# Spectral Clustering Analysis*

Yueshuwei Wu, Yifei Li, Chenxi Jiang

*Abstract*— **Spectral Clustering is one of the most popular clustering techniques. It uses the eigenvectors of similarity matrices derived from the data to form clusters. We illustrate Spectral Clustering's mathematical foundations and demonstrate this algorithm's application on synthetic data and the Twitter dataset.**

## I. INTRODUCTION

The rapid penetration of the Internet has facilitated the growing popularity of numerous social networks and their respective applications, enabling social networks to play progressively vital roles in spreading information and aggregating public opinions. In practice, many social networks can be effectively modeled as a collection of communities, in which each community has significantly more interactions locally than with users in other communities. Within many social networks, the direct connections among the network users are established by personal ties in the first place. Over time, the community structure may evolve with membership changing due to various reasons like political opinions.

Community detection is the key to understanding complex networks' structures and ultimately extracting useful information from them. Since there is no coordinated system for networks, detecting community structures is a challenging problem in many scientific fields. In this paper, we intend to investigate a machine learning approach for community detection in social networks using the contents users produced on the Internet.

Due to modern technology development, a general segmentation process is not usually feasible for large volumes of very data. Therefore we need an analytical approach to derive segments and groups from large datasets. In the field of machine learning, clustering is a critical step to process the data samples for the purpose of dimensional reduction or pattern detection. Generally speaking, detecting clusters is similar to detecting communities in social networks. A cluster is a group of data points or objects in a dataset similar to other objects in the group and dissimilar to data points in other clusters.

Clustering is unsupervised machine learning and groups the data into different segmentation where the data is unlabeled. Many clustering techniques and models have been developed over the years, including k-means, hierarchical clustering, density-based clustering, and graph-based clustering. It is worth pointing out that density-based clustering can be view as a special case of spectral clustering[4], but one which allows more efficient algorithms. In this paper, we will focus on spectral clustering and its applications.

The work in this paper is motivated by the following facts:

1) Because of the large size of the social network, it is unrealistic to rely on a single node or a sub-graph to detect the similarity-based communities.
2) Online communities should share common characteristics in terms of their features and form clusters in their features space.
3) We want to experiment with machine learning models to see if results similar to network studies could be obtained.

The rest of this paper is organized as follows. We first present the mathematical foundations of spectral clustering by formulating the problem and illustrating the definitions and derivations. We then discuss the drawbacks of spectral clustering and some improvements. Finally, we experiment with spectral clustering on synthetic data and social media data and compare it with other clustering models.

## II. MATHEMATICAL FOUNDATIONS

In this section, we present the mathematical foundations of spectral clustering. First, we explain the graphical view of the spectral clustering and the graph cut approach. We then discuss application to high-dimensional data. Finally, We discuss the drawbacks of spectral clustering and some improvements.

### A. Undirected weighted graph

Since spectral clustering is based on graph theory, we first review the concept of the graph below. For a graph $G$, we usually use the vertex set $V$ and the set of edges $E$ to describe it. It is called $G(V, E)$ where $E$ is all the vertices in our dataset ($v_1$, $v_2$,... $v_n$). For any two vertices in $V$, there can be edge connection or no edge connection. We define the weight $w_{i,j}$ as the weight between the vertex $v_i$ and the vertex $v_j$ , since $G$ is an undirected graph, $w_{i,j} = w_{j,i}$

For two vertices $v_i$ and $v_j$ with edge connection, $w_{i,j} > 0$, for two vertices $v_i$ and $v_j$ without edge connection, $w_{i,j} = 0$. For any vertex $v_i$ in a graph, its degree $d_i$ is defined as the sum of the weights of all the edges connected to it

$$d_i = \sum_{j=1}^{N} w_{i,j}$$

Using the definition of the degree of each vertex, we can get a NxN degree matrix $D$, which is a diagonal matrix. Only the main diagonal has a value, corresponding to the degree of the i-th point in the i-th line

*This work was not supported by any organization.

Presented by students from Department of Mathematics, University of California, Los Angeles, CA, 90024

$$D = \begin{bmatrix} d_1 & \cdots & \cdots & \cdots \\ \vdots & d_2 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & d_n \end{bmatrix}$$

Using the weight values of all points, we can get the adjacency matrix $W$ which is also an NxN matrix. The j-th value of the i-th row corresponds to our weight $w_{ij}$ In addition, for a subset of the point set $V$, where $A \subset V$ we define

$$|A| := \text{The number of vertices in a subset } A$$

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

### B. Different Similarity Graphs

To transform a given set $x_1, ..., x_n$ of data points with pairwise similarities $s_{i,j}$ into a graph, we need to first construct the similarity matrix, then consider it as the matrix representation of the graph. There are three commonly used methods for constructions:

1) $\varepsilon$-neighborhood
2) $k$-nearest neighborhood
3) fully connected

The first two methods can construct a sparse matrix, which is suitable for large sample projects. However, in large sample projects, the iterative speed will be affected and restricted.Thus, the third one is quite different with the first two. Before explaining the construction methods, we need to firstly introduce the similarity function, that is, to calculate the distance between two sample points. Generally, Euclidean distance is used as $s_{i,j} = ||x_i - y_i||^2$, $s_{i,j}$, which is the distance between sample points $x_i$ and $x_j$, or using Radial Basis Function: $s_{ij}$= $e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}$, where the selection of $\sigma$ affects the result.This expression showed the degree of dispersion of data distribution. Using one of the above two ways, the matrix $S : S_{i,j} = [S]_{i,j}$ can be initially constructed, which is generally called similarity matrix.

