Networks
Social Networks of Doctors

Overview

The provided network has as its nodes doctors representing 85% of doctors in 4 cities. The edges represent who they socialised with or had discussions about medical practice with. The nodes have different attributes such as the city the doctor is from and there are 4 types of edges but this is not used. The network also has directed edges but these are converted to undirected.

Community detection

I applied the following community detection methods to the graph to find communities in the graph and compare it with the cities doctors are from.

- Newman's eigenvector method
- Edge betweenness
- Propagating labels
- Multi-level optimisation of modularity
- Spin-glass model and simulated annealing
- Random walks

Newman's eigenvector method was the best at identifying the communities overall having the highest minimum accuracy and the second highest average accuracy and only identifying one more community than there are number of cities. The network is plotted with the communities represented as vertex colours in fig 1.
Edge betweenness identified the most communities at 13 more than the 4 cities but had the highest mean accuracy. All methods identified more than 4 communities. Random Walks had the lowest mean accuracy out of all the methods. Every method had a 100% accuracy in identifying one community but not the same community for each method.

Giant components

A giant component of a graph of n verticies is a component of the network of size $O(n)$. So is a component that contains a high proportion of the verticies of a network. The network does have a giant component the same size as the network.
A Erdos-Renyi random graph is a random graph where two verticies are connected with probability p. So if we deleted edges randomly until the graph is likely to be very disconnected (there's no giant component) we find the number of edges we need to delete as follows with u the probability of a vertex not being connected to the giant component and $< k >$ as the mean degree. We need to solve the equation

$$u = e^{-<k>\cdot(1-u)}$$

If we define $x = 1 - u$ so the probability of not being connected to the GC we need to solve

$$x = 1 - e^{<k>\cdot x}$$

The solutions to this equation as $< k >$ the mean degree increases are

- For small $< k >$ the only solution is $x = 0$ (no giant component)
- At a critical $< k >$ a second solution appears
- for large $< k >$ there are two solutions

The critical value happens when the differentials of each side of the equation are equal.

$$\frac{d}{dx}\left(1 - e^{<k>\cdot x}\right)\Big|_{x=0} = 1$$
$$\implies < k >= 1$$

So theoretically if the network was Erdos-Renyi we would need to remove $\frac{1108 - 242}{1108} \backsim 0.7816$ as a fraction of the edges of the graph.
Removing edges at ranom and recording the size of the largest component over 100 iterations we get the graph in fig 1. We see that the size of the largest suddenly changes at around 80% of the edges removed which is close to the theoretical value if the graph was Erdos-Renyi random on average. The variation is likely because the graph isn't completely random and has some structure which is also shown by the community detection before.

Network with communities identified by Newman's

A plot of the network with the identified represented by vertex colours on the graph.

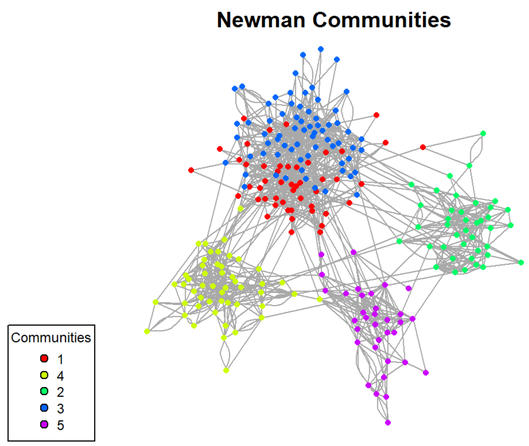

Figure: Communites identified by Newman's eigenvector method

Plot of the size of the largest component as you delete edges randomly

Graph showing the size of the largest component as you randomly delete edges over 100 iterations with the theoretical value if the network was a Erdos-Renyi random graph plotted as a red vertical line.

