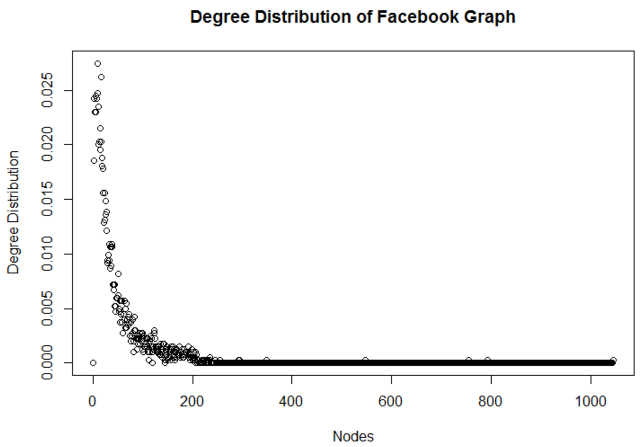
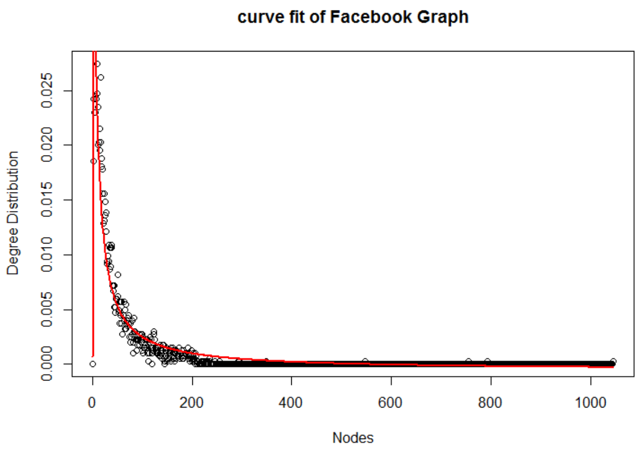
EE232e Project 1
Haixi Ni 404383125
Yu Zhang 904512554

Problem 1

The network is connected. The diameter of the network is 8. Degree distribution of Facebook graph is shown in the following figures. We use the polynomial function to fit a curve using on it. After several trials on curve fit functions, (1/x)^3 is chosen to obtain the minimum mean squared error, which is shown in Figure 2. The total mean squared error is 8.427251e-07. Average degree of the graph is 43.69.





Problem 2

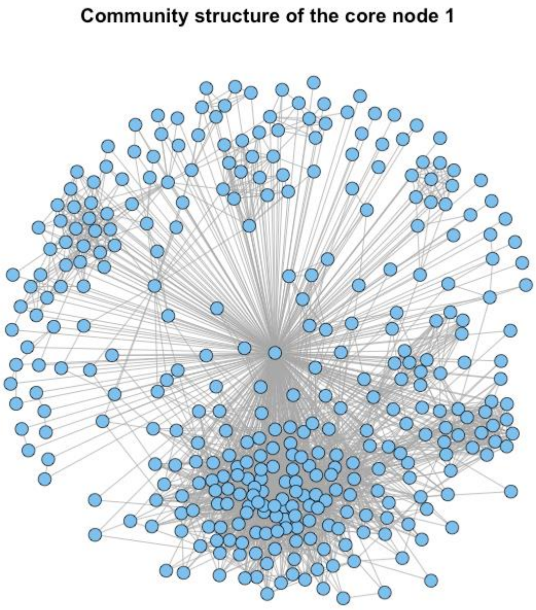
Take the first node in the graph. Create a graph that consists of node 1 and its neighbors and the edges that have both ends within this set of nodes. The code is as follows.

