A spectral-based clustering algorithm for directed graphs

Chase King
chasek22@uw.edu

CSE 521: "Design and Analysis of Algorithms" — Fall 2020

1 Introduction

Fundamentally, a graph is a set of vertices and edges, where each edge connects two vertices. Graphs can be undirected (where if vertex v is connected to vertex u then vertex u is connected to vertex v), or directed (where an edge from v to u does not necessarily imply the converse). Graphs can be used to represent data in a variety of applications; examples include users in a social network (where nodes are users and edges are friends) to functional connectivity in the brain (where nodes are brain areas and edges represent connection strengths between areas).[1] A fundamental algorithmic graph theory problem is clustering: how can we define the notion of a "cluster" in a graph, and how can we design efficient algorithms to identify such clusters? Clustering is a natural choice to analyze graph-based datasets. Returning to our previous two examples, clusters can represent friend groups in a social network or brain areas with spatiotemporally correlated activity.

In this paper we will address a spectral clustering method for undirected graphs proposed in Laenen and Sun (2020). This method maximizes the "flow ratio" between clusters and relies on using a Hermitian adjacency matrix as a directed graph representation (Cucuringu, Li, Sun, & Zanetti, 2019). We will see how clusters are approximately encoded in the bottom eigenvector of the normalized Laplacian matrix associated with this Hermitian adjacency matrix. Finally, we will discuss an algorithm designed to recover these structures (Laenen & Sun, 2020).

1.1 Acknowledgements

Thanks to Steinar Laenen, the primary author of the paper I reviewed, for generously answering a few questions that came up when I was writing this review paper. Also thanks to Professor Shayan Oveis Gharan for being open to letting undergraduates take his course and for introducing me to many new algorithmic topics like spectral graph theory that I otherwise wouldn't have been exposed to. Finally thanks to the teaching assistants Farzam and Kuikui for their generous help in office hours.

1.2 Previous Work

Most literature on graph clustering has focused on undirected graphs. Objective functions that these clustering algorithms seek to optimize include the clustering coefficient (where

[1] Note that the former example would typically be represented as an undirected graph (since in a network like Facebook, "A is a friend of B" implies that "B is a friend of A"), whereas the latter example would lend itself to an undirected graph (e.g. in the case that activity in one brain area incites activity in another area, but not the other way around).

we count the number of triangles in a graph[2]) and conductance (which is the total weight of edges going out of a cluster divided by the sum of degrees of vertices in the cluster[3]). At a high level, these objective functions tend to put vertices in a cluster when they are better connected to other vertices in the cluster than to the rest of the graph.

However, as Laenen and Sun (2020) point out, these algorithms are typically unable to identify useful clusters in directed graphs. As an example, they give the international oil trade network graph, which is a digraph of fuel and oil trading across the globe. In this graph, an edge from u to v indicates country u selling country v goods, and the weight indicates the volume of trade. Using typical undirected graph clustering objective functions (as noted above), little structural information is gleaned from this graph. However, there are clearly structures in this graph: countries that primarily export oil can be clustered together, for example. They state that "these clusters are characterised by the imbalance of the edge directions between clusters" and refer to these types of patterns as "higher-order structure among the clusters" (Laenen & Sun, 2020).

2 Main contributions

The main purpose of Laenen and Sun (2020) is to uncover higher-order structure of clusters in a directed graph; there is no previous work that analyzes digraph spectral clustering algorithms.

2.1 Definitions

We begin by enumerating some key notations used in this paper[4]:

