Spectral Clustering and Its Applications

Yueshuwei Wu, Victoria Li, Chenxi Jiang

University of California, Los Angeles

March 8, 2021

Introduction

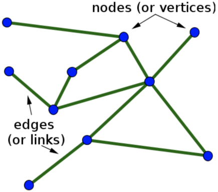
In multivariate statistics and the clustering of data, spectral clustering techniques make use of the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions.

Motivation

Our work is motivated by the following facts:

1. Due to the large size of the social network, it is unrealistic to rely on a sub-graph to detect the similarity based communities.
2. Online communities should share common characteristics in terms of their features and thus may form clusters in their features space.
3. Want to experiment with machine learning models to see if similar or associated results of community structures could be obtained.

Undirected Graph



$$G(V,E)$$

The weighted adjacency matrix of the graph is the matrix

$$W = (w_{i,j})_{i,j=1...n}$$

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

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

Undirected Graph

Definition

The Degree matrix is defined as

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

For two not necessarily disjoint sets $A, B \subset V$ we define

$$W(A, B) := \sum_{i \in A, j \in B} w_{i,j}$$

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

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

Different Similarity Graph

3 popular constructions

1. ε-neighborhood
2. k-nearest neighborhood
3. fully connected

Gaussian Similarity Function

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

Graph Laplacians

Definition

The unnormalized graph Laplacian matrix is defined as

$$L = D - W$$

Properties of L

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

$$f' L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{i,j} (f_i - f_j)^2$$

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 $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$

Graph Cut

Definition

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 $\overline{S}$.

Rcut and *Ncut*

Better cuts

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

$$vol(S) = \sum_{i \in S} \sum_{j \in V} w_{ij}.$$

Our ultimate goal is to minimize $Rcut$ or $Ncut$. However, this is **NP-hard**!

Relaxation of Ratio Cut

We define a simple step function $f_S : V \to \mathbb{R}$ to be

$$f_S(i) := \begin{cases} \sqrt{\frac{|\overline{S}|}{|V||S|}}, i \in S \\ -\sqrt{\frac{S}{|V||\overline{S}|}}, i \in \overline{S} \end{cases} \tag{1}$$

Proposition 5

For all f_S, we have $f_S^T L f_S = Rcut(S)$

Proposition 6

For all $S \subseteq V$ and f_S, we have $||f_S|| = 1, f_S^T \mathbf{1} = 0$

Proposition 7

The minimizer of the relaxation is v_2, which is the eigenvector corresponding to the second smallest eigenvalue λ_2 of the Laplacian L.

Relaxation of Normal Cut

Similarly, we define a simple step function $f_S : V \to \mathbb{R}$ to be

$$f_S(i) := \begin{cases} \sqrt{\frac{|vol(\overline{S})|}{|vol(V)||vol(S)|}}, i \in S \\ -\sqrt{\frac{vol(S)}{|vol(V)||vol(\overline{S})|}}, i \in \overline{S} \end{cases} \tag{2}$$

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

General Algorithm

Require: $\{x_n\}_{n=1}^{N} \subset \mathbb{R}^D$

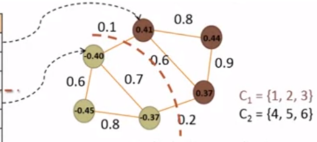
1. Construct a NxN similarity matrix using any suitable method (i.e. Euclidean distance/similarity between points). Treat the similarity matrix as a matrix representation of a graph G.
2. Pre-processing: build the Laplacian matrix L of the graph G.
3. Decomposition: Minimize the graph cut using the second smallest eigenvalue of L. 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.

	1	2	3	4	5	6
1	1.5	-0.8	-0.6	-0.1	0	0
2	-0.8	1.7	-0.9	0	0	0
3	-0.6	-0.9	1.7	0	0	-0.2
4	-0.1	0	0	1.4	-0.6	-0.7
5	0	0	0	-0.6	1.4	-0.8
6	0	0	-0.2	-0.7	-0.8	1.7

Laplacian of **G**

	$\mathbf{v}_2$
1	0.41
2	0.44
3	0.37
4	-0.40
5	-0.45
6	-0.37

Second eigenvector of L_G



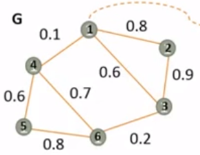
Nodes in **G** mapped onto $\mathbf{v}_2$.
Bipartition of G based $\mathbf{v}_2$

Common algorithms

1. Recursive bi-partitioning
   1. Recursively apply the bi-partitioning algorithm in a hierarchical divisive manner.
   2. Disadvantage: Inefficient, unstable.
2. Use multiple eigenvectors of the normalized Laplacian
   1. Other eigenvectors correspond to the smallest eigenvalues also works!
   2. By using each node's corresponding component in these eigenvectors as their features, we can cluster these nodes through **k-means**.
   3. This is a preferable approach in recent practices.