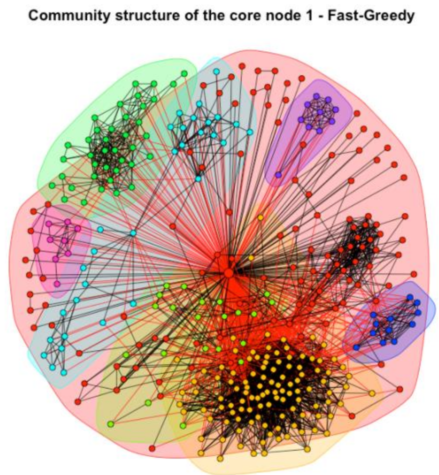
```
subGraphNodes <- neighborhood(g, 1, nodes=V(g)[1])
subGraphNodes <- subGraphNodes[[1]]
nonSubGraphNodes <- which( !( (1:vcount(g)) %in% subGraphNodes) )
subGraph <- delete.vertices(g , nonSubGraphNodes)
```

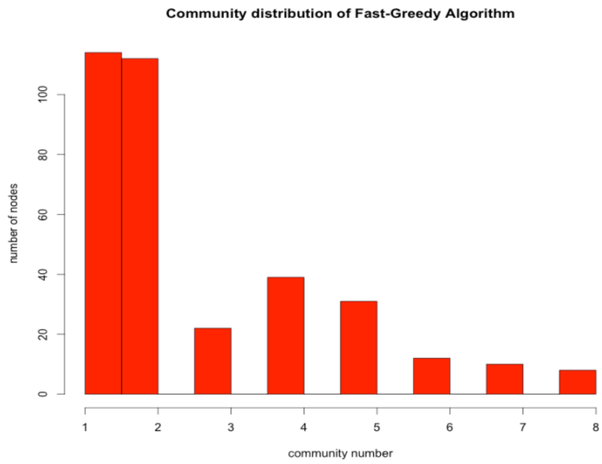
The graph has 2866 edges and 348 nodes.

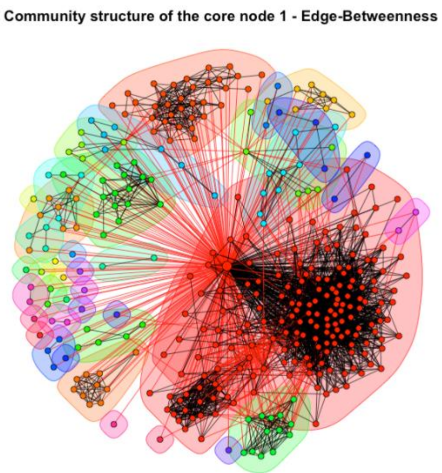
Problem 3

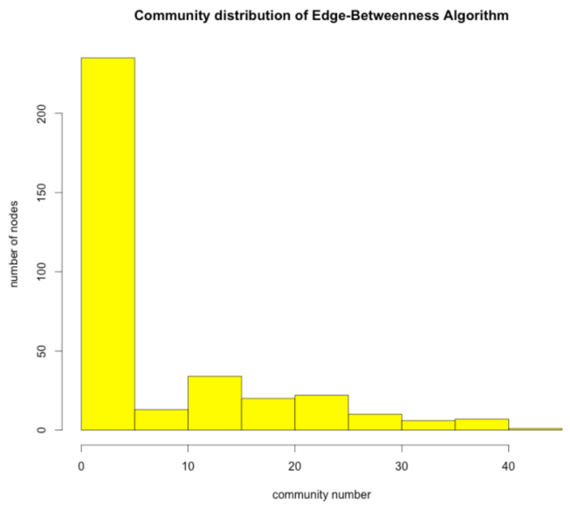
Find nodes in the graph that have more than 200 neighbors (core nodes). We find 41 core nodes in the network. The average degree of core nodes is 277.4. The community structure of core node 1 is shown below. Then use Fast-Greedy, Edge-Betweenness, and Infomap community detection algorithms to find the community structure. The network is plotted and compared in following figures, in which the core node is represented by the largest one in the center and communities are distinguished with colors. The modularity of the three algorithms is 0.413, 0.353 and 0.389 respectively.

