplacian

Let $G = (V, E)$ be a graph with a weighted adjacency matrix W. Let $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$ be the diagonal matrix of (weighted) vertex degrees,

$$d_i = \sum_{j:j\sim i} w_{ij}.$$

The Laplacian matrix of G is defined as

$$L = D - W.$$

There are two versions of the normalized graph Laplacians

$$\mathcal{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2},$$
$$\mathcal{Q} = D^{-1} L = I - D^{-1} W.$$

Spectral properties of normalized Laplacian $\mathcal{L}$

Lemma

- *$\mathcal{L}$ is symmetric and positive semi-definite. The eigenvalues are non-negative reals*
$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \leq 2.$$
- *$\lambda_k = 0$ iff G has at least k connected components. ($\Rightarrow \lambda_2 > 0$ iff G is connected).*
- *$\lambda_n = 2$ iff G has a bipartite connected component.*

Courant-Fischer characterization of eigenvalues

Let $M \in \mathbb{R}^{n \times n}$ be a symmetric matrix with eigenvalues

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

Then

$$\lambda_1 = \min_{x \in \mathbb{R}^n} \frac{x^T M x}{x^T x},$$
$$\lambda_n = \max_{x \in \mathbb{R}^n} \frac{x^T M x}{x^T x}.$$

More generally,

$$\lambda_k = \min_{U : k-\text{dim subspace of } \mathbb{R}^n} \max_{x \in U} \frac{x^T M x}{x^T x}.$$

Moreover, the optimal U^* is spanned by the first k eigenvectors of M.

Eigenvalues and connected components (unweighted case)

For simplicity, assume that the graph is unweighted. Then

$$x^T L x = \sum_{i\sim j} (x_i - x_j)^2 \geq 0.$$

$$\begin{aligned}
\lambda_k(\mathcal{L}) &= \min_{\tilde{U}:k-\textsf{dim subspace}} \max_{x\in\tilde{U}} \frac{x^T \mathcal{L} x}{x^T x} \\
&= \min_{U:k-\textsf{dim subspace}} \max_{x\in U} \frac{(D^{1/2}x)^T \mathcal{L} (D^{1/2}x)}{(D^{1/2}x)^T (D^{1/2}x)} \\
&= \min_{U:k-\textsf{dim subspace}} \max_{x\in U} \frac{x^T L x}{x^T D x} \\
&= \min_{U:k-\textsf{dim subspace}} \max_{x\in U} \frac{\sum_{i\sim j} (x_i - x_j)^2}{\sum_i d_i x_i^2}
\end{aligned}$$

Optimal $\quad U^* = D^{-1/2}\tilde{U}^* = D^{-1/2} \cdot \mathrm{span}(\text{the first } k \text{ eigenvectors of } \mathcal{L}).$

$\lambda_k = 0 \Leftrightarrow G$ *has at least* k *connected components*

- Suppose $\lambda_k = 0$. There exists a k-dimensional subspace U^* such that for all $x \in U^*$
$$\sum_{i \sim j} (x_i - x_j)^2 = 0.$$
This implies that if $x \in U^*$, then x must be constant on connected components of G. Thus
$$k = \dim(U^*) \leq \#\text{connected components}.$$

- If G has k connected components $S_1, S_2, \ldots S_k$, then take
$$U^* = \operatorname{span}(\mathbb{1}_{S_1}, \ldots, \mathbb{1}_{S_k}).$$

Suppose G has k connected components ($\Rightarrow \lambda_k = 0$). Fix an optimal $U^* \in \mathbb{R}^{n\times k}$ of dimension k.

(i) The rows of U^* are constant vectors in $\mathbb{R}^k$ over each connected components.

(ii) At least k rows of U^* are different.

We can take

$$U^* = D^{-1/2}[f^{(1)} : f^{(2)} : \cdots : f^{(k)}]$$

where $f^{(1)}, f^{(2)}, \ldots, f^{(k)}$ are first k eigenvectors of $\mathcal{L}$. Then the map

$$F(i) = d_i^{-1/2}(f_i^{(1)}, f_i^{(2)}, \ldots, f_i^{(k)}) : [n] \to \mathbb{R}^k.$$

has the above two properties.

$\mathcal{L}$ = normalized laplacian of a (possibly weighted graph) G,
$f^{(1)}, f^{(2)}, \dots, f^{(k)}$ are first k eigenvectors.