For the first kind of composition $\epsilon$-neighborhood, it takes the vertices of $S_{i,j} \leq \epsilon$, then the similarity matrix s can be further reconstructed into adjacency matrix $W$:

$$W_{i,j} = \begin{cases} 0, s_{i,j} > \epsilon \\ \epsilon, s_{i,j} \leq \epsilon \end{cases} \tag{1}$$

The above formula indicates that the weight between the two points is either $\epsilon$ or 0. Hence, the distance measurement is not accurate and $\epsilon$-neighborhood method is not widely used.

For $k$-nearest neighborhood composition, it uses $KNN$ algorithm to traverse all sample points, and takes the nearest k points of each sample as the nearest neighbors. However, this method will cause adjacency matrix $W$ to be asymmetric after reconstruction. In order to solve this problem, two methods are generally adopted below:

The first method is that only one vertex is in the k-nearest neighbor of another point, then $s_{i,j}$ can be remained, so

$$\text{W}_{i,j} = W_{j,i} = \begin{cases} 0, x_i \notin KNN(X_j) \; and \; x_j \notin KNN(X_i) \\ e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}, x_i \in KNN(X_j) \; or \; x_j \in KNN(X_i) \end{cases} \tag{2}$$

The second method is that two vertices must be k-nearest neighbors for each other, then $s_{i,j}$ can be remained, so

$$\text{W}_{i,j} = W_{j,i} = \begin{cases} 0, x_i \notin KNN(X_j) \text{ or } x_j \notin KNN(X_i) \\ e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}, x_i \in KNN(X_j) \text{ and } x_j \in KNN(X_i) \end{cases} \tag{3}$$

For fully connected construction, we simply connect all points with positive similarity with each other, and we weight all edges by $s_{i,j}$. As the graph should represent the local neighborhood relationships, this construction is only useful if the similarity function itself models local neighborhoods. Different kernel functions can be selected to define the edge weights. The most common one is the Gaussian kernel function RBF. In this case, the similarity matrix and the adjacency matrix are the same:

$$W_{i,j} = s_{i,j} = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}$$

where the parameter $\sigma$ controls the width of the neighborhoods. This parameter plays a similar role as the parameter $\epsilon$ in case of the $\epsilon$-neighborhood graph.

### C. Graph Laplacians

The standard graph Laplacian matrix is defined as

$$L = D - W$$

**Proposition 1 (Properties of L)** The matrix $L$ satisfies the following properties [9]:

1) For every vector $f \in \mathbb{R}^n$, we have
$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{i,j}(f_i - f_j)^2$$
2) $L$ is symmetric and positive semi-definite.
3) The smallest eigenvalue of $L$ is 0, the corresponding eigenvector is the constant one vector $\vec{1}$.
4) $L$ has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$

**Proposition 2 (Number of connected components and the spectrum of L)**

Let $G$ be an undirected graph with non-negative weights. Then the multiplicity $k$ of the eigenvalue 0 of $L$ equals the number of connected components $A_1, ..., A_k$ in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, ..., \mathbb{1}_{A_k}$ of those components.

The normalized graph Laplacians are defined as

$$L_{sym} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

$$L_{rw} = D^{-1} L = I - D^{-1} W$$

We denote the first matrix by $L_{sym}$ as it is a symmetric matrix, and the second one by $L_{rw}$ as it is closely related to a random walk.

**Proposition 3 (Properties of $L_{sym}$ and $L_{rw}$)**

The normalized Laplacians satisfy the following properties [3]:

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

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

2. $\lambda$ is an eigenvalue of $L_{rw}$ with eigenvector $u$ if and only if $\lambda$ is an eigenvalue of $L_{sym}$ with eigenvector $w = D^{1/2}u$.

3. $\lambda$ is an eigenvalue of $L_{rw}$ with eigenvector $u$ if and only if $\lambda$ and $u$ solve the generalized eigen- problem $Lu = \lambda Du$.

4. 0 is an eigenvalue of $L_{rw}$ with the constant one vector $\vec{1}$ as eigenvector. 0 is an eigenvalue of $L_{sym}$ with eigenvector $D^{1/2}\vec{1}$.

5. $L_{sym}$ and $L_{rw}$ are positive semi-definite and have n non-negative real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$.

**Proposition 4 (Number of connected components and spectra of $L_{sym}$ and $L_{rw}$)** Let $G$ be an undirected graph with non-negative weights. Then the multiplicity $k$ of the eigenvalue 0 of both $L_{rw}$ and $L_{sym}$ equals the number of connected components $A_1, ..., A_k$ in the graph. For $L_{rw}$ , the eigenspace of 0 is spanned by the indicator vectors $\vec{1}$ of those components. For $L_{sym}$, the eigenspace of 0 is spanned by the vectors $D^{1/2}\vec{1}A_i$ .