- $G = (V, E, w)$ is a digraph with $|V| = n$ vertices, $|E| = m$ edges, and edge weight function $w : V \times V \to [0, \infty)$.
- k is the number of clusters we wish to form.
- $u \rightsquigarrow v$ means there is a (directed) edge from vertex u to vertex v.
- Total degree of a vertex: $\deg(v) = \sum_{u:\ u \rightsquigarrow v} w(u, v) + \sum_{u:\ v \rightsquigarrow u} w(v, u)$ (in degree plus out degree).
- Volume of a set of vertices $S \subseteq V$: $\mathrm{vol}(S) = \sum_{v \in S} \deg(v)$.
- Adjacency matrix $M \in \mathbb{R}^{n \times n}$ where $M_{i,j} = w(i, j)$ if $i \rightsquigarrow j$ and 0 otherwise.
- Hermitian adjacency matrix $A \in \mathbb{C}^{n \times n}$, where $A_{u,v} = \overline{A_{v,u}} = w(u, v) \cdot \omega_{\lceil 2\pi k \rceil}$ if $u \rightsquigarrow v$ and zero otherwise. (Here, $\overline{a + bi} = a - bi$ is the complex conjugate and $\omega_{\lceil 2\pi k \rceil}$ is a $\lceil 2\pi k \rceil$-th root of unity) (Cucuringu et al., 2019)
- Normalized Laplacian matrix $\mathcal{L} = D^{-1/2}(D - A)D^{-1/2} = I - D^{-1/2}AD^{-1/2}$ where $D = \mathrm{diag}\{\deg(v)\}_{v=1}^{n}$.

[2]The global clustering coefficient is the ratio of closed triplets in a graph to the total number of (open and closed) triplets.

[3]The optimal conductance is the minimum because we want vertices to be better connected with other vertices in the cluster and not so much with vertices out of the cluster

[4]I slightly changed some of the notation that was used in Laenen and Sun (2020) to conform to both what I am used to and what was used in class, so notation in this paper may not exactly match notation in Laenen and Sun (2020)

- For a Hermitian matrix[5] $B \in C^{n \times n}$ we have eigenvalues $\lambda_1(B) \leq \lambda_2(B) \leq \cdots \leq \lambda_n(B)$ with corresponding eigenvectors $v_i \in \mathbb{C}^n$.
- For a complex vector $x \in \mathbb{C}^n$, x^* is the complex conjugate transpose of x.
- Finally, for indexing vectors we will denote the jth element ($1 \leq j \leq n$) of some vector $x \in \mathbb{C}^n$ by $x[j]$ (brackets are chosen instead of subscripts to avoid confusion when vectors are defined with subscripts).

2.2 Objective function: Flow Ratio

Suppose $\{S_i\}_{i=0}^{k-1}$ forms a k-way partition of the vertices V.[6] The *flow ratio* is defined as

$$\Phi(S_0, ..., S_{k-1}) = \sum_{j=1}^{k-1} \frac{w(S_j, S_{j-1})}{\text{vol}(S_j) + \text{vol}(S_{j-1})} \tag{1}$$

where

$$w(S,T) = \sum_{\substack{u \rightsquigarrow v \\ u \in S,\ v \in T}} w(u,v) \tag{2}$$

is the cut value from S to T and $\text{vol}(S)$ (defined above) is the sum of the degrees of the vertices in S. An optimal partition is a partition that maximizes the flow ratio. Laenen and Sun (2020) shows that the optimal partition is embedded in the bottom eigenspace of $\mathcal{L}$. This objective function contrasts conductance because we are trying to maximize the flow ratio, whereas with conductance-based objective functions the goal is to minimize. Let θ_k define the optimum (i.e. max flow ratio) when vertices of G are partitioned into k clusters.

2.3 Embedding of optimal cluster into bottom eigenspace of $\mathcal{L}$

A key result of Laenen and Sun (2020) is that the optimal partition is embedded into the bottom eigenspace of $\mathcal{L}$. To study this relationship, pick some optimal clustering $\{S_j\}_{j=0}^{k-1}$ (i.e. one that maximizes the flow ratio) and define for every S_j a complex indicator vector $x_j \in \mathbb{C}^n$ whose uth entry is $(\omega_{\lceil 2\pi k \rceil})^j$ if $u \in S_j$, and zero otherwise. Furthermore, define the normalized indicator random vector

$$\widehat{x_j} = \frac{D^{1/2} x_j}{\|D^{1/2} x_j\|} \tag{3}$$

and let

$$y = \frac{1}{\sqrt{k}} \sum_{j=0}^{k-1} \widehat{x_j}. \tag{4}$$

Note that $y \in \mathbb{C}^n$ can encode information of all clusters simultaneously due to the use of complex numbers and roots of unity in the definition of v_j (different clusters are separated by angles). This result hints that, unlike in the case of undirected graphs where k (perpendicular) eigenvectors are needed to study the cluster structure, a single eigenvector of $\mathcal{L}$ can yield information about all k clusters (Laenen & Sun, 2020). This is a core component that makes this paper different from previous work, and is formalized below.

