CLUSTER RANKING USING SPECTRAL CLUSTERING AND EIGENVECTOR CENTRALITY

Keerthiraj Nagaraj
Department of Electrical and Computer Engineering, University of Florida
University of Florida, Gainesville, Florida 32611-6200
k.nagaraj@ufl.edu

***Abstract*—In the recent past, network sciences has gained prominent position in data analysis. One of the main reasons being the correctness in representation of various networks like social networks, transportation networks, communication networks using concepts of network sciences. Most of the networks operate in hierarchical structure where they are divided into different communities based on similarity between different network nodes. Analyzing clusters is the first step to understand the behavior of the whole network. Identifying the clusters with utmost importance to the network plays a major role as it has applications such as understanding the vulnerability of transportation networks, resource optimization among communication networks. In this paper, the implementation of one of the popular clustering algorithms known as Spectral clustering which can be used to divide a given network into various clusters is discussed. Finally, one of the centrality measures known as Eigenvector centrality is calculated for all network nodes. These values are utilized to rank the clusters identified using spectral clustering.**

***Index Terms*—Spectral Clustering, Eigenvector Centrality, Cluster ranking.**

I. INTRODUCTION

A network is a collection of sample points which represents different kinds of data. In our daily life we keep encountering various networks such as communication networks, transportation networks and biological networks. Networks are used to show the interconnection of different sample points.

Graph theory is a branch of mathematics which deals with analysis of data by representing it in the form of graphs. Graph is a collection of nodes, which may or may not be connected by edges based on the features of nodes. Graph theory is one of the important tools to analyze networks as the structure of various units of networks like wireless networks, communication networks and transportation networks can be easily represented by nodes and connection between various units can be represented by edges.

Graphs can be categorized based on different features but for our work we will discuss about the classification of graphs based on edge weights. Graphs can be broadly classified into weighted and unweighted graphs. Unweighted graphs have unit weights on the edges whereas the weighted graphs have weights on edges based on the similarity between the end nodes. Weighted graphs has an advantage over unweighted graphs as they can be used to represent both connectivity between different nodes and the strength of connections but unweighted graphs can be used to represent connectivity only.

Spectral clustering is one among the rich family of clustering algorithms. This algorithm uses spectral information of the network to divide it into various clusters. Spectral clustering technique uses the spectral properties i.e. Eigenvalues and Eigenvectors of Graph laplacian matrix and K-means clustering algorithm in the end to form proper clusters of a given network. In this work, we are employing Spectral clustering as it is easy to understand and implement. Spectral algorithm usually provides better results compared to traditional clustering techniques. [1]

In a network, usually not all nodes will have same importance. In networks, the more the number of connections a node has, more important it is for the network than the node with fewer number of connections. It is also obvious that if an important node is connected to many other important nodes then it has more influence in the network than other important nodes with fewer connections. There is a need to identify the important set of nodes as this helps in understanding the vulnerability of network and there are many techniques to achieve the same and one of them is centrality measure. In our work, we will be employing the popular Eigenvector centrality measure to rank the clusters which are obtained using spectral clustering. [4], [5]

The remaining paper is organized as the following: Section 2 depicts the problem in hand and Section 3 explains the different techniques used to solve the problem formulated. In section 4, the method to generate different dataset and the results obtained from the proposed technique are discussed. The paper is concluded in section 5, which is followed by references.

II. RELATED WORK

In the literature many papers have been published which discusses about clustering in networks, community detection, and network division etc using classical clustering techniques and only few have used spectral clustering approach which is easy and efficient technique to cluster the give network. Paper [1] gives an overview of spectral clustering technique and also discusses about different variants of spectral clustering which is helpful for easy understanding of the approach. Paper [2]

discusses about the variant of normal spectral clustering which is used in our work. Paper [3] deals about the application of spectral clustering in detection of communities in graphs.

Papers [4], [5] discusses about the important properties of eigenvector centrality, and are helpful for understanding of the same. Papers [6], [7] deals about the creation of benchmark graphs for community detection algorithms. Paper [8] discusses about the cluster ranking approach but uses a more computationally complex procedure compared to eigenvector centrality for assigning ranks to clusters. In our work, we use a simple and efficient technique to solve the cluster ranking problem in networks.

III. PROBLEM FORMULATION

Consider a network such that $x_1, x_2, x_3....x_n$ represents the n nodes of the network. Let there be k clusters in the network such that $c_1, c_2, c_3....c_k$ represents the different clusters.