## D. Graph Cut on Spectral Clustering

The intuition behind any clustering method is the separation of points based on their similarities, and the separation is essentially a cut on the graph. First we introduce what a graph cut is. Suppose that we are separating a set of points into two disjoint groups. Those two groups of vertices are in the graph $S$ and $\overline{S}$ where $S \cap \overline{S} = \emptyset$ and $S \cup \overline{S} = V$. We define a cut with respect to $S \subseteq V$ to be

$$cut(S) = \sum_{i \in S, j \in \overline{S}} w_{ij} = cut(\overline{S})$$

which is the total sum of the edge weights whose two end point (vertices) are in different groups. It is clear that smaller $cut(S)$ is, fewer connections between $S$ and $\overline{S}$.

However, minimizing $cut(S)$ would not be a good partition of the graph because the cut can choose a single point as a cluster. In order for the cut to give us balanced clusters, we introduce ratio cut ($Rcut$) and normalized cut ($Ncut$) defined as

$$Rcut = \frac{cut(S)}{|S|} + \frac{cut(\overline{S})}{|S|}$$

$$Ncut = \frac{cut(S)}{vol(S)} + \frac{cut(\overline{S})}{vol(\overline{S})}$$

where $vol(S)$ is the volume of $S$ defined as $vol(S) = \sum_{i \in S} \sum_{j \in V} w_{ij}$.

Given the same value of $cut(S)$, both $Rcut(S)$ and $Ncut(S)$ will be smaller if the size of two clusters are approximately the same. Hence, we can get balanced clusters. However, $Rcut(S)$ and $Ncut(S)$ are difficult to minimize because they are discrete optimization that require us to use brute force to calculate $Rcut(S)$ and $Ncut(S)$ with respect to all possible subsets of $V$.

## E. Relaxation of Ratio Cut

Since it is NP-hard to minimize ratio cut , we can take an alternative route to approximate the minimum of ratio cut[15]. We define a simple step function $f_S : V \to \mathbb{R}$ to be

$$f_S(i) := \begin{cases} \sqrt{\frac{|\overline{S}|}{|V||S|}}, i \in S \\ -\sqrt{\frac{S}{|V||\overline{S}|}}, i \in \overline{S} \end{cases} \tag{4}$$

In particular, we have $f_S(i) = -f_{\overline{S}}(i)$.

**Proposition 5** For all $f_S$, we have

$$f_S^T L f_S = Rcut(S)$$

**Proof:**

$$\begin{aligned} f_S^T L f_S &= \frac{1}{2} \sum_{i \in V, j \in V} w_{ij} (f_S(i) - f_S(j))^2 \\ &= \frac{1}{2} \left( \sum_{i \in S, j \in \overline{S}} + \sum_{i \in \overline{S}, j \in S} \right) w_{ij} (f_S(i) - f_S(j))^2 \\ &= \sum_{i \in S, j \in \overline{S}} w_{ij} (f_S(i) - f_S(j))^2 \\ &= \sum_{i \in S, j \in \overline{S}} w_{ij} \left( \sqrt{\frac{|\overline{S}|}{|V||S|}} + \sqrt{\frac{S}{|V||\overline{S}|}} \right)^2 \\ &= \frac{1}{|V|} \sum_{i \in S, j \in \overline{S}} w_{ij} \left( \frac{|\overline{S}|}{|S|} + \frac{|S|}{|\overline{S}|} + 2 \right) \\ &= \frac{1}{|V|} \sum_{i \in S, j \in \overline{S}} w_{ij} \left( \frac{|\overline{S}|}{|S|} + \frac{|S|}{|\overline{S}|} + \frac{|S|}{|S|} + \frac{|\overline{S}|}{|\overline{S}|} \right) \\ &= \frac{1}{|V|} \sum_{i \in S, j \in \overline{S}} w_{ij} \left( \frac{|V|}{|S|} + \frac{|V|}{|\overline{S}|} \right) \\ &= \sum_{i \in S, j \in \overline{S}} w_{ij} \left( \frac{1}{|S|} + \frac{1}{|\overline{S}|} \right) = Rcut(S) \end{aligned}$$

□

**Proposition 6** For all $S \subseteq V$ and $f_S$, we have

$$||f_S|| = 1, f_S^T \mathbf{1} = 0$$

**Proof:**
By definition of $f_S$,

$$\begin{aligned} ||f_S||^2 &= \sum_{i \in V} |f_S(i)|^2 \\ &= \sum_{i \in S} \frac{|\overline{S}|}{|V||S|} + \sum_{i \in \overline{S}} \frac{|S|}{|V||\overline{S}|} \\ &= \frac{|\overline{S}|}{|V||S|}|S| + \sum_{i \in \overline{S}} \frac{|\overline{S}|}{|V||S|}|\overline{S}| = 1 \\ f_S^T \mathbf{1} &= \sum_{i \in V} |f_S(i)| \\ &= \sum_{i \in S} f_S(i) - \sum_{i \in \overline{S}} f_S(i) \\ &= \sqrt{\frac{|S||\overline{S}|}{|V|}} - \sqrt{\frac{|S||\overline{S}|}{|V|}} = 0 \end{aligned}$$

