Maniell Workman

Social Network Analysis

Eigenvector Centrality

**Tools needed**

This implementation of network analysis was created with python 3.7. The packages imported were networkx, Matplotlib.pyplot, and csv are necessary for the operation of the simulation. The packages scipy, pandas, and sys are included for supplementary purposes such as debugging and improvements. IDE in which this was developed is Visual studio code. Other environments such as Jupyter, Python terminal, and any with Python 2 or 3 capabilities are sufficient.

**Abstract**

Social network analysis is a focus area of Network analysis dealing with interpersonal connections. In network analysis, the typical topography is a hypergraph structure comprised of points connected by edges. Attributes such as node degree, edge density, and edge weight are aggregated to reveal network structure both locally and overall. In social networks, nodes can be comprised of people, hashtags, discussion topics or any quantifiable point in the structure. Edges are connections between these nodes. These connections take form as data transmission, shared people between the groups, or a tangible affect the nodes have between each other. Community detection is a challenge of social network analysis. In K means clustering, a given number of nodes are chosen at random as centroids. These centroids are used as distance comparison points across the graph. The distance from a given point to the k centroids are compared and the centroid that is closest said point is said to belong to that cluster. This method of clustering is widely popular for both its simplicity and accuracy. With k means clustering community detection is made accurately and precisely.

**Implementation**

This form of community detection requires only one function. The first part of the main function builds the graph given the data in the CSV file. The call to k\_core function takes the graph and a given number of centroids as arguments. Applying the method edges to this call creates a group of edges based around the nodes closest to the centroids finally storing them in a list. The call to clear after these calls clears the graph as a blank slate. The new clustered graph revealing communities is then drawn in its place. In this new graph the clusters are clearly visible.