The importance of a node in network is represented by its Eigenvector centrality. Let the eigen vector centrality of each node xi be denoted by e_i, such that $e_1, e_2, e_3...e_n$ represents the eigenvector centrality of all the n nodes of the network. We will assign the mean of eigenvector centrality of all the nodes that belongs to the particular cluster as the effective eigenvector centrality measure for each cluster.

$$m_j = \frac{\sum_{e_a \epsilon C_j} e_a}{|C_j|} \tag{1}$$

Where, m_j represents the effective eigenvector centrality of each cluster j, $\mid C_j \mid$ represents the number of nodes that belong to cluster j, e_a represents the eigenvector centrality of node a which belongs cluster j.

Using the values of m_j, the different clusters of the network can be ranked, Higher the value of m_j for a cluster, higher is the rank of that particular cluster. Clusters $c_1, c_2, c_3....c_k$ are found using Spectral clustering technique and values of $e_1, e_2, e_3...e_n$ are found using Eigenvector centrality measure. Mathematical explanation of spectral clustering and eigenvector centrality are discussed in the following section.

IV. METHODOLOGY

A. Spectral Clustering

Spectral clustering can be implemented by using standard linear algebra techniques. It is better to first understand the meaning following terms rather than directly getting into the algorithm. Graph laplacian matrix is defined as the difference of weighted adjacency matrix and degree diagonal matrix.

$$L = D - A \tag{2}$$

Where, L represents Graph laplacian matrix, D represent degree diagonal matrix which is a diagonal matrix whose diagonal elements are degree of each vertices and A represents the weighted adjacency matrix. There are other variants [1], [2] of L and the following expressions show how to calculate them,

$$L_{sym} = D^{-1/2} A D^{1/2} \tag{3}$$

$$L_{rw} = D^{-1/2} L \tag{4}$$

Where L_{sym} and L_{rw} represents normalized graph laplacians, L represents unnormalized graph laplacian, D and A represents degree diagonal matrix and weighted adjacency matrix respectively.

In our work, we will be using L_{sym} which is one of the normalized graph laplacians as it is proved better results when compared to unnormalized graph laplacian matrix. [1]

One of the drawbacks of spectral clustering is that it doesnt converge to optimal number of clusters on its own and it needs the number of clusters as the input. In this work, we use Eigengap heuristic to find the optimal number of clusters for each dataset generated and tested. We have to first evaluate the set of all eignevalues of graph laplacian matrix and arrange them in ascending order and find the difference between consecutive sorted eigenvalues and form a new set of these values, if k^{th} item of this set is maximum then decide k as the optimal number of clusters. [1]

$$k = argmax_i(l_{i+1} - l_i) \tag{5}$$

Where l_i represents i^{th} element of sorted eigenvalue set of graph laplacian matrix.

The following algorithm was proposed by Ng et al in 2002, It is the latest form of spectral clustering algorithm and is proved to provide better results than un-normalized spectral clustering or normalized spectral clustering proposed by Shi and malik in 2000.

Algorithm 1: To find k-clusters in the given graph [2]

Input: End nodes and weights for all the edges of a network with n nodes

1) Find degree of each node and form a (nxn) diagonal matrix D whose ith principal diagonal element represents degree of i^{th} node.
2) Form adjacency matrix A using the edge weights such that:$a_{ij} = w_{ij}$ - only if an edge exists between node i and node j
   0 - otherwise: Where, a_{ij} represents $(i,j)^{th}$ element of matrix A and w_{ij} represent the edge weight between node i and node j.
3) Find normalized graph laplacian L using the below equation 3,
4) Decide the optimal number of clusters k using Eigengap heuristic.
5) Find the first k eigenvectors of L and form a matrix E using eigenvectors as columns.
6) Normalize rows of E to make the norm of each row equal to 1.
7) Apply K-means algorithm to normalized E to find the k clusters of the original data.

Output: k clusters for the data which was given as input.

B. Eigenvector centrality

Eigenvector centrality uses the spectral information of graph laplacian matrix to find the importance of each node in the network. It assigns each node with a particular score and higher the score is higher is the importance of that particular node. Eigenvector centrality measure for each node is obtained by the following algorithm.

Algorithm 2: To find Eigenvector centrality for all the nodes [4]

Input: Graph laplcain matrix, Degree matrix and adjacency matrix

1) Find the maximum of eigenvalues $lmax$ of graph laplacian matrix L.
2) Find the degrees of all the nodes and form an $(nx1)$ column matrix C using node degrees
3) Find the nxn adjacency matrix A for the given dataset.
4) Evaluate score of each node using the following expression,

$$E = \frac{A * C}{l_{max}} \tag{6}$$

Where E is a column matrix whose i^{th} element represents the eigenvector score for i^{th} node.

Output: Eigenvector scores of each node.

C. Ranking of clusters

We have the k clusters of data obtained from spectral clustering and eigenvector scores all the nodes. The rank of each clusters is evaluated using the following algorithm.

