Task 12: Clustering

The data

The network, facebook_circle.edges, is a network of friends for a single facebook user. Note that the user has been removed and the resulting graph is not connected.

Part 1: Calculate the number of edges in the graph

Determine the number of edges in the graph. You should get an answer of 2519.

Part 2: Connected components

Implement a depth-first search algorithm to check for connectivity and then use it to find all connected components in the graph. (Two nodes v and u are in the same connected component if and only if there is a path between them.) You should find there are 5 connected components in the graph facebook_circle.edges.

Part 3: Edge betweenness

Implement edge betweenness. This is a minor variant of your code from Task 11. The pseudocode for is in Brandes (2008: section 3.5): http://www.sciencedirect.com/science/article/pii/S0378873307000731 Note: the last statement in the pseudocode can be ignored as this is only when considering node betweenness in tandem with edge betweenness which we are not doing here.

Part 4: Girvan-Newman

Implement the Girvan-Newman algorithm, which works by splitting a graph into relevately dense components. In the version we will use, the number of clusters to be returned is specified as a parameter. The pseudocode is as follows:

1. while number of connected subgraphs < specified number of clusters and the number of edges in graph > 0:
2. calculate edge betweenness for every edge in the graph
3. remove edge(s) with highest betweenness
4. recalculate connected components

In the case where there are ties for highest betweenness, you should remove all the highest betweenness edges. (This means that the number of connected components you end up with may be higher than the originally specified number. Girvan and Newman suggest two possible approaches: remove all tied edges or remove an edge at random.)

When comparing betweenness values (including determining ties for highest betweenness), use a tolerance of 1e-06 for a fuzzy comparison of doubles.

Reading

We are following the description of the algorithm in: http://www-personal.umich.edu/~mejn/papers/epjb.pdf.

Newman, Mark EJ. "Detecting community structure in networks." The European Physical Journal B-Condensed Matter and Complex Systems 38.2 (2004): 321-330.

The original paper is: https://arxiv.org/pdf/cond-mat/0308217.pdf

Newman, Mark EJ, and Michelle Girvan. "Finding and evaluating community structure in networks." Physical review E 69.2 (2004): 026113.

Starred Tick

In the implementation of Girvan-Newman for the tick, the number of clusters is specified in advance. However, this is a problem if we have no idea of the correct number of clusters. To take this into account, you can implement a 'goodness of clustering' called modularity which is described in Newman (2004, page 6). This is equivalent to calculating the complete dendrogram and then choosing the level with the best modularity (see the slides).

Each time we create a cluster in the network, the modularity of the clustering is calculated and we store the components found at that stage. We run the Girvan-Newman code to completion (i.e. to the point where there are no edges and every node is separate), and then take the set of components associated with the highest modularity.