Spectral embedding into $\mathbb{R}^k$.

$$V = \{1, 2, \dots, n\} \to (F(1), F(2), \dots, F(n)),$$

where

$$F(i) = d_i^{-1/2}(f_i^{(1)}, f_i^{(2)}, \dots, f_i^{(k)}).$$

Intuition: If G has k 'approximately' connected components, F maps the vertices into k distinct closely packed group of points in $\mathbb{R}^k$.

For a connected graph $f^{(1)} = D^{1/2}1$. So, $D^{-1/2}f^{(1)} = 1$. So, we can ignore the first component of F and just define the spectral embedding as

$$\tilde{F}(i) = d_i^{-1/2}(f_i^{(2)}, \dots, f_i^{(k)}) \in \mathbb{R}^{k-1}.$$

Spectral embedding on random graph

Let $G = (V, E)$ be a random graph such that $V = S_1 \cup S_2 \cup S_3 \cup S_4$.

Independently for each pair i, j

$$\mathbb{P}((ij) \in E) = \begin{cases} 0.2 & \text{if } i, j \in S_k \\ 0.02 & \text{if } i \in S_k, j \in S_l \end{cases}$$

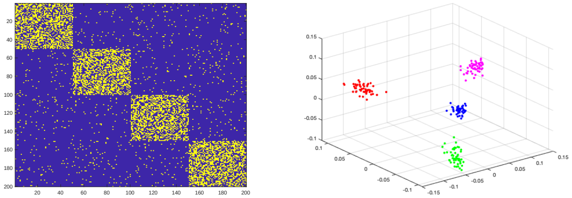


Figure: (left) adjacency matrix of G, (right) spectral embedding.

It would be now easy for the k-mean algorithm to identify these groups of points.

Spectral clustering algorithm

Ng-Jordan-Weiss '02, Shi-malik '00:

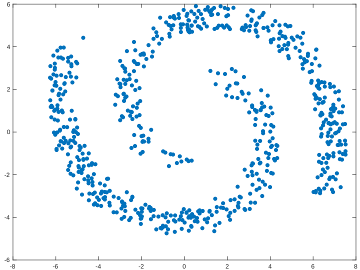
Step I. Construct a similarity graph G (unweighted or weighted). Compute its normalized Laplacian matrix $\mathcal{L} = I - D^{-1/2} W D^{-1/2}$.

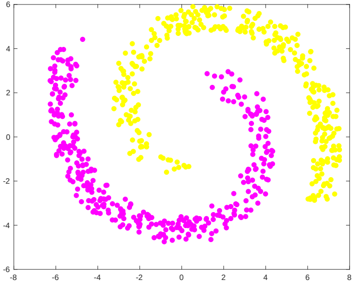
Step II. Map the points to $\mathbb{R}^k$ using the spectral embedding

$$F(i) = d_i^{-1/2} (f_i^{(1)}, f_i^{(2)}, \ldots, f_i^{(k)}).$$

Step 3. (rounding step) Apply k-means to $F(1), F(2), \ldots, F(n)$ into k clusters.

spectral clusters for double-moon data using Gaussian weights





Another example

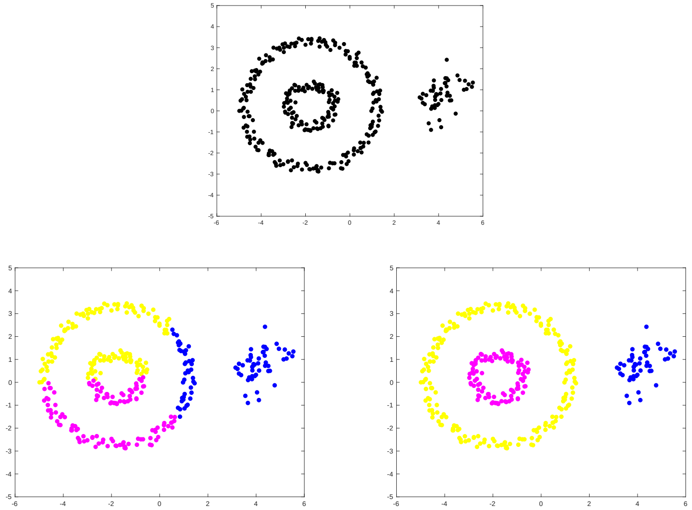


Figure: (top)data, (bottom left) k-mean clustering, (bottom right) spectral clustering

Next

- Rigorous worst-case analysis?

- For simplicity, we will consider $k = 2$. In this case, the spectral embedding says the second eigenvector $f^{(2)}$ contains good information about the optimal partition of G into two "clusters". Why?