Algorithm 3: Ranking of clusters

Input: k clusters obtained using spectral clustering, column matrix E which represents eigenvector score of each node.

1) Assign an ID for each node based on the cluster to which it belongs to i.e ID of the node will be i if it belongs to cluster i.
2) Calculate the mean m of eigenvector score of all the nodes with same IDs and assign it to each cluster.
3) Assign ranks to the k clusters based on the value m of each cluster i.e. higher the value of m better is the rank.

Output: Rank of each k clusters.

Algorithms $1, 2\&3$ are implemented in the respective order to solve the problem formulated. Techniques for generating datasets and results obtained for various datasets are mentioned in the following section.

V. RESULTS AND DISCUSSIONS

A. Dataset generation

We have used the programs and algorithms designed by Andrea Lancichinetti and Santo Fortunato in our work to generate different datasets of graphs to test the proposed technique. We have used packages 2, using which undirected weighed graphs with possibly overlapping communities can be generated. The method of compiling and running the code is mentioned in a file called ReadMe in package 2. The table 1 lists the important parameters considered while creating the datasets. [6], [7]

Parameter flag	Parameter value
N	Number of nodes in the graph
k	Average degree of nodes
maxk	Maximum degree of nodes
minC	Minimum for community sizes
maxC	Maximum for community sizes
C	Clustering coefficient

TABLE I: Parameters considered for Dataset generation

We generated 3 datasets with different values for parameter flags and have used them for our work. The values of parameters used for 3 datasets are shown in the table 2.

Parameter flag	Dataset1	Dataset2	Dataset3
N	500	500	500
k	15	10	10
maxk	50	15	20
minC	20	10	30
maxC	50	30	60
C	0.5	0.1	0.6

TABLE II: Values of parameter flags for generated datasets

B. Observations for different datasets

We implemented the algorithms proposed using MATLAB software and used the generated 3 datasets to test the technique proposed. We have discussed in earlier sections that we need to evaluate the optimal number of clusters of a graph has to be found using Eigengap heuristic, we have mentioned the optimal number of clusters for each dataset in table 3.

Dataset number	k-clusters
1	14
2	32
3	27

TABLE III: Number of optimal clusters found using Eigengap heuristic for different datasets

The following figures show the edges and nodes of networks generated using dataset 1, 2 and 3 respectively. These networks were formed using the NodeXL, which is an open-source template for graphing network data.

Figures 1, 2 and 3 represents the edges and nodes for datasets 1, 2 and 3 respectively where blue dots indicates network nodes and grey lines indicate network edges or network links.