□

Note, these two properties about $f_S$ hold for all $S$, so we call

$$\min f^T L f, \text{s.t. } ||f||^2 = 1, f^T \mathbf{1} = 0$$

the *relaxation* of min $Rcut(S)$.
**Proposition 7** The minimizer of the relaxation is $v_2$, which is the eigenvector corresponding to the second smallest eigenvalue $\lambda_2$ of the Laplacian $L$.
**Proof:**
We know that $\mathbf{1}$ is inside the null space of $L$. For any $f$ that satisfies **Proposition 6**, we have $f = \sum_{i=1}^n \beta_i v_i$ where $v_{i}{}_{i=1}^n$ is an orthonormal basis in $\mathbb{R}^n$ and $\beta_i = f^T v_i$. From $f^T \mathbf{1} = 0$ and $\sum_{i=2}^n \beta_i^2 = 1$, we get $\beta_1 = 0$, and thus $f = \sum_{i=2}^n \beta_i v_i$ for all $f$.

$$f^T L f = \sum_{i=2}^n \lambda_i \beta_i^2 \geq \lambda_2 \sum_{i=2}^n \beta_i^2 = \lambda_2$$

□

In short, we have that $\{f_S\}_{S \subseteq V} \subseteq \{f : ||f|| = 1, f^T \mathbf{1} = 0\}$ and thus

$$Rcut(S) \geq \lambda_2, \ \min_{S \subseteq V} Rcut(S) \geq \lambda_2$$

### F. Relaxation of Normal Cut

Similar to ratio cut, normal cut also has a relaxed form. We define a simple step function $f_S : V \to \mathbb{R}$ to be

$$f_S(i) := \begin{cases} \sqrt{\frac{|vol(\overline{S})|}{|vol(V)||vol(S)|}}, i \in S \\ -\sqrt{\frac{vol(S)}{|vol(V)||vol(\overline{S})|}}, i \in \overline{S} \end{cases} \tag{5}$$

We can see that this setup is a replica of (4), the step function for relaxed ratio cut, with $S$, $\overline{S}$, and $V$ being replaced by $vol(S)$, $vol(\overline{S})$, and $vol(V)$.
**Proposition 8** For all $f_S$, we have

$$f_S^T L f_S = Ncut(S)$$

**Proposition 9** For all $S \subseteq V$ and $f_S$, we have

$$f_S^T D f_S = 1, f_S^T D \mathbf{1} = 0$$

Proofs for these two propositions above are omitted due to their similarity to **Proposition 5** and **Proposition 6**. Moreover, we know that $L_S = I - D^{-1/2} W D^{-1/2} = D^{-1/2} L D^{-1/2}$. Let $x = D^{-1/2} f$ and we get the relaxation for (5)

$$\min \ x^T L_S x, \ \text{s.t. } ||x|| = 1, x^T D^{-1/2} \mathbf{1} = 0$$

We know that $L_S D^{-1/2} = D^{-1/2} L \mathbf{1} = 0$, which means that $D^{-1/2}\mathbf{1}$ is in the null space of $L_S$. Based on **Proposition 7**, we can conclude that the minimizer is also $v_2$, the eigenvector corresponding to the second smallest eigenvalue.

**Algorithm 1** Spectral Clustering

```
Require: $\{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^D$
1: Construct a $N \times N$ similarity matrix using any suitable method. Treat the similarity matrix as a matrix representation of a graph $G$.
2: Preprocessing: build the Laplacian matrix L of the graph
3: Decomposition: map vertices to their corresponding entries in the second eigenvector of L.
4: Grouping: sort these entries and split the list in two to arrive at a graph partition.
5: Repeat steps 2 to 4 until stopping criteria are met.
```

### G. Variations of spectral clustering

*1) Normalized vs. unnormalized spectral clustering:* So far, we have only used the standard graph Laplacian to bipartite the graph. Some other versions of spectral clustering use normalized Laplacians to solve for graph cut. In previous sections, we introduced $L_{sym}$ and $L_{rw}$, the properties of these normalized graph Laplacians can be found in the standard reference Chung (1997)[3].

In [8], authors investigated that the normalized and unnormalized versions of the spectral clustering algorithm and found that the convergence of unnormalized spectral clustering is more challenging to handle than the normalized case. It turns out that while in the normalized case, spectral clustering usually converges to a nice partition of the data space, in the unnormalized case, the same only holds under strong additional assumptions which are not always satisfied. The study suggests that the normalized version of spectral clustering should be preferable in practice. This suggestion also extends to other applications of graph Laplacians, including semi-supervised learning.

*2) Different Partition Methods:* There are two common methods to divide a graph into more than two clusters. One intuitive approach is to recursively apply the bi-partitioning algorithm in a hierarchical divisive manner in the algorithm stated above [6]. In other words, divide the graph into two clusters, then further subdivide those clusters. However, that can be inefficient and unstable since we may encounter situations where a cluster should no longer be further divided into smaller clusters.