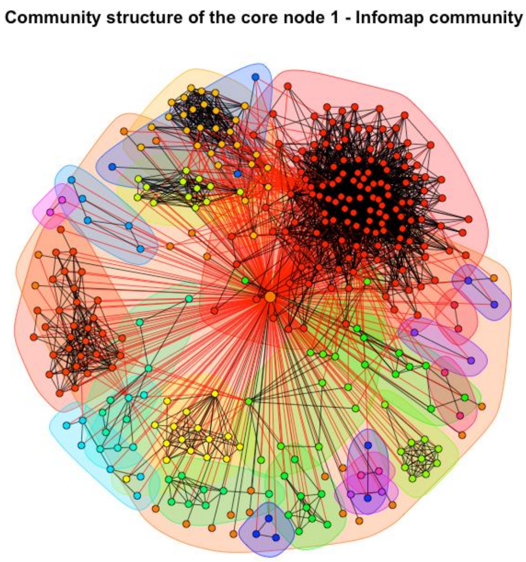

Community structure of the core node 1

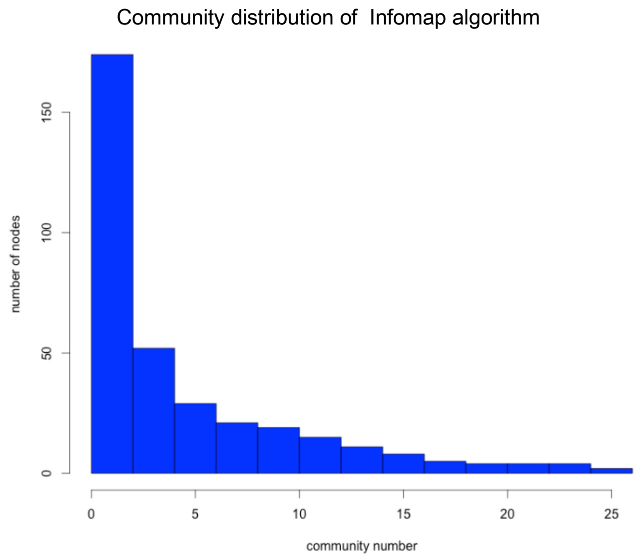

Community structure of the core node 1 - Fast-Greedy


Community distribution of Fast-Greedy Algorithm
number of nodes
0
20
40
60
80
100
1
2
3
4
5
6
7
8
community number


Community structure of the core node 1 - Edge-Betweenness

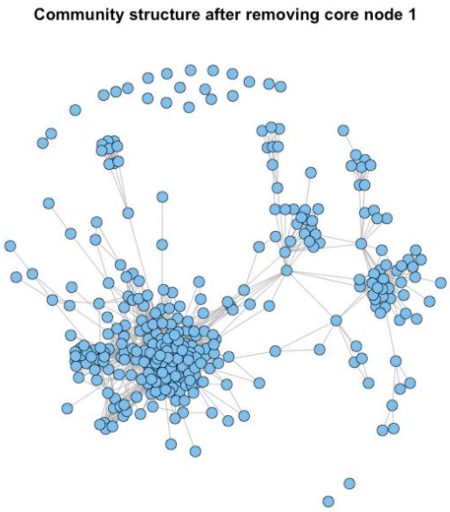

Community distribution of Edge-Betweenness Algorithm
number of nodes
0
50
100
150
200
0
10
20
30
40
community number