Graph conductance, optimal cut

$G = (V, E, W)$ be a weighted graph.

For $S \subset V$, the total weight of the edges cut by S

$$\omega(S, S^c) = \sum_{i \in S, j \in S^c} W(u, v).$$

Normalized cut of S:

$$\phi(S) = \frac{\omega(S, S^c)}{\min(\mathrm{vol}(S), \mathrm{vol}(S^c))},$$

where $\mathrm{vol}(S) = \sum_{i \in S} d_i$.

Conductance of G.

$$\Phi_G = \min_{S : \mathrm{vol}(S) \le \mathrm{vol}(V)/2} \phi(S) \in [0, 1].$$

If $S^* = \mathrm{argmin} \phi(S)$, then $(S^*, (S^*)^c)$ gives us the optimal cut.

- Solving combinatorial optimization problem Φ_G is NP-hard (search over binary vectors).
- (Roughly) If we relax the search space from $\{0,1\}^n$ to $\mathbb{R}^n$, we get $\lambda_2(\mathcal{L})$.
- $$\lambda_2 = 0 \Leftrightarrow G \text{ is disconnected } \Leftrightarrow \Phi_G = 0.$$

- Solving combinatorial optimization problem Φ_G is NP-hard (search over binary vectors).
- (Roughly) If we relax the search space from $\{0,1\}^n$ to $\mathbb{R}^n$, we get $\lambda_2(\mathcal{L})$.
- $$\lambda_2 = 0 \Leftrightarrow G \text{ is disconnected } \Leftrightarrow \Phi_G = 0.$$

Theorem (Cheeger's inequality)

For any finite graph G,

$$\frac{\lambda_2}{2} \le \Phi_G \le \sqrt{2\lambda_2},$$

where λ_2 is the second smallest eigenvalue of $\mathcal{L}$.

- The lower bound is easy. The upper bound is hard.
- The Cheeger's inequality also holds for weighted graph.

- The proof of the upper bound $\Phi_G \leq \sqrt{2\lambda_2}$ is constructive.
- The proof gives a good cut S^o, obtained from suitably rounding $f^{(2)}$, such that

$$\phi(S) \leq \sqrt{2\lambda_2} \leq 2\sqrt{\Phi_G}.$$

Relaxation step: proof of $\lambda_2/2 \leq \Phi_G$

For simplicity, we will take G to be an unweighted d-regular graph. In this case, $\mathcal{L} = I - \frac{1}{d}A$.

$$
\begin{aligned}
\lambda_2 &= \min_{x \perp 1, x \neq 0} \frac{x^T L x}{d x^T x} = \min_{x \perp 1, x \neq 0} \frac{\sum_{i \sim j}(x_i - x_j)^2}{d \sum_i x_i^2} \\
&= \min_{x \perp 1, x \neq 0} \frac{\sum_{i \sim j}(x_i - x_j)^2}{\frac{d}{2n}\sum_{i,j}(x_i - x_j)^2} \\
&= \min_{x \text{ non-constant}} \frac{\sum_{i \sim j}(x_i - x_j)^2}{\frac{d}{2n}\sum_{i,j}(x_i - x_j)^2}
\end{aligned}
$$

Suppose we perform the above minimization over binary vectors. Take $x = \mathbb{1}_S$. Then the optimal value becomes

$$
\begin{aligned}
&\min_S \frac{\sum_{i\sim j}(\mathbb{1}_S(i) - \mathbb{1}_S(j))^2}{\frac{d}{2n}\sum_{i,j}(\mathbb{1}_S(i) - \mathbb{1}_S(j))^2} \\
&= \min_S \frac{\omega(S, S^c)}{\frac{d}{n}|S||S^c|} \\
&= \min_S \frac{\omega(S, S^c)}{\frac{\mathrm{vol}(S)\mathrm{vol}(S^c)}{\mathrm{vol}(V)}} \qquad [\mathrm{vol}(S) = d|S|] \\
&\leq 2\Phi_G.
\end{aligned}
$$

Cheeger's inequality: hard direction

- Need to show $\Phi_G \le \sqrt{2\lambda_2}$.

Fiedler's sweep algorithm

(a) Compute the eigenvector $f^{(2)}$ of $\mathcal{L}$ corresponding to λ_2.

(b) Order the vertices so that

$$\frac{f_1^{(2)}}{\sqrt{d_1}} \le \frac{f_2^{(2)}}{\sqrt{d_2}} \le \cdots \le \frac{f_n^{(2)}}{\sqrt{d_n}}.$$

(c) Choose "sweep" cut $(S^o, (S^o)^c) = (\{1, 2, \ldots, i\}, \{i+1, \ldots, n\})$ with smallest conductance.