Instead, we can cluster using multiple eigenvectors [14]. As discussed in previous sections, the second smallest eigenvalue of the symmetric normalized Laplacian matrix $L_{sym}$ provides insights into finding cluster structures, especially the smaller the eigenvalue, the better the cut is. Therefore, the eigenvectors corresponding to sufficiently small eigenvalues (the third-smallest, the fourth-smallest .etc) can also be good tools for dividing the graph. Using each node's corresponding component in these eigenvectors as their features, we can cluster these nodes through other clustering methods such as k-means. This method is commonly used as a dimension reduction technique for clustering in recent practices.

The multiple eigenvector method is also more principled than the recurrence bi-partition method because it approximates the optimal k-way normalized cut, emphasizes cohesive clusters, and maps points to a well-separated embedded space [14]. Furthermore, using an eigenvector basis ensures that less information is lost since we can choose to keep the (more informative) components corresponding to bigger eigenvalues.

*3) Stopping Criteria:* Like many other clustering algorithms, a challenge with spectral clustering is that we need to specify the number of clusters $k$ in order to use it. In other words, the number of clusters must be predefined to make the clustering process feasible. Unfortunately, we do not know what a reasonable $k$ value would be most of the time.

In cluster analysis, the elbow method, proposed by Robert L. Thorndike in 1953[16], is heuristically used in determining the number of clusters in a data set. The idea behind this is to keep increasing the number of clusters until the diminishing return is observed. Also, in a more recent research[17], the authors presented a natural bicriteria measure for assessing the number of clusters that avoids the drawbacks of existing measures. However, since we limit ourselves to perform data clustering exclusively depending on the graph Laplacian matrix's eigenvalues, finding the exact number of clusters remains a critical challenge.

It is also worth pointing out that the second smallest eigenvalues' magnitude could serve as a stopping criterion for the spectral clustering algorithm. Logically, when eigenvalues become insignificantly large in magnitude, they present a pattern of disorder and will cut almost all found clusters into two smaller clusters, making the number of clusters nearly doubled.

*4) Relationship with DBSCAN:* Spectral clustering is also related to the DBSCAN clustering algorithm. Density-based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu in 1996 [5]. DBSCAN finds connected components that correspond to optimal spectral clusters but uses an asymmetric neighbor graph with edges removed when source points are not dense[13]. For theoretical interest, DBSCAN can be viewed as a special case of spectral clustering but one which allows more efficient algorithms (worst case $O(n^2)$) than standard spectral clustering implementations (usually $O(n^3)$). It is worth mentioning that DBSCAN does not require a predefined number of clusters to be feasible, which is desired. However, estimating critical parameters like neighborhood radius $Eps$ still requires complex algorithms [10], potentially becoming a drawback in some cases.

## III. EXPERIMENTS

In this section, we apply spectral clustering to both synthetic data and real Twitter data. The model we used in the experiments are the symmetric normalized Laplacian spectral clustering model, the DBSCAN model, and the k-means clustering model imported from the Scikit-learn Python machine learning library.

### A. Synthetic Data

In our first experiment, we build a synthetic data set on which we expect spectral clustering and DBSCAN will perform well. Specifically, we generate some points in $\mathbb{R}^2$ by plotting points of two concentric circles with a small amount of Gaussian noise, that is:

$$\mathbf{x}_{outer}^2 + \mathbf{y}_{outer}^2 = 1^2 + 0.1\epsilon$$
$$\mathbf{x}_{inner}^2 + \mathbf{y}_{inner}^2 = (0.5)^2 + 0.1\epsilon$$

where $x$ and $y$ are points on the two circles and $\epsilon$ is sampled from $\mathcal{N}(\mathbf{0}, I)$.

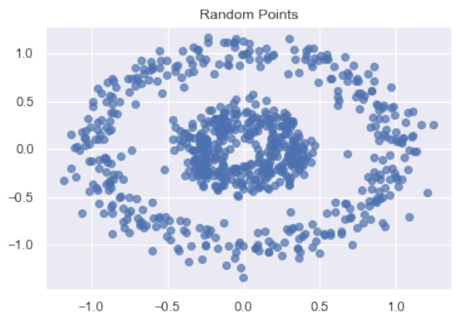

Fig. 1. Randomly generated points based on concentric circles.

Since the data set consists of non-spherical points sampled two concentric circles, we expect the spectral clustering model and DBSCAN to achieve much better results than the K-means clustering model. In Figure 1, we plot the sample points for this data set. In Figure 3 and Figure 2, the results of spectral clustering and DBSCAN model is almost perfect. In Figure 4, we plot the results of the K-mean cluster algorithm. Note that K-means gives a completely incorrect output and varies in every iteration. This is because the result of k-means depends on the initial centroids' values (called k-means seeding)[2]. Since K-means does not work with non-spherical data (and has other drawbacks), the output agrees with our expectations. The takeaway of this experiment is that when applying spectral clustering, we can make no assumptions on clusters' shape.