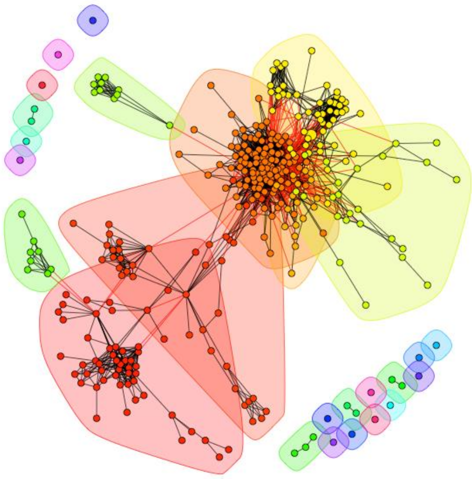


Problem 4

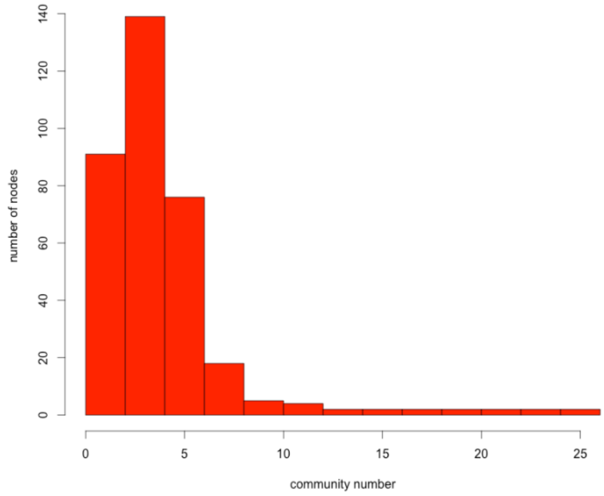
Remove the core node itself from its personal network and running the above community detection algorithms again. The results are shown as follows. The modularities of Fast-Greedy, Edge-Betweenness, and Infomap are 0.246, 0.151 and 0.245 respectively. Modularities are much smaller than not removing the core node itself. The difference can be clearly shown between histograms.



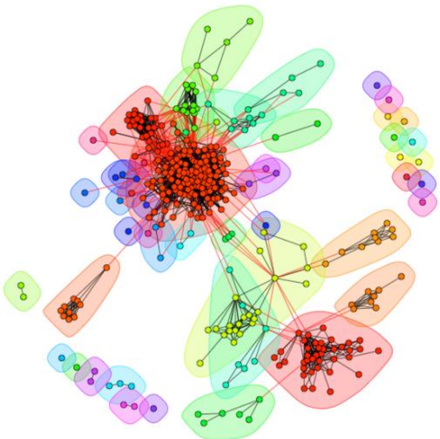
Community structure after removing core node 1 - Fast-Greedy



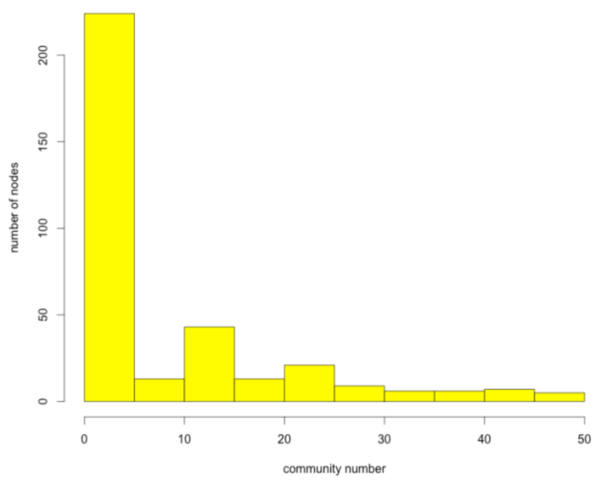
Community distribution of Fast-Greedy Algorithm



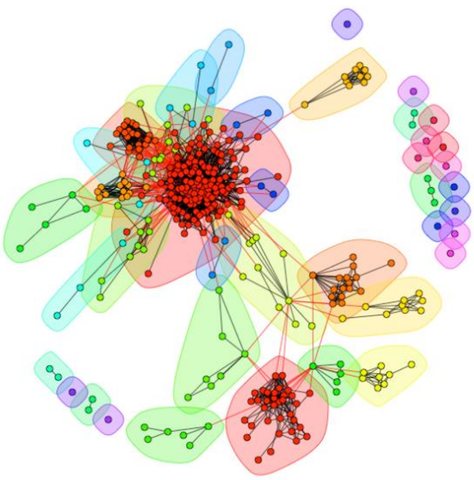
Community structure after removing core node 1 - Edge-Betweenness