---

[5] A Hermitian matrix is one that is equal to its conjugate transpose; i.e., $A \in \mathbb{C}^{n \times n}$ is Hermitian if and only if $A = \overline{A^T}$ where the conjugate operation is taken elementwise.

[6] That is, the union of all the S_i's is V and each one is disjoint from the others.

Theorem 1. *Let $G = (V, E, w)$ be a weighted directed graph with normalized Hermitian Laplacian matrix $\mathcal{L} \in \mathbb{C}^{n \times n}$. Suppose we want to form k clusters. Then, if θ_k is the optimal max flow ratio of G, we have*

$$\lambda_1(\mathcal{L}) \leq 1 - \frac{4}{k}\theta_k.$$

Proof. The Rayleigh quotient for $\mathcal{L}$ w.r.t. y is given by $\frac{y^* \mathcal{L} y}{y^* y}$. However, note that

$$y^* y = \frac{1}{k} \sum_{j=0}^{k-1} \widehat{x_j}^* \widehat{x_j} = \frac{1}{k} \sum_{j=0}^{k-1} k = 1$$

since $\widehat{x_j}$ given in (3) is normalized to unit length, so it suffices to analyze the numerator of the Rayleigh quotient, $y^* \mathcal{L} y$. First, applying the definitions of y and $\mathcal{L}$, we have that

$$y^* \mathcal{L} y = \frac{1}{k} \left(\sum_{j=0}^{k-1} \widehat{x_j} \right)^* (I - D^{-1/2} A D^{-1/2}) \left(\sum_{j=0}^{k-1} \widehat{x_j} \right)$$

Now distributing this into two parts, we first have that

$$\frac{1}{k} \left(\sum_{j=0}^{k-1} \widehat{x_j} \right)^* I \left(\sum_{j=0}^{k-1} \widehat{x_j} \right) = \frac{1}{k} \sum_{j=0}^{k-1} \sum_{\ell=0}^{k-1} (\widehat{x_j})^* \widehat{x_\ell} = \frac{1}{k} \sum_{j=0}^{k-1} \widehat{x_j}^* \widehat{x_j} = 1$$

since by definition $(\widehat{x_j})^* \widehat{x_\ell} = 0$ for any $j \neq \ell$. For the second part, we have