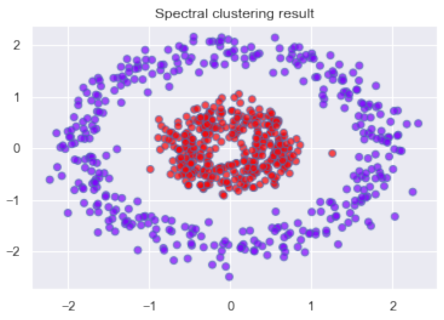

Fig. 2. Spectral clustering result.

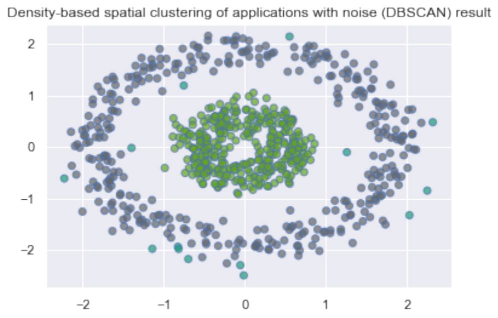

Fig. 3. Density-based spatial clustering of applications with noise (DBSCAN) result.

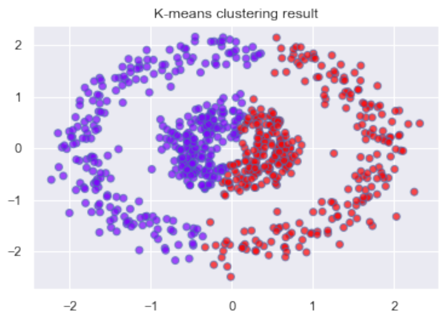

Fig. 4. K-means clustering result.

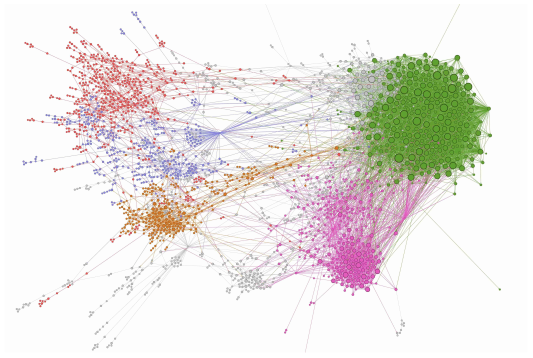

Fig. 5. ForceAtlas2 layout of the target sub-graph (network) consists of 5K nodes/users. Five largest communities were colored. A total number of 10 communities was detected.

## B. Twitter Data

We next apply spectral clustering to a retweet network datasets composed of 105K Twitter users and various features obtained from Kaggle [12].

*1) Backgound & Previous Study & Objective:* In this dataset, about 5K users are marked as "hateful"(∼500) or "normal"(∼4.5k) by Twitter using a classification model that detects hateful content (i.e. "hate speech"). Presented in Figure 5, our previous analysis of the marked users' network structure using the Louvain method for community detection [1] shows that about half of these marked users(∼2.5K) had formed 10 communities. This experiment's objective is to examine if the spectral clustering model can produce a cluster structure similar to or associate with the communities found in the network studies. We also want to see if spectral clustering can provide useful insight into users' distribution in the feature space.

*2) Model Assumptions:* One critical but reasonable assumption we made is that any user in a social network is able to find the distance between any other user and itself, provided that the users not directly interacting may still be able to observe the contents produced by others. We further assume that only the contents a user posts will significantly affect its distance from other users. Therefore we approach only using content-exclusive features calculated based on users' tweets.

*3) Feature Selection:* We use spaCy's off-the-shelf 300-dimensional GloVe's vector [11] that came with the dataset as features. The GloVe model is an unsupervised learning algorithm for obtaining vector representations for words. The algorithm also averaged the representation across all words in a given tweet, and subsequently, across all tweets that a user has, and assign it to the user. In other words, a user's feature vector is calculated based on all his posts, not just the recent ones. We refer to this feature matrix as **glove** where the row vectors are each user's GloVe's vector.

*4) Experimental Settings:* Recall the objective of this experiment is to compare the clustering results with the network communities. Considering that half of the marked users formed ten communities, and the other half were grouped into a giant null community due to the network structure, we set the desired clusters number $k$ to 11 to detect possible clustering structures within the null community.

Since the network community labels and the labels returned from the clustering algorithms can be viewed as categorical variables that describe a user, we will use the Chi-Square test of independence to determine if there is a statistically significant association between different labels. For the test of independence, also known as the test of homogeneity, a chi-squared probability of less than or equal to 0.05 (or the chi-squared statistic being at or larger than the 0.05 critical point) is commonly interpreted by applied workers as justification for rejecting the null hypothesis that the two variables are independent of each other

*5) Results:* We first applied DBSCAN and spectral clustering to the **glove** feature matrix. One giant cluster consists of about 80% (∼4000) users were found by the spectral clustering model, and the algorithm failed to find 11 clusters. The DBSCAN algorithm yields a similar result where only one big cluster was found, and about rest users were considered outliers.