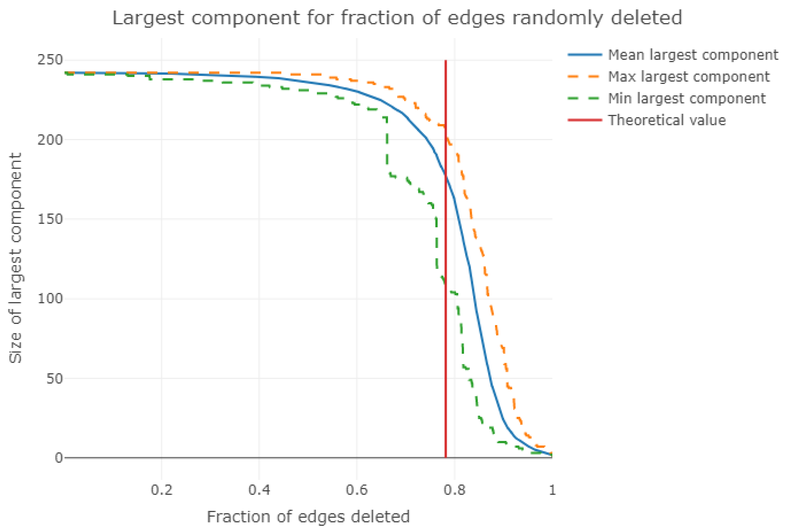

Figure: Plot of the largest component as tyou randomly remove edges.

Centrality

We use four centrality measures to identify the centrality of nodes in the network.

- Degree
- Betweenness
- Closeness
- Eigenvector

I also to compare them with an ensemble technique to see if it would improve performance.
Normalised the scores of each of the aforementioned centrality measures and took their mean to make a mean centrality.
After creating a centrality score for each vertex you put them in descending order. To disrupt the network as much as possible we want to delete the verticies with the highest centrality as these are the most important verticies in the network.
The most important verticies for each measure was different. Some did appear in more than 1 of the top 10 highest. To test the importance of these verticies we remove them and test to what extent they disconnect the network using the following measures.

- Size of the largest component
- Number of edges
- Number of components
- Cluster coefficient

If we look at the these we see that the largest component size is very similar for all of the measures. However for number of edges we see that betweenness does the best in figure 3. The more and faster they remove edges the more they disrupt the network.

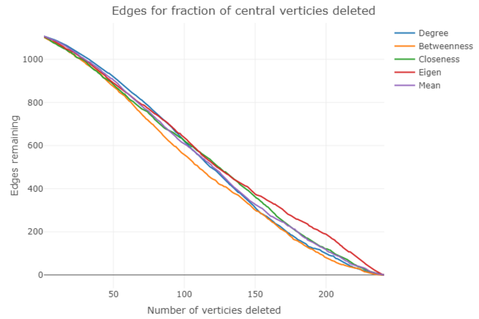

Figure: Edges of each graph as verticies are removed

The number of components also shows disruption as they increase and degree made the most components early on and betweenness the most overall. If we finally look at cluster coefficents the lower the cluster coefficent the more disrupted the network is as shown in figure 4. Betweenness and closeness performed the best here.

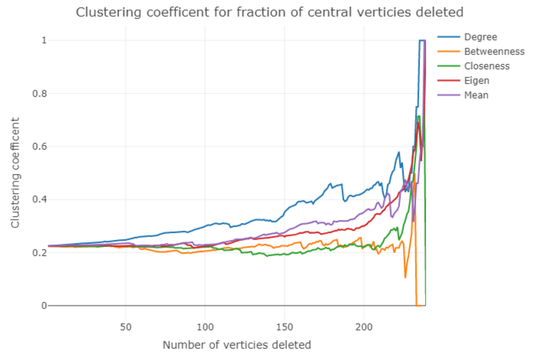

Figure: Cluster coefficient

Overall deleting by betweenness caused the most disruption the mean centrality method didn't offer better performance than the others but was never the worst performing.

Alexander John Pinches
CID:01201653 "The contents of this work and the associated code are my own unless otherwise stated"
Imperial College London