$$
\begin{aligned}
&\frac{1}{k} \left(\sum_{j=0}^{k-1} \widehat{x_j} \right)^* (D^{-1/2} A D^{-1/2}) \left(\sum_{j=0}^{k-1} \widehat{x_j} \right) \qquad \text{(Def. } \widehat{x_j}\text{)} \\
&= \frac{1}{k} \left(\sum_{j=0}^{k-1} \frac{D^{1/2} x_j}{\|D^{1/2} x_j\|} \right)^* (D^{-1/2} A D^{-1/2}) \left(\sum_{j=0}^{k-1} \frac{D^{1/2} x_j}{\|D^{1/2} x_j\|} \right) \qquad (D^{1/2} \text{ real and diag)} \\
&= \frac{1}{k} \left(\sum_{j=0}^{k-1} \frac{x_j}{\|D^{1/2} x_j\|} \right)^* A \left(\sum_{j=0}^{k-1} \frac{x_j}{\|D^{1/2} x_j\|} \right) \\
&= \frac{1}{k} \left(\sum_{j=0}^{k-1} \frac{x_j}{\sqrt{\mathrm{vol}(S_j)}} \right)^* A \left(\sum_{j=0}^{k-1} \frac{x_j}{\sqrt{\mathrm{vol}(S_j)}} \right) \\
&= \frac{1}{k} \sum_{j=0}^{k-1} \sum_{\ell=0}^{k-1} \sum_{\substack{u \rightsquigarrow v \\ u \in S_j,\, v \in S_\ell}} \left(\frac{\overline{x_j[u]}}{\sqrt{\mathrm{vol}(S_j)}} \cdot A_{u,v} \cdot \frac{x_\ell[v]}{\sqrt{\mathrm{vol}(S_\ell)}} + \frac{\overline{x_\ell[v]}}{\sqrt{\mathrm{vol}(S_\ell)}} \cdot A_{v,u} \cdot \frac{x_j[u]}{\sqrt{\mathrm{vol}(S_j)}} \right) \\
&= \frac{1}{k} \sum_{j=0}^{k-1} \sum_{\ell=0}^{k-1} \sum_{\substack{u \rightsquigarrow v \\ u \in S_j,\, v \in S_\ell}} \frac{\overline{(\omega_{\lceil 2\pi k \rceil})^j} \cdot (w(u,v) \cdot \omega_{\lceil 2\pi k \rceil}) \cdot (\omega_{\lceil 2\pi k \rceil})^\ell + \overline{x_\ell[v]} \cdot \overline{(w(u,v) \cdot \omega_{\lceil 2\pi k \rceil})} \cdot (\omega_{\lceil 2\pi k \rceil})^j}{\sqrt{\mathrm{vol}(S_j)} \sqrt{\mathrm{vol}(S_\ell)}} \\
&= \frac{1}{k} \sum_{j=0}^{k-1} \sum_{\ell=0}^{k-1} \sum_{\substack{u \rightsquigarrow v \\ u \in S_j,\, v \in S_\ell}} \frac{w(u,v)}{\sqrt{\mathrm{vol}(S_j)} \sqrt{\mathrm{vol}(S_\ell)}} \cdot 2 \cdot \mathrm{Re}((\omega_{\lceil 2\pi k \rceil})^{\ell - j + 1})
\end{aligned}
$$

$$= \frac{1}{k}\sum_{j=0}^{k-1}\sum_{\ell=0}^{k-1}\sum_{\substack{u\rightsquigarrow v\\ u\in S_j,\ v\in S_\ell}} \frac{w(u,v)}{\sqrt{\mathrm{vol}(S_j)}\sqrt{\mathrm{vol}(S_\ell)}}\cdot 2\cos\left(\frac{2\pi(\ell-j+1)}{\lceil 2\pi k\rceil}\right)$$

$$\geq \frac{1}{k}\sum_{j=0}^{k-1}\sum_{\ell=0}^{k-1}\sum_{\substack{u\rightsquigarrow v\\ u\in S_j,\ v\in S_\ell}} \frac{w(u,v)}{\sqrt{\mathrm{vol}(S_j)}\sqrt{\mathrm{vol}(S_\ell)}}\cdot\left(2-\left(\frac{2\pi(\ell-j+1)}{\lceil 2\pi k\rceil}\right)^2\right) \quad (1-\cos(x)\leq \frac{1}{2}x^2)$$

$$\geq \frac{1}{k}\sum_{j=0}^{k-1}\sum_{\ell=0}^{k-1} \frac{w(u,v)}{\sqrt{\mathrm{vol}(S_j)}\sqrt{\mathrm{vol}(S_\ell)}}\cdot\left(2-\left(\frac{\ell-j+1}{k}\right)^2\right) \quad (\lceil 2\pi k\rceil \geq 2\pi k)$$