To investigate further, we took the advice from [7] and normalized the **glove** matrix. We then successfully partitioned the data points into 11 clusters with reasonable sizes using spectral clustering, while DBSCAN still returns one giant cluster. However, We observed that in both versions of our spectra clustering experiment, the results seem to present a pattern of disorder and cut some clusters into two, producing clusters of similar sizes as shown in Table I. It is worth mentioning that the **glove** feature matrix entries measure the probability of co-occurrence. Thus the **glove** may not need to be normalized. The observed disorder could be the evidence of force splitting one good cluster multiple times, and the users may not be very distinguishable in the content-based feature space.

| Spectral Clustering Results | | | |
|---|---|---|---|
| Glove version | | normalized Glove version | |
| label | size of cluster | label | size of cluster |
| 1 | 4655 | 2 | 985 |
| 6 | 66 | 4 | 771 |
| 8 | 7 | 1 | 616 |
| 0 | 6 | 7 | 461 |
| 5 | 6 | 8 | **407** |
| 7 | 5 | 6 | **392** |
| 3 | 4 | 9 | **339** |
| 4 | 3 | 5 | **325** |
| 2 | 3 | 0 | **216** |
| 9 | 1 | 3 | **190** |
| | | 10 | 54 |

TABLE I

Spectral clustering results

We also applied the K-means clustering model. Unlike the synthetic data experiment, we got consistent results of clusters as shown in table II. This provides us valuable insights into the distribution of data points in high-dimensional space. We can conclude that the data is spherical, and the clusters can be separated by convex boundaries. Although we just argued that users are not significantly distinguishable based on content-exclusive features, we can intemperate this result as the users form smaller clusters that are close to each other. These smaller clusters are close enough to be considered a giant cluster by spectral clustering and DBSCAN.

We then compared the results from spectral clustering and k-mean clustering with the labels of network community structures. The colored patterns presented in Table III and Table IV show that both clustering models were able to find clusters based on content-exclusive features. Applying the Chi-Square test of independence, both cross tables yield a $p$ value close to 0, indicating the independence hypothesis can be rejected with a confidence of at least 95%. This means we have sufficient evidence to say that there is an association between clustering results and network communities.

Finally, we test the independence between results from spectral clustering and k-mean clustering. As shown in Table V, we observed some similarities in terms of clustering results. Applying the Chi-Square test of independence once again, we get an even smaller $p$ value close to 0. The minimal $p$ value signifies that the association between the two clustering results is statistically significant. In general, spectral clustering and k-means clustering helps give valuable insight into the potential community structure.

| K-Mean | K-Mean Cluster Labels 1 | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Cluster Labels 2** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 408 |
| **1** | 0 | 6 | 12 | 0 | 0 | 0 | 0 | 773 | 2 | 0 | 0 |
| **2** | 95 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 0 |
| **3** | 0 | 24 | 0 | 874 | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| **4** | 6 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 370 | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 0 | 0 |
| **6** | 3 | 0 | 0 | 0 | 0 | 197 | 0 | 0 | 0 | 0 | 0 |
| **7** | 0 | 0 | 904 | 0 | 0 | 0 | 0 | 0 | 1 | 14 | 9 |
| **8** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 | 521 | 0 | 9 |
| **9** | 1 | 444 | 4 | 0 | 0 | 0 | 0 | 0 | 15 | 1 | 0 |
| **10** | 0 | 0 | 0 | 0 | 29 | 0 | 3 | 0 | 0 | 0 | 0 |

TABLE II

Cross table of results from two rounds of K-mean clustering. The colored pattern indicates almost the same clusters were identified in two rounds.

| Network | Normalized Glove Spectral Cluster Labels | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Community Labels | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| null | 120 | 134 | 480 | 125 | 244 | 76 | 302 | 337 | 269 | 216 | 31 |
| 188 | 5 | 155 | 3 | 4 | 105 | 37 | 15 | 9 | 5 | 5 | 0 |
| 613 | 0 | 11 | 57 | 5 | 71 | 0 | 1 | 7 | 6 | 22 | 3 |
| 775 | 2 | 107 | 9 | 6 | 193 | 8 | 2 | 2 | 1 | 21 | 0 |
| 1227 | 8 | 21 | 348 | 5 | 36 | 1 | 5 | 19 | 71 | 12 | 6 |
| 1242 | 0 | 32 | 4 | 2 | 16 | 8 | 1 | 7 | 5 | 3 | 0 |
| 1340 | 3 | 92 | 53 | 7 | 47 | 17 | 35 | 42 | 33 | 14 | 6 |
| 1351 | 6 | 25 | 3 | 29 | 19 | 153 | 11 | 12 | 2 | 12 | 6 |
| 1459 | 72 | 1 | 0 | 1 | 4 | 15 | 1 | 0 | 0 | 0 | 0 |
| 1709 | 0 | 18 | 24 | 2 | 12 | 0 | 0 | 0 | 7 | 12 | 0 |
| 1739 | 0 | 20 | 4 | 4 | 24 | 10 | 19 | 26 | 8 | 22 | 2 |

TABLE III

Cross table of spectral clustering results and network communities.

