Binary Vertex Clustering using the Fiedler Vector

Student Number: 10219553

0.1 Abstract:

PURPOSE: Investigate & evaluate the effectiveness of the implementation of the Fiedler vector for the Binary Clustering of the vertices of a graph.

METHODS: The Adjacency, Degree, & Laplacian matrices were found from a sample edge list. The Fiedler vector was calculated using the eigenvectors & eigenvalues of the Laplacian matrix. This vector was used to partition the vertices into 2 distinct clusters.

RESULTS: The Fiedler vector is effective for the Binary Clustering of a graph. It can provide clusters where $> 90\%$ of all edges are contained within one cluster.

CONCLUSIONS: The algorithm implemented over the given dataset met the metrics evaluation criteria specified in the scientific question, with $> 91\%$ of edges falling within a single cluster. This method is highly effective for this use-case, and further extension of the algorithm using fuzzy clustering may yield more insightful results.

Word Count: 135

0.2 Introduction:

The objective of this study was to investigate the effectiveness of the usage of the Fiedler vector of a Laplacian matrix for the Binary Clustering of a graph from an edge list.

An edge list is a list describing the edges of a graph. It is structured as a list with 2 columns, which stand for vertices v_i & v_j. The Laplacian matrix is the result of $D - A$, where D is the degree matrix of the graph and A is the adjacency matrix of the graph [1]. It has values $>= 0$ along the diagonal and values of 0 or -1 in all other entries. Binary clustering is the partitioning of the dataset into two distinct clusters. It partitions the vertices of the graph, and can determine whether the graph is made up of 2 distinct components. The Fiedler vector of the graph is the eigenvector of the second smallest eigenvalue of the Laplacian matrix. It is transformed into entries of ± 1, with each indicating the Binary Cluster it belongs to [1].

The scientific question to be looked at is, given a graph, can the Fiedler vector provide us with a Binary Clustering where $> 90\%$ of all edges are contained within either of the clusters.

0.3 Methods:

The main algorithm used in this report has three main segments, the first of which is the translation from the edge list of a graph into an adjacency matrix, and subsequently a degree matrix. This report requires both of these matrices for a later calculation. The initial step is to load the edge list from a text document, and find the largest vertex value that has at least one edge. For the purposes of this report, this number will always be $n = 20$. The next steps are to initialize a blank adjacency matrix of $n * n$, and to find the number of entries in the edge list. This number is then used to fill in the edges of the graph into the adjacency matrix. This matrix is then multiplied by $\vec{1}_{20}$ to find the degree vector, which is subsequently applied to a diagonal matrix. The final step in this segment is to find the Laplacian Matrix, which can be calculated as $L = D - A$.

The next section involves binary clustering and the calculations that precede it. After acquiring the Laplacian matrix, it is necessary to calculate it's eigenvectors & eigenvalues. Once this is done, the Fiedler vector is assigned as the eigenvector of the second smallest eigenvalue. Next, the set12 vector is calculated as the following element-wise logical comparison over the entries of the Fiedler vector: $>= 0$. This new vector is a binary vector with all entries equal to either 0 or 1. It is then transformed such that all zero-values are represented by -1, resulting in 2 clusters of vertices.

The final segment of the algorithm involves cleaning up and presenting the data. The first part of this process is to create two new vectors with $n = 20$ columns, each representing one of the clusters. The indexes of the set12 vector correspond to the same vertex in the graph. Using this information, the indexes of each 1 or -1 in set12 is partitioned into either of the new vectors. The transposes of these vectors are then displayed, ensuring that they appear as rows of vertices. The second part of this process is to plot the results. This is done using the *plot271a1* function, which also relies on the *circlen* function. Both of these functions were provided in class by Professor Ellis. The resulting plot has two sections, the first of which is the graph represented as a blue rectilinear grid. On the bottom is the graph plotted according to the prior binary clustering computation.

The algorithm is initially tested against the provided *"testedge.txt"* & *"testsets.txt"* files, after which is applied to the unique *"21zp16.txt"* file. The initial test files are accompanied by the corresponding clustering table & graphs.