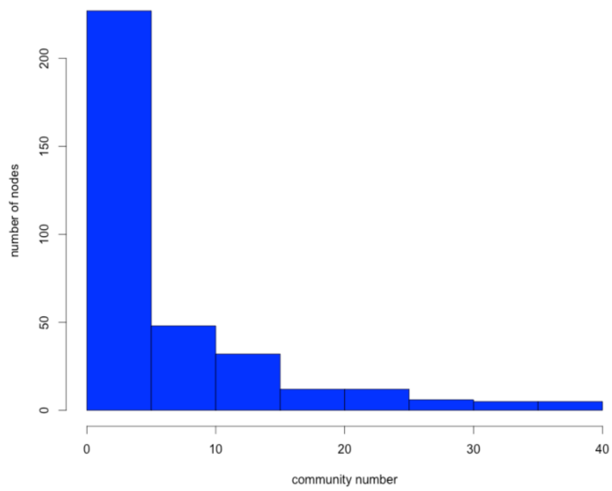
Community distribution of Edge-Betweenness Algorithm



Community structure after removing core node 1 - Infomap community



Community distribution of Infomap algorithm

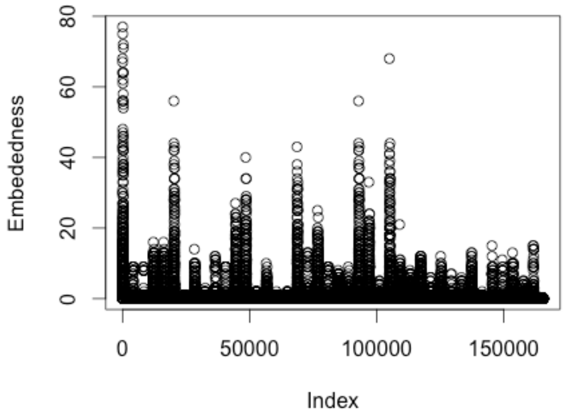


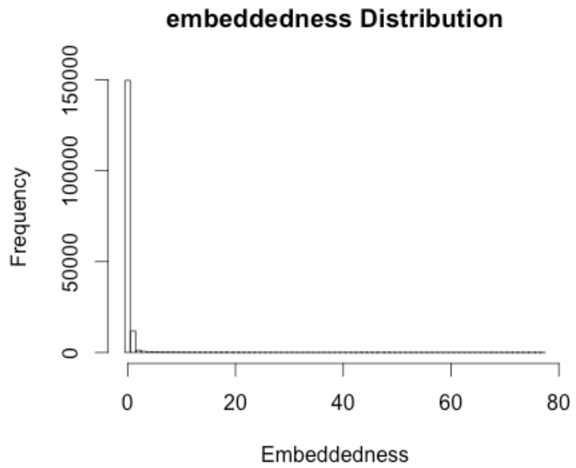
Problem 5

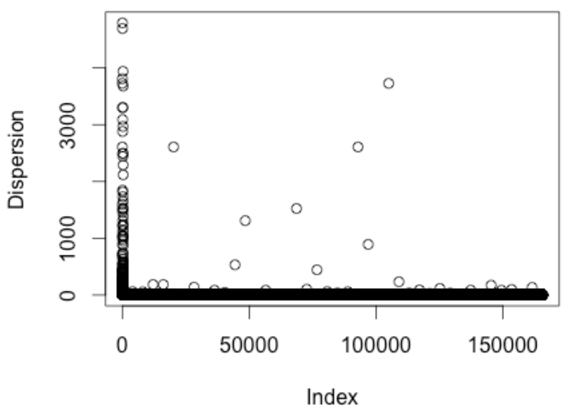
According to "Romantic Partnerships and the Dispersion of Social Ties: A Network Analysis of Relationship Status on Facebook", Lars Backstrom, Jon Kleinberg, *embeddedness* is defined as number of mutual friends two endpoints share, a quantity that typically increases with tie strength. We also define the absolute *dispersion* of u-v link, disp(u,v) to be the sum of all pairwise distances between nodes in C_{uv}, as measured in G_u - {u,v}; that is