| Network | K-Mean Cluster Labels | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Community Labels | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| null | 404 | 71 | 247 | 108 | 13 | 315 | 329 | 205 | 169 | 455 | 18 |
| 188 | 7 | 0 | 3 | 31 | 0 | 6 | 62 | 227 | 2 | 4 | 1 |
| 613 | 5 | 2 | 28 | 1 | 2 | 1 | 67 | 26 | 3 | 48 | 0 |
| 775 | 3 | 2 | 16 | 20 | 0 | 2 | 128 | 174 | 0 | 6 | 0 |
| 1227 | 24 | 2 | 105 | 4 | 1 | 5 | 64 | 21 | 7 | 298 | 1 |
| 1242 | 5 | 0 | 5 | 10 | 0 | 3 | 14 | 35 | 1 | 4 | 1 |
| 1340 | 40 | 2 | 37 | 18 | 2 | 24 | 71 | 106 | 5 | 44 | 0 |
| 1351 | 5 | 12 | 4 | 140 | 1 | 27 | 11 | 61 | 5 | 2 | 10 |
| 1459 | 2 | 11 | 2 | 56 | 0 | 0 | 3 | 19 | 1 | 0 | 0 |
| 1709 | 3 | 1 | 15 | 0 | 0 | 0 | 17 | 10 | 1 | 28 | 0 |
| 1739 | 31 | 2 | 2 | 11 | 0 | 25 | 18 | 39 | 8 | 3 | 0 |

TABLE IV

Cross table of k-means clustering result and network communities.

| K-Mean | Spectral Cluster Labels | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Cluster Labels | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 0 | 2 | 3 | 0 | 22 | 5 | 2 | 49 | 215 | 122 | 108 | 1 |
| 1 | 57 | 1 | 0 | 15 | 0 | 8 | 1 | 0 | 2 | 3 | 18 |
| 2 | 27 | 24 | 119 | 38 | 4 | 1 | 3 | 31 | 158 | 52 | 7 |
| 3 | 84 | 54 | 0 | 15 | 0 | 212 | 15 | 2 | 2 | 12 | 3 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 |
| 5 | 0 | 16 | 0 | 44 | 4 | 10 | 236 | 86 | 1 | 10 | 1 |
| 6 | 4 | 68 | 46 | 1 | 490 | 6 | 12 | 64 | 40 | 53 | 0 |
| 7 | 26 | 446 | 0 | 11 | 237 | 85 | 53 | 19 | 10 | 36 | 0 |
| 8 | 2 | 4 | 0 | 23 | 0 | 1 | 22 | 26 | 68 | 54 | 2 |
| 9 | 4 | 0 | 820 | 1 | 31 | 0 | 1 | 18 | 4 | 11 | 2 |
| 10 | **10** | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

TABLE V

Cross table of spectral cluster and K-mean cluster results.

## IV. CONCLUSIONS

In this manuscript, we have described and applied the popular clustering technique, the spectral clustering algorithm. We have derived the formulation for spectral clustering and demonstrate how it can be applied efficiently to high-dimensional data. We indicate how spectral clustering can be computed with the eigenvalue corresponding to the Laplacian matrix's second smallest eigenvalue. We also have discussed the drawbacks and improvements of spectral clustering.

In our experiments, we use synthetic data to demonstrate spectral clustering is a good model for general clustering purposes and makes no assumption on clusters' shapes. We also illustrate that spectral clustering could be a good tool for clustering data points using content-based features. We show how the results of spectral clustering could provide us valuable insights into the Twitter data set. In addition to demonstrating the behavior of spectral clustering on this data set, we compared it with DBSCAN and K-means' results, which are popular clustering methods.

## REFERENCES

[1] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. 2008.

[2] M. Emre Celebi, Hassan A. Kingravi, and Patricio A. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *CoRR*, abs/1209.1960, 2012.

[3] F.R.K. Chung, F.C. Graham, CBMS Conference on Recent Advances in Spectral Graph Theory, Conference Board of the Mathematical Sciences (U.S.). Conference on Recent Advances in Spectral Graph Theory, Calif.. Conference on Recent Advances in Spectral Graph Theory. 1994, Fresno, Conference on recent advances in spectral graph theory held at California state university 1994 ; Fresno, National Science Foundation (U.S.), American Mathematical Society, and Conference Board of the Mathematical Sciences. *Spectral Graph Theory*. CBMS Regional Conference Series. Conference Board of the mathematical sciences, 1997.

[4] Chris Ding, Xiaofeng He, and Horst D. Simon. *On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering*, pages 606–610.

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[6] L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[8] Ulrike Luxburg, Olivier Bousquet, and Mikhail Belkin. Limits of spectral clustering. volume 17, 01 2004.

[9] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[10] Fatma Ozge Ozkok. A new approach to determine eps parameter of dbscan algorithm. *International Journal of Intelligent Systems and Applications in Engineering*, 4(5):247–251, 2017.

[11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[12] Manoel Ribeiro. Hateful users on twitter detecting hate speech with context, 2017.

[13] Erich Schubert, Sibylle Hess, and Katharina Morik. Lwda 2018 lernen, wissen, daten, analysen 2018.

[14] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[15] Aarti Singh. Spectral clustering. 2010.

[16] Robert L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

[17] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. volume 17, 01 2004.