- Will show that $\phi(S^o) \le \sqrt{2\lambda_2} \Rightarrow \Phi_G \le \sqrt{2\lambda_2}$.

Proof of Cheeger's inequality: hard direction

Assume that G is d-regular.

Let x be the second eigenvector of $\mathcal{L}$. Set $y = x^+$. WLOG, $\mathrm{supp}(y) \le n/2$.

Define the Rayleigh quotient of z as

$$R(z) = \frac{z^T \mathcal{L} z}{z^T z} = \frac{\sum_{i \sim j} (z_i - z_j)^2}{d \sum_i z_i}.$$

Claim. $R(y) \le R(x) = \lambda_2$.

Proof. Take i such that $y_i = x_i > 0$. Then

$$(\mathcal{L}y)_i = y_i - \frac{1}{d} \sum_{j : j \sim i} y_j \le x_i - \frac{1}{d} \sum_{j : j \sim i} x_j = (\mathcal{L}x)_i = \lambda_2 x_i = \lambda_2 y_i.$$

So,

$$y^T \mathcal{L} y = \sum_i y_i (\mathcal{L}y)_i = \sum_{i : y_i > 0} y_i (\mathcal{L}y)_i \le \sum_{i : y_i > 0} \lambda_2 y_i^2 = \lambda_2 \sum_i y_i^2.$$

Key Lemma

Lemma

Let $y \geq 0$. *For* $t \in (0, \max_i y_i]$, *set*

$$S_t = \{i : y_i^2 \geq t\}.$$

There exists t *such that*

$$\phi(S_t) \leq \sqrt{2R(y)}.$$

Plugging in $y = x_+$, we obtain $S_t \subseteq \operatorname{supp}(y) \leq n/2$ and

$$\phi(S_t) \leq \sqrt{2R(y)} \leq \sqrt{2R(x)} = \sqrt{2\lambda_2},$$

which shows that

$$\Phi_G \leq \sqrt{2\lambda_2}.$$

Moreover, every such S_t is examined by Fiedler's algorithm.

Proof of the key lemma

Proof in one line - "just pick a random threshold"!

By appropriate scaling, assume that $0 \le y_i \le 1$ for all i. Let $t \sim U(0,1)$.

Recall $S_t = \{i : y_i^2 \ge t\}$.

$$
\begin{aligned}
\mathbb{E}[\omega(S_t)] &= \sum_{i\sim j} \mathbb{P}(\text{the edge } (ij) \text{ is cut by } S_t) \\
&= \sum_{i\sim j} \mathbb{P}(y_i^2 < t \le y_j^2 \text{ or } y_j^2 < t \le y_i^2) \\
&= \sum_{i\sim j} |y_i^2 - y_j^2| = \sum_{i\sim j} |y_i - y_j||y_i + y_j| \\
&\le \sqrt{\sum_{i\sim j}(y_i - y_j)^2}\sqrt{\sum_{i\sim j}(y_i + y_j)^2} && \text{[Cauchy Schwarz]} \\
&\le \sqrt{\sum_{i\sim j}(y_i - y_j)^2}\sqrt{2d\sum_i y_i^2} && [(a+b)^2 \le 2(a^2+b^2).] \\
&= \sqrt{2R(y)} \cdot d \sum_i y_i^2 .
\end{aligned}
$$

On the other hand,

$$\mathbb{E}\mathrm{vol}(S_t) = d\mathbb{E}|S_t| = d\sum_i \mathbb{P}(y_i \geq t) = d\sum_i y_i^2.$$

Therefore,

$$\frac{\mathbb{E}[\omega(S_t)]}{\mathbb{E}\mathrm{vol}(S_t)} \leq \sqrt{2R(y)},$$

which implies

$$\mathbb{E}\Big[\omega(S_t) - \sqrt{2R(y)}\mathrm{vol}(S_t)\Big] \leq 0.$$

We conclude that there exists a deterministic $t \in [0, 1]$ such that

$$\omega(S_t) - \sqrt{2R(y)}\mathrm{vol}(S_t) \leq 0.$$

Consequently,

$$\phi(S^o) \leq \phi(S_t) \leq \sqrt{2R(y)} \leq \sqrt{2R(x)} = \sqrt{2\lambda_2}.$$

Discussion on Cheeger: approximate eigenvector

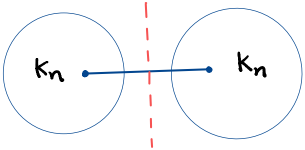
The proof can be modified to show that if we perform sweep algorithm on any vector $x \perp 1$, we obtain

$$\phi(S^o) \leq \sqrt{2R(x)}.$$

Therefore, the sweep algorithm produces good cut if we feed an approximate second eigenvector (can be computed more efficiently using power method).