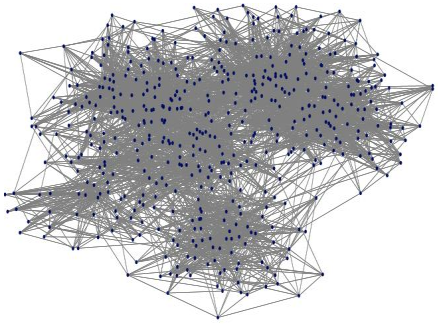

Fig. 1: Edges and nodes of network for dataset 1

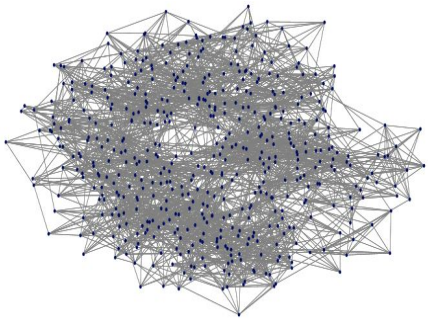

Fig. 2: Edges and nodes of network for dataset 2

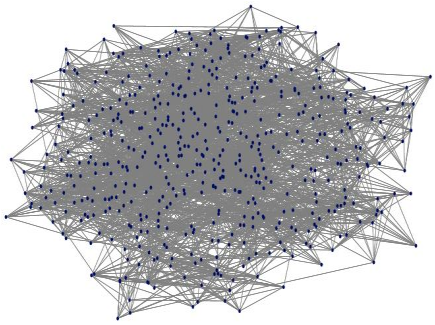

Fig. 3: Edges and nodes of network for dataset 3

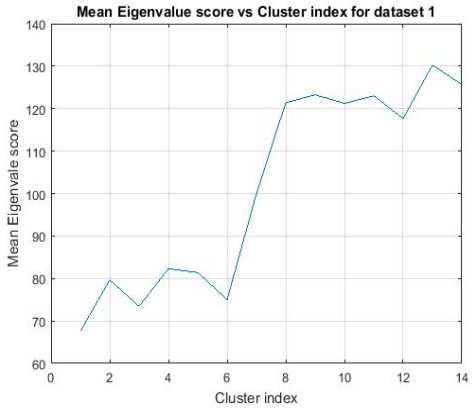

Fig. 4: Variation of Mean Eigenvalue score for different clusters in dataset 1

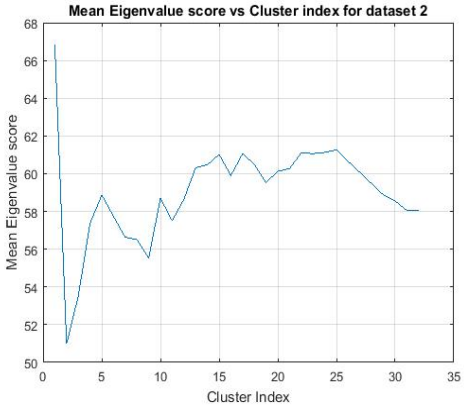

Fig. 5: Variation of Mean Eigenvalue score for different clusters in dataset 2

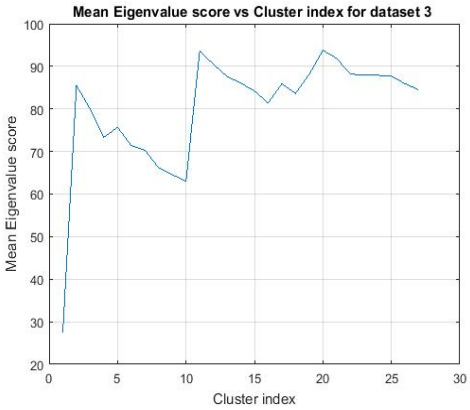

Fig. 6: Variation of Mean Eigenvalue score for different clusters in dataset 3

Figures 4, 5 and 6 shows the variation of Mean Eigenvalue scores for different clusters of networks for dataset1, dataset2 and dataset3 respectively. The variation of Mean Eigenvalue scores for different clusters in each network show that, the importance of each cluster is different from others. Hence, the need to find the rank of different clusters of the network so that the higher rank networks can get better resources, which in turn improves the efficiency of the whole network.

VI. SUMMARY AND CONCLUSIONS

In this work, we have worked on identifying and ranking the clusters of any given network. We used spectral clustering techniques to find the different clusters of a given network. Eigengap heuristic was used to decide the optimal number of clusters for different datasets. We evaluated the eigenvector centrality measure for all the nodes and used the algorithm proposed to find the Mean eigenvalue score for each cluster and this parameter can be used to rank the clusters. Mean eigenvalue score represents the importance of a cluster and shows connectivity of nodes which belong to that cluster. Higher the Mean eigenvalue score of a cluster is, better is the connectivity and more important nodes of the whole network are interconnected to nodes belonging to that particular cluster.

The proposed technique is useful when we have to find the vulnerable groups of a network, which can be helpful when the resources available for the network are limited. These limited resources can be distributed to the clusters of nodes which are more important so that the network efficiency can be increased. We wish to take this work forward and implement proposed technique to find important clusters in wireless networks and check how the overall network efficiency gets affected.

VII. ACKNOWLEDGMENT

I would like to thank Dr.Dapeng Oliver Wu (Professor at the Department of Electrical and Computer Engineering, University of Florida), for his valuable inputs which helped me to successfully complete this project.

REFERENCES

[1] Von Luxburg, U., *A tutorial on spectral clustering*, Statistics and computing, 17(4), 395-416 (2007)

[2] Ng, A. Y., Jordan, M. I., & Weiss, Y., *On spectral clustering: Analysis and an algorithm*, Advances in neural information processing systems, 2, 849-856 (2002)

[3] White, S., & Smyth, P., *A Spectral Clustering Approach To Finding Communities in Graph*, In SDM (Vol. 5, pp. 76-84) (2005)

[4] Ruhnau, B., *Eigenvector-centralitya node-centrality?*, Social networks, 22(4), 357-365 (2000)

[5] Bonacich, P., *Some unique properties of eigenvector centrality*, Social Networks, 29(4), 555-564 (2007)

[6] Lancichinetti, A., Fortunato, S., & Radicchi, F., *Benchmark graphs for testing community detection algorithms*, Physical review E, 78(4), 046110 (2008)

[7] Lancichinetti, A., & Fortunato, S., *Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities*, Physical Review E, 80(1), 016118 (2009)

[8] Bar-Yossef, Z., Guy, I., Lempel, R., Maarek, Y. S., & Soroka, V., *Cluster ranking with an application to mining mailbox networks*, Knowledge and Information Systems, 14(1), 101-139 (2008)