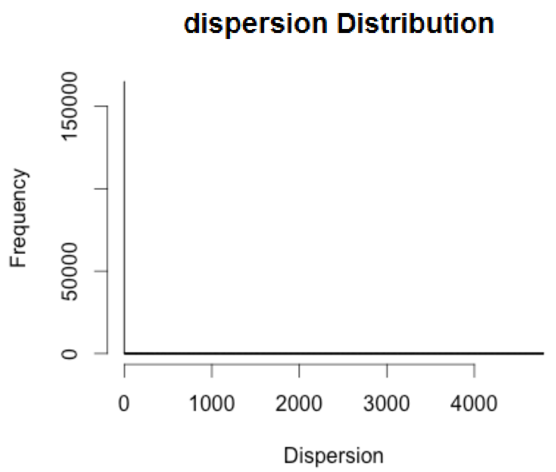
$$disp(u,v) = \sum_{s,t \in C_{uv}} d_v(s,t),$$

where d_v is a distance function on the nodes of C_{uv}. In this problem, we define $d_v(s,t)$ to be the function equal to 1 when s and t are not directly linked and also have no common neighbors in G_u other than u and v, and equal to 0 otherwise.
The distribution of *embeddedness* and *dispersion* over all the personal networks created by the core nodes in problem 3 are shown below.



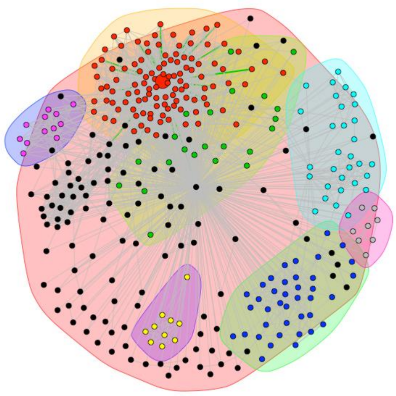




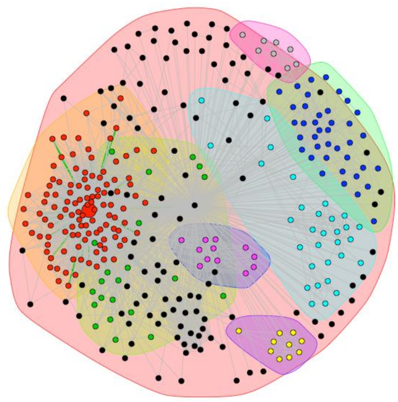


We select 3 personal networks showing their community structure with color, and highlight the node with maximum *dispersion*, *embeddedness*, or *dispersion*/*embeddedness* with bigger point; and mark the edges incident to this node with color green.
Personal network of core node 1:

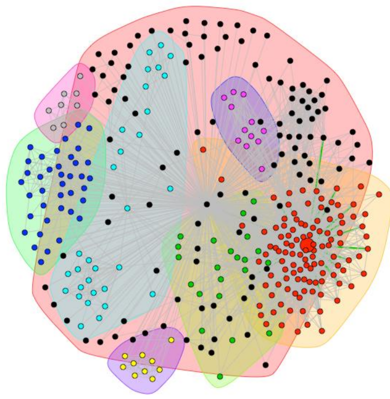
Max embeddedness



Max dispersion

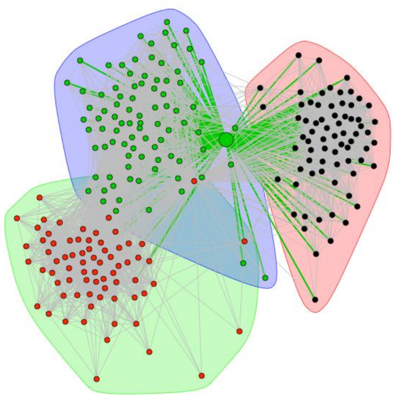


Max dispersion/embeddedness

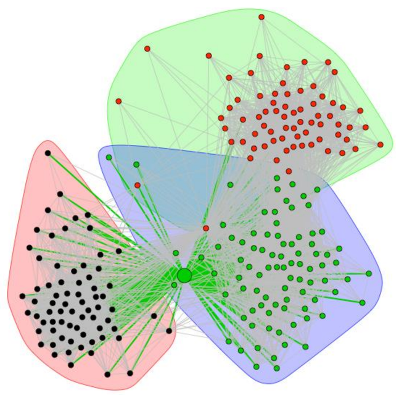


Personal network of core node 6:

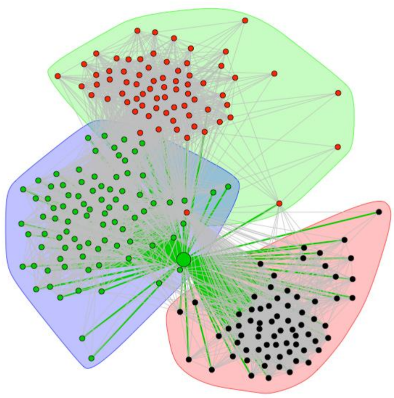
Max embeddedness



Max dispersion

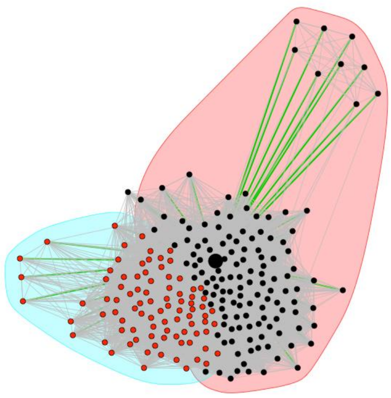


Max dispersion/embeddedness

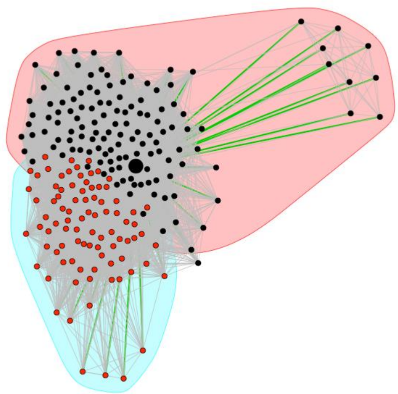


Personal network of core node 10:

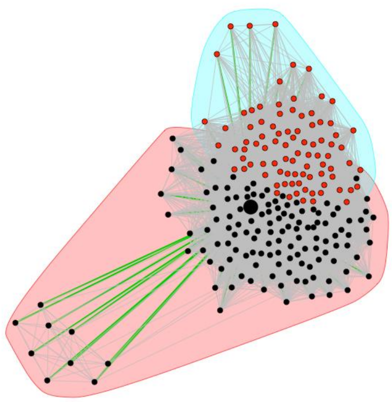
Max embeddedness



Max dispersion



Max dispersion/embeddedness



Problem 6

In graph theory the conductance of graph G=(V,E) measures how "well-knit" the graph is. The conductance of a cut $(S, \bar{S})$ in a graph is defined as:

$$\varphi(S) = \frac{\sum_{i \in S, j \in \bar{S}} a_{ij}}{\min\left(a(S), a(\bar{S})\right)}$$

where the a_{ij} are the entries of the adjacency matrix for G, so that

$$a(S) = \sum_{i\in S}\sum_{f\in V} a_{ij}$$

is the total number of the edges incident with S. The conductance of the whole graph is the minimum conductance over all the possible cuts.

In this problem, we use conductance to provide a measure for the structural features of the cut $(S, \bar{S})$. Let v be the sum of degrees of nodes in community S, and s be the number of edges with one endpoint in S and the other in $\bar{S}$, which is the complement of S. Then the conductance of S is

$$\varphi(S) = \frac{s}{v} = \frac{v - 2e}{v}$$

where e is the number of edges with both endpoints in S. Here communities are considered as groups of nodes with more intra-connections than inter-connections. Therefore, we prefer communities with small conductance, which is equivalent to densely-linked inside. Here we denote two types of each network.