Tightness of Cheeger's inequality $\Phi_G = \Omega(\lambda_2)$

The dumbbell graph.



In this case,

$$\Phi_G = \Theta(n^{-2}).$$

By Cheeger, $\lambda_2 = O(n^{-2})$. It can be shown that (exercise) $\lambda_2 = \Theta(n^{-2})$.

The second eigenvector $f^{(2)}$ will approximately $+1$ on the left part and -1 on the right part. Hence, the sweep algorithm gives the best cut.

Tightness of Cheeger's inequality $\Phi_G = O(\sqrt{\lambda_2})$

- Cycle C_n. Let $x = (1, 1 - \frac{4}{n}, 1 - \frac{8}{n}, \dots, -1, -1 + \frac{4}{n}, \dots, 1) \perp 1$. Then

$$\lambda_2 \leq \frac{x^T L x}{2 x^T x} = \frac{\sum_{i \sim j} (x_i - x_j)^2}{2 \sum_i x_i^2} = O(n^{-2}).$$

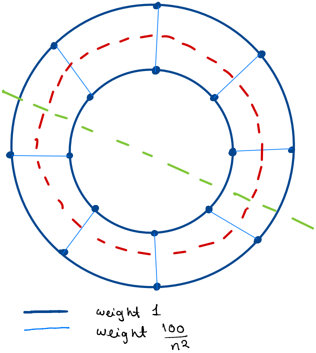
By Cheeger, $\Phi_G = O(\sqrt{\lambda_2}) = O(n^{-1})$. On the other hand, $\Phi_G = \Omega(n^{-1})$.

Therefore,

$$\Phi_G = O(n^{-1}), \quad \lambda_2 = O(n^{-2}).$$

Interestingly, the sweep algorithm still produces the optimal cut $\phi(S^o) = \Theta(n^{-1})$.

Two cycles C_n + hidden matching.



The red cut is the optimal cut, $\Phi_G = \Theta(n^{-2})$.

Second eigenvector = two copies of second eigenvectors of $C_n \Rightarrow \lambda_2 = \Theta(n^{-2})$.

The sweep algorithm still produces the green cut $\phi(S^o) = \Theta(n^{-1})$.

Improved Cheeger

- The sweep algorithm outputs a set S^o such that
$$\phi(S^o) = O(\sqrt{\lambda_2}) = O(\sqrt{\Phi_G}).$$

- Kwok-Lau-Lee-Oveis Gharan-Trevisan '13: For any $k \geq 2$, the sweep algorithm outputs a set S^o such that
$$\phi(S^o) = O\left(\frac{k\lambda_2}{\sqrt{\lambda_k}}\right) = O\left(\frac{k\Phi_G}{\sqrt{\lambda_k}}\right).$$
For example, if $\lambda_3 \geq c$, we get a $O(1)$-approximation of Φ_G.

Beyond $k = 2$: higher order Cheeger

Order-k conductance. Let $k \geq 2$.

$$\Phi_G(k) = \min_{S_1, S_2, \ldots, S_k \textsf{ distjoint}} \max_i \phi(S_i).$$

$$\Phi_G(k) = 0 \Leftrightarrow G \text{ has } \geq k \text{ connected components} \Leftrightarrow \lambda_k = 0.$$

Lee-Oveis Gharan-Trevisan '12:

$$\frac{\lambda_k}{2} \leq \Phi_G(k) = O(k^2 \sqrt{\lambda_k}).$$

Furthermore, there is an algorithm (spectral embedding + geometric partitioning) which returns

k disjoint sets $S_1, S_2, \ldots, S_k$ such that $\max_i \phi(S_i) = O(k^2 \sqrt{\lambda_k})$.

References

- U von Luxburg (2007). A tutorial on spectral clustering
- D Spielman (2019). Spectral and Algebraic Graph Theory
- L Trevisan (2016). Lecture Notes on Graph Partitioning, Expanders and Spectral Methods
- A. Ng, M. Jordan, and Y. Weiss (2002). "On spectral clustering: Analysis and an algorithm".
- J Shi, and J. Malik (2000). Normalized cuts and image segmentation.