Then, combining the above parts, we have that

$$\begin{aligned}
y^*\mathcal{L}y &\leq 1-\frac{1}{k}\sum_{j=0}^{k-1}\sum_{\ell=0}^{k-1}\frac{w(u,v)}{\sqrt{\mathrm{vol}(S_j)}\sqrt{\mathrm{vol}(S_\ell)}}\cdot\left(2-\left(\frac{\ell-j+1}{k}\right)^2\right)\\
&= 1-\frac{2}{k}\sum_{j=0}^{k-1}\sum_{\ell=0}^{k-1}\frac{w(u,v)}{\sqrt{\mathrm{vol}(S_j)}\sqrt{\mathrm{vol}(S_\ell)}}+\frac{1}{k}\sum_{j=0}^{k-1}\sum_{\ell=0}^{k-1}\frac{w(u,v)}{\sqrt{\mathrm{vol}(S_j)}\sqrt{\mathrm{vol}(S_\ell)}}\left(\frac{\ell-j+1}{k}\right)^2\\
&\leq 1-\frac{2}{k}\sum_{j=0}^{k-1}\sum_{\ell=0}^{k-1}\frac{2w(u,v)}{\sqrt{\mathrm{vol}(S_j)}\sqrt{\mathrm{vol}(S_\ell)}}+\frac{1}{k}\sum_{j=0}^{k-1}\sum_{\substack{0\leq\ell\leq k-1\\ \ell\neq j-1}}\frac{2w(u,v)}{\sqrt{\mathrm{vol}(S_j)}\sqrt{\mathrm{vol}(S_\ell)}}\left(\frac{\ell-j+1}{k}\right)^2\\
&\leq 1-\frac{1}{k}\sum_{j=0}^{k-1}\sum_{\substack{0\leq\ell\leq k-1\\ \ell\neq j-1}}\frac{2w(u,v)}{\sqrt{\mathrm{vol}(S_j)}\sqrt{\mathrm{vol}(S_\ell)}}\left(1-\left(\frac{\ell-j+1}{k}\right)^2\right)-\frac{1}{k}\sum_{j=1}^{k-1}\frac{2w(S_j,S_{j-1})}{\sqrt{\mathrm{vol}(S_j)}\sqrt{\mathrm{vol}(S_{j-1})}}\\
&\leq 1-\frac{1}{k}\sum_{j=1}^{k-1}\frac{2w(S_j,S_{j-1})}{\sqrt{\mathrm{vol}(S_j)}\sqrt{\mathrm{vol}(S_{j-1})}}\\
&\leq 1-\frac{4}{k}\sum_{j=1}^{k-1}\frac{w(S_j,S_{j-1})}{\mathrm{vol}(S_j)+\mathrm{vol}(S_{j-1})} \quad \text{(AM-GM)}\\
&= 1-\frac{4}{k}\theta_k
\end{aligned}$$

By the Raleigh principle for calculating the lowest eigenvalue, we have that

$$\lambda_1(\mathcal{L}) = \min_{\substack{x\in C^n\\ x\neq 0}}\frac{x^*\mathcal{L}x}{x^*x}\leq\frac{y^*\mathcal{L}y}{y^*y}\leq 1-\frac{4}{k}\theta_k$$

hence we have found the desired upper bound on the lowest eigenvalue of $\mathcal{L}$. □

Theorem 1 gives some intuition about the lowest eigenvalue, but what about the lowest eigenvector? We first introduce a scalar parameter $\gamma\in\mathbb{R}$ (Laenen & Sun, 2020):

$$\gamma_k = \frac{\lambda_2(\mathcal{L})}{1-(4/k)\theta_k} \tag{5}$$

Note that following from Theorem 1, γ_k is essentially a lower bound on the second lowest eigenvalue of $\mathcal{L}$ divided by the lowest eigenvalue of $\mathcal{L}$. Recall that the vector y (defined in (4)) encodes the information of an optimal clustering. Theorem 2 below shows that y can be approximated from v_1, the lowest eigenvector of $\mathcal{L}$, with some approximation ratio inversely proportional to γ_k.

Theorem 2. *(1) There is some $\alpha \in \mathbb{C}$ such that $\|y - \alpha v_1\|^2 \leq 1/\gamma_k$. (2) There is some $\beta \in \mathbb{C}$ such that $\|v_1 - \beta y\| \leq 1/(\gamma_k - 1)$.*

Proof. We will prove the first statement; for the proof of the second statement, refer to Laenen and Sun (2020). First write $y = \sum_{i=1}^{n} \alpha_i v_i$ as a linear combination of the eigenvectors of $\mathcal{L}$ ($\alpha_i \in \mathbb{C}$). Taking the Rayleigh quotient, we have