$$type_1 = \min(\varphi(S))$$
$$type_2 = \max(\varphi(S))$$

Type_1 communities have minimum conductance, with dense links inside and sparse links outside. On the other hand, type_2 have maximum conductance with denser links to outside. They are obtained using fast-greedy algorithm,

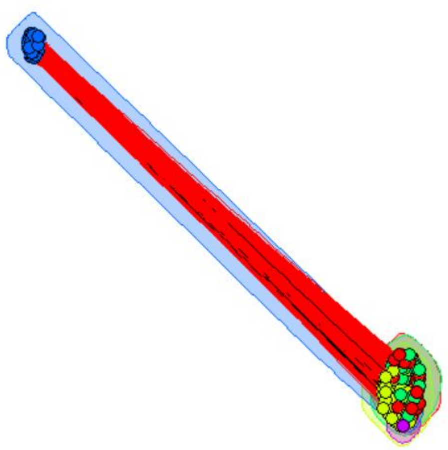
type_1
4 2 2 3 1 1 2 2 3 1 2 3 1 3 1 2 2 2 3 1 2 1 2 3 1 3 1 3 2 2 1 1 1 2 3 2 2 2 2 3 2

type_2
3 4 1 2 4 2 1 1 1 2 1 1 2 2 2 3 1 3 2 2 6 2 3 2 2 2 2 2 3 1 3 2 2 1 1 1 1 1 1 1 1

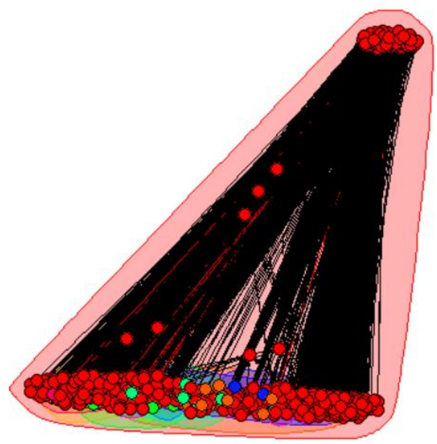
The number of 1-6 means the index of community in each core node’s personal network.

Problem 7
We now run the same kind of analysis on another real social network with tagged relationships. Google+ ,unlike Facebook, has a directed network structure, where you can have someone in your circles regardless of whether they have you in their circles or they don’t. Circles are tags you put on your relationships when you add people. After extract all the personal network with its circles, we choose one to show its community structure using both Walktrap and Infomap algorithms.

Community Structure of ego node 1 by Walktrap

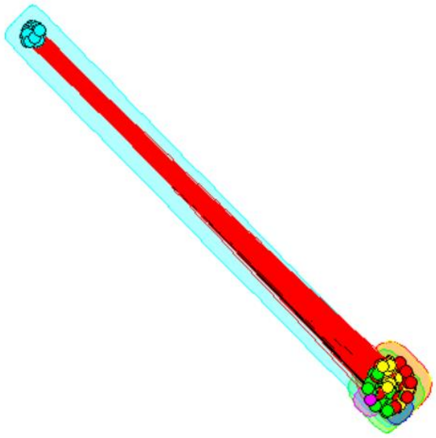


Community Structure of ego node 1 by Infomap

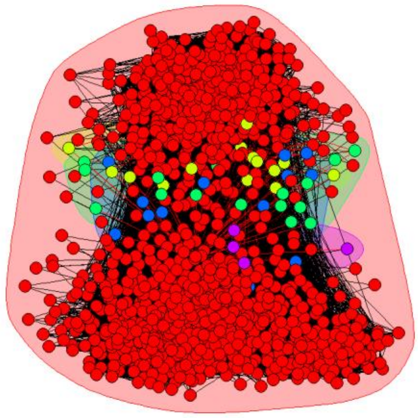


If viewing it as an undirected graph, the community structure is shown below.

Community Structure of ego node 1 by Walktrap



Community Structure of ego node 1 by Infomap



With further analysis, if we can tag the relationships more clearly and reduce the overlap between different circles, we can get better performance given by community detection algorithms.