	1	2	3	4	5	6
1	1.0	-0.5	-0.4	-0.1	0	0
2	-0.5	1.0	-0.5	0	0	0
3	-0.4	-0.5	1.0	0	0	-0.1
4	-0.1	0	0	1.0	-0.4	-0.5
5	0	0	0	-0.4	1.0	-0.5
6	0	0	-0.1	-0.5	-0.5	1.0

$L_{norm}(G)$



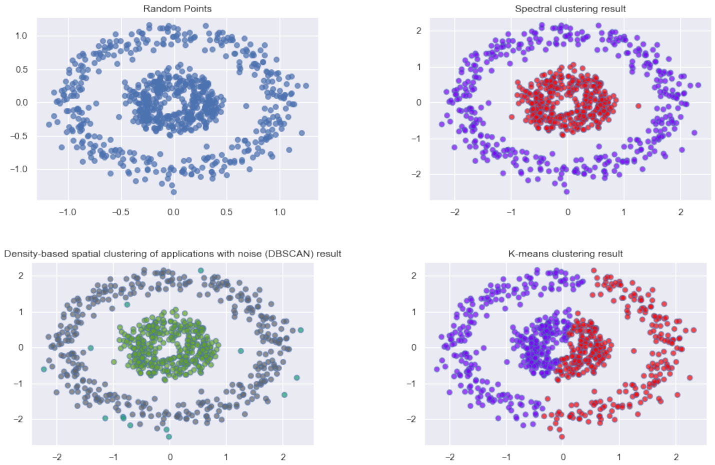
	v_1	v_2	v_3
1	$v_1(1)$	$v_2(1)$	$v_3(1)$
2	$v_1(2)$	$v_2(2)$	$v_3(2)$
3	$v_1(3)$	$v_2(3)$	$v_3(3)$
4	$v_1(4)$	$v_2(4)$	$v_3(4)$
5	$v_1(5)$	$v_2(5)$	$v_3(5)$
6	$v_1(6)$	$v_2(6)$	$v_3(6)$

U for $k = 3$

Relationship with Density-based Spatial Clustering (DBSCAN)

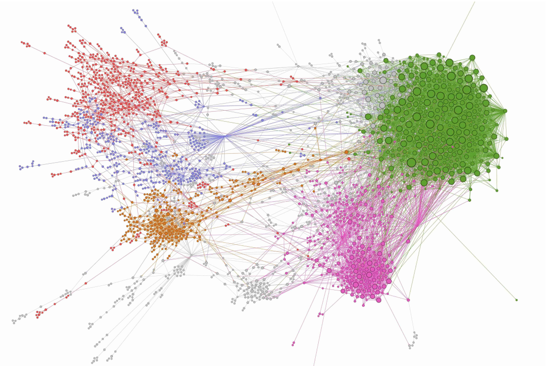
1. For theoretical interest, **DBSCAN can be viewed as a special case of spectral clustering** but one which allows more efficient algorithms (worst case $O(n^2)$) than standard spectral clustering implementations (usually $O(n^3)$).
2. It is worth to mention that tuning the parameters of DBSCAN is also complex.

Synthetic Data



Spectral clustering makes no assumption on the shape of clusters.

Twitter Data Application - Background



1. We discovered 10 communities in previous studies based on this retweet network structure.

Objectives

To examine if the spectral clustering model can produce a cluster structure similar to or associate with the communities found in previous network studies using content-exclusive features.

Assumption & Feature Selection

1. 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.
2. Only the contents posted by the user will significantly affect its distance with other users.
3. Use the GloVe algorithm for obtaining vector representations for words of a user's tweets. We will use the 300-dimensional vector as our features.

Results-Spectral Clustering and DBSCAN

1. DBSCAN found one giant cluster.
2. Spectral cluster found one giant cluster.
3. Spectral cluster with normalized feature matrix successfully found 11 clusters but seem to present a pattern of disorder and cut some clusters into two, producing clusters of similar sizes.
4. Users are hardly distinguishable from others just by content-exclusive features.

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

Results - K-means

1. We found consistent clusters using K-means.
2. The data is spherical and the data points can be separated by some convex boundaries.

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.

Results - Independence of clustering results

1. Applying the Chi-Square test of independence to check independence of clustering results.
2. Discovered the association between community labels, spectral clustering results and K-means clustering results are statistically significant ($p - value < 0.5$).

Network Community Labels	Normalized Glove Spectral Cluster 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 Community Labels	K-Mean Cluster 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 Cluster Labels	Spectral 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.

Conclusions and Thoughts

1. Spectral clustering is useful for performing dimensionality reduction before clustering in fewer dimensions.
2. Spectral clustering is robust against the shape of clusters (unlike K-means).
3. Even the bad results can still provided useful insight about data.
4. "All models are wrong, but some are helpful."

The End

We would like to express our very great appreciation to Dr. Haddock for her valuable and constructive suggestions during the planning and development of this project. Her willingness to give her time so generously has been very much appreciated. We've learned a lot and had fun in this class.