$$\begin{aligned}
\frac{y^* \mathcal{L} y}{y^* y} &= \left(\sum_{i=1}^{n} \alpha_i v_i\right)^* \mathcal{L} \left(\sum_{i=1}^{n} \alpha_i v_i\right) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \overline{\alpha_i} \cdot \alpha_j \cdot v_i^* \mathcal{L} v_j \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \overline{\alpha_i} \cdot \alpha_j \cdot \lambda_j(\mathcal{L}) \cdot v_i^* v_j \\
&= \sum_{j=1}^{n} \overline{\alpha_j} \cdot \alpha_j \cdot \lambda_j(\mathcal{L}) \cdot v_j^* v_j \\
&= \|\alpha_1\|^2 \lambda_1(\mathcal{L}) + \cdots + \|\alpha_n\|^2 \lambda_n(\mathcal{L}) \\
&\geq \|\alpha_1\|^2 \lambda_1(\mathcal{L}) + (\|\alpha_2\| + \cdots + \|\alpha_n\|^2) \lambda_2(\mathcal{L}) \\
&\geq \|\alpha_1\|^2 \lambda_1(\mathcal{L}) + (1 - \|\alpha_1\|^2) \lambda_2(\mathcal{L}) \\
&\geq (1 - \|\alpha_1\|)^2 \lambda_2(\mathcal{L})
\end{aligned}$$

where the first inequality follows from the ordering of the eigenvalues and the second inequality follows from the fact that $\sum \|\alpha_i\|^2 = 1$ (since y and all v_i are unit length). We then have

$$\|y - \alpha_1 v_1\|^2 = \left\|\sum_{i=2}^{n} \alpha_i v_i\right\|^2 = \sum_{i=2}^{n} \|\alpha_i\|^2 = 1 - \|\alpha_1\|^2 \leq \frac{y^* \mathcal{L} y}{y^* y \cdot \lambda_2(\mathcal{L})} \leq \frac{1}{\gamma_k}$$

where the second equality follows since $v_i^* v_j = 0$ for $i \neq j$, the third equality follows from the fact above, the first inequality follows from the observation above, and the second inequality follows from the definition of γ_k (and the observation in the statement following (5)). Thus setting $\alpha = \alpha_1$ proves the desired claim. □

2.4 Algorithm

The main algorithm proposed in Laenen and Sun (2020) exploits the fact that the cluster structure is roughly encoded in the bottom eigenvector of $\mathcal{L}$. The algorithm consists of three steps (Laenen & Sun, 2020):

1. Compute the bottom eigenvector $v_1 \in \mathbb{C}^n$ of $\mathcal{L}$.
2. Embed the vertices of G into $\mathbb{R}^2$ by mapping each vertex v through the function

$$F(v) = \frac{1}{\sqrt{\deg(v)}} \cdot v_1[v] \tag{6}$$

   where the latter term is the vth entry of the bottom eigenvector.[7]

[7] Technically, the function in (6) maps vertices to the complex plane $\mathbb{C}$, but this can be thought of as $\mathbb{R}^2$.

3. Run k-means on the embedded points and return clusters $A_0, \ldots, A_{k-1}$.

For more detailed theorems on the analysis of the above algorithm, refer to Laenen and Sun (2020). One key implication is that the embedded vertices in a smaller cluster are far from embedded vertices in other clusters. Thus the algorithm can do a good job of approximating the structure of all clusters.

3 Conclusion

In this paper we have introduced a clustering algorithm for directed graphs that relies on the lowest eigenvector of the normalized Hermitian Laplacian matrix $\mathcal{L}$. The use of complex numbers and roots of unity yields clusters that are separated by angles, enabling information about the optimal clustering to be gleaned from only the lowest eigenvector of $\mathcal{L}$.

Though it was not discussed in the present paper, Laenen and Sun (2020) discusses an application of this algorithm to finding clusters in trade network graphs. Another interesting and particularly relevant application for future work would be for modeling infectious diseases. For example, contact tracing databases could yield a directed graph representing transmissions of COVID-19 between individuals in some community. Clusters in this graph would represent highly infectious groups of people, or groups of people most susceptible to infection. Work along these lines could have social impact by providing useful information to the containment policies of a particular community.

References

Cucuringu, M., Li, H., Sun, H., & Zanetti, L. (2019). *Hermitian matrices for clustering directed graphs: insights and applications.*

Laenen, S., & Sun, H. (2020). *Higher-order spectral clustering of directed graphs.*