0.4 Results:

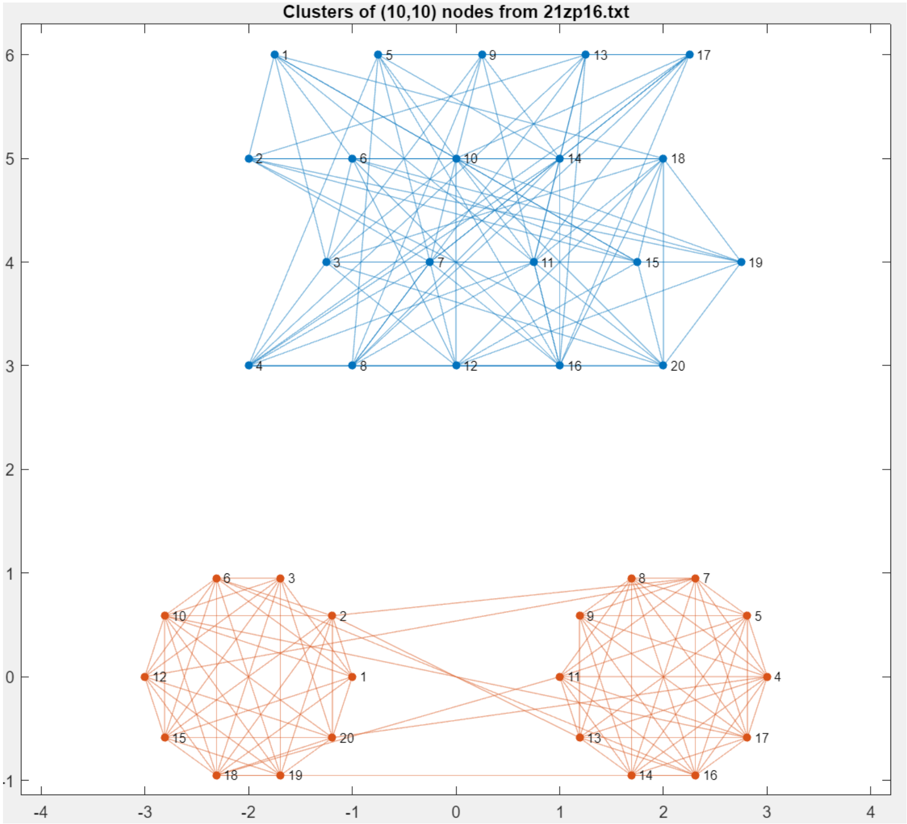

Figure 1: The above figure is partitioned into two representations of the same graph. Above, in blue, is a rectilinear grid. Below, in orange, is the graph plotted according to the binary clustering computation. Both representations are derived from the provided edge list.

Table 1: The two clusters of vertices on the graph, as dictated by the clustering vector & the edge list. Each row is a set of vertices corresponding to one of the clusters.

Sets	Vertices
Set 1	1 2 3 6 10 12 15 18 19 20
Set 2	4 5 7 8 9 11 13 14 16 17

0.5 Discussion:

The findings of this experiment are wholly contained within the figure and table in the results section. The rows of the table describe the binary clusters, and each entry is a vertex that belongs to this set. The blue graph in figure one provides a visual understanding of the provided edge list, while the orange graph in describes the relationships between the vertices in each cluster and those between the cluster.

Both clusters contain an equal number of vertices, and that there does not appear to be any obvious pattern to the vertices, other than the fact that they are mostly evenly laid out. The largest consecutive sequence of vertices is of length 3. It is important to note that there are 3 sequences of 3 consecutive integers. The first two occur in cluster 1, from vertices 1-3 & 18-20, while the third can be found near the middle of cluster 2, with vertices 7-9. This implies that there may be a distinct spatial relation between the clusters. For example, the graph could be thought of in space with vertices 1-3 on one extreme, 7-9 central, and 18-20 on another extreme, with noise in between. Additionally, the 2 sequences from cluster can also be considered as a single sequence from 18-20, 1-3, as if counting with mod 20.

There is an equal number of vertices in cluster 1 & cluster 2, a split of 10:10 from the original 20. This initially implies that the Fiedler Vector and Binary Clustering were fairly effective. This hypothesis is solidified when observing the clustering visually as a graph, as there is a distinct segment for each set. One of the main uses of binary clustering is to show the connectivity of a graph and whether it can be partitioned into 2 components. In this graph, while it cannot be completely separated into components, it is relatively close to doing so. In total there are 182 entries in the edge list, which translates to 91 edges in the graph. Of these 91 edges, only 8 are adjacent to vertices in both clusters (91.2%), while the rest are contained within a single set. This demonstrates that the scientific question that was previously outlined is true: the Fiedler vector can be used for Binary Clustering such that $> 90\%$ of all edges are contained within either of the clusters.

Vertices 7 & 18 share the largest number of edges adjacent to a vertex in the opposing set, which is 2. Together they account for 50% of the edges that join the two sets together. One interpretation of this is that these vertices are outliers, and have similar qualities to both clusters. However, these edges account for $<=$ 20% of the degree of their respective vertices, indicating that while there may be some relationship, they are still strongly related to their one of the clusters. This is a limitation of binary clustering, where only 2 distinct clusters can exist. In some real-world applications, only having 2 clusters may not have the depth to show any strong relationships. A possible solution would be to extend the algorithm to incorporate fuzzy clustering. Baraldi and Blonda [1] discuss pattern recognition with fuzzy clustering algorithms. Implementing these algorithms would allow for new patterns involving vertices 18 & 7 to be discovered. According to Ruspini, Bezdek, & Keller [2], these fuzzy clustering algorithms can be applied to a breadth of mathematical & statistical problems involving qualitative object description. Some potential applications involve time series, molecular simulations, and problems involving genetic material [2].

In conclusion, binary clustering using the Fiedler vector met the evaluation criteria proposed in the scientific question, and is an effective method to cluster the vertices of a graph. The algorithm achieved $> 91\%$, surpassing expectations of 90%, and further algorithmic exploration using fuzzy clustering may yield better results.

0.6 References:

1. Ellis RE. CISC 271 Class 3: Graphs - The Laplacian Matrix [Internet]. 2021 [2024 January 17]. Available from: https://research.cs.queensu.ca/home/cisc271/pdf/Class03.pdf

2. Ruspini EH, Bezdek JC, Keller JM. Fuzzy Clustering: A Historical Perspective. IEEE Comput Intell Mag. 2019 Feb;14(1):45-55. doi: 10.1109/MCI.2018.2881643.

3. Baraldi A, Blonda P. A survey of fuzzy clustering algorithms for pattern recognition. I. IEEE Trans Syst Man Cybern Part B (Cybernetics). 1999 Dec;29(6):778-85. doi: 10.1109/3477.809032.

4. Kondruk N. Clustering method based on fuzzy binary relation. East-Eur J Enterp Technol. 2017;2(4 (86)):10–16. doi: 10.15587/1729-4061.2017.94961.