Q1 Report

The problem involves the emails between individuals of an organization and we are required to find the communities within this organization by using the email communications.

Data Loading

A simple bit of Python code is used to read the data, split the data and load the data into a pandas dataframe.

Algorithm and Approach

For my algorithm I have used Spectral Clustering using the SpectralClustering package from sklearn.cluster. I have used a numpy array to build a graph where each column and row index correspond to a node in the graph. A for loop of the source,target email dataframe allows to fill the graph according to index. This graph is known as an Adjacency matrix and each row and column index represents a node and the value in each cell represents the edge between a node. The diagonal is zeros as the no emails are sent to themselves.

Spectral Clustering also creates a Degree matrix. It shows how many nodes are connected to any given node and is represented by the value in the diagonal.

After calculating the Adjacency Matrix and Degree Matrix, the Graph Laplacian is calculated which is simply the Degree Matrix subtracted by the Adjacency matrix.

The Graph Laplacian is broken down to its eigenvalues and eigen vectors using Singular Value Decomposition. The first nonzero eigenvalues is the spectral gap and it indicates to us how densely connected the graph is. The second value Is the Fielder value and its corresponding eigen vector is the Fielder vector. It indicates how the graph should be cut into 2 components. Values in the fielder vector indicate to us where in each cut the node belongs to.

To identify clusters, in the eigen values, we look to when the largest jump I eigen values occur. The index at which the largest jump occurs indicates how many clusters/communities are there. For example if there is a jump from eigen value index 8 to index 9, it indicates 8 possible clusters.

After finding the number of possible clusters, we use that as input to the number of clusters(n_clusters) we require to a KMeans clustering algorithm. Then we do clustering on the first n_clusters eigenvectors to find the clusters on the graph. Essentially the clustering is done on a low dimensional space.This is the basic principles of the algorithm.

Spectral clustering is useful when the measure of the center and spread of a cluster is not a suitable description of the complete cluster [1]. Spectral clustering has roots to graph theory, like the one we have in this question, where we want to find communities of nodes within the graph given the edges between the nodes.

In scikit learn the implementation of Spectral clustering, I have used affinity matrix to be ‘precomputed’, meaning that it will use the Adjacency matrix that I built as the affinity matrix. N_components is the number of eigen vectors to do KMeans clustering on. I have varied this value to identify which n_components gives the best NMI value. It was found to be 22.

Results are saved as Q1_communities.csv

[1] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html

[2] https://towardsdatascience.com/spectral-clustering-aba2640c0d5b