EE 232E Project 2

Social Network Mining

Wei DU
UID: 005024944
Email: ericdw@g.ucla.edu

Xiao Yang
UID: 104946787
Email: avadayang@icloud.com

Fangyao Liu
UID:204945018
Email:fangyaoliu@g.ucla.edu

Ruchen Zhen
UID:205036408
Email:rzhen@ucla.edu

1 Facebook network

1.1 Structural properties of the facebook network

In this section, we will study many properties of the a realistic network--Facebook, which is a connected graph with diameter = 8.

Its degree distribution is as follows:

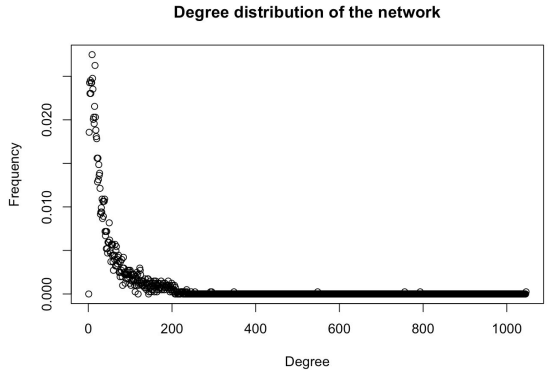


Figure 1. Degree distribution of Facebook

And it has average degree = 43.69101.

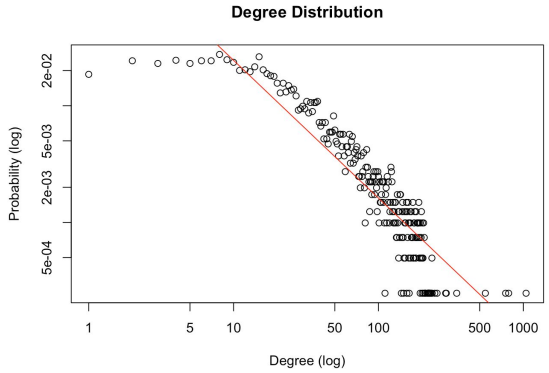


Figure 2. Degree distribution of Facebook (log-log scale)

The slope is -1.18

1.2 Personalized network

A personalized network of a node i is defined as the graph only contains i and all its neighbors. In this part, we will study structural properties of node 1's personalized network. Its graph is given below and it has 348 nodes and 2866 edges. Diameter = 2 which is obvious since node 1 has more than 2 neighbors.

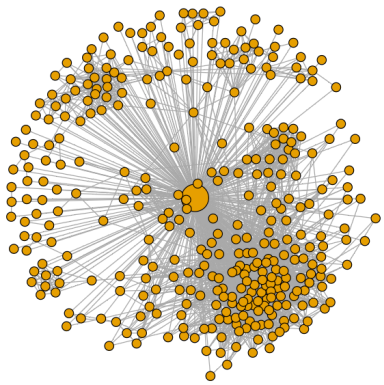


Figure 3. Personalized network of node 1

Trivial upper and lower bound for the diameter of the personalized network is [0,2]. It is 0 when there is only one node in graph, and 2 when there are more than or equal three nodes in graph, which is target with 2 or more friends.

1.3 Core node's personalized network

In this section we will focus on personalized network with core nodes, which is defined as the nodes that have more than 200 neighbors.

In fact, from our Facebook network dataset, we discovered *40* core nodes with average degree = *279.375*

1.3.1 Community structure of core node's personalized network and that with core node removed

In this section, we apply three popular community detection algorithms (Fast-Greedy, Edge-Betweenness, and Infomap) to study personalized networks of core nodes id= 1, 108, 349, 484, 1087. Then we remove the core node from its personalized network, and apply same algorithm to discover the community structure in the remaining graph.

To quantify each result of community detection, we used Modularity-Q as the main measurement method which is defined as

$$Q = \sum_{i=1}^{c} (e_{ii} - a_i^2)$$

where

$$e_{ij} = \sum_{vw} \frac{A_{vw}}{2m} 1_{v \in ci} 1_{w \in cj} \qquad a_i = \frac{k_i}{2m}$$

Where c is for community, m is for number of edges, A_{ij} is an entry in the adjacency matrix where A_{ij} = 1 when j and i are connected, and k is the degree of node.

1.3.1 community detection on core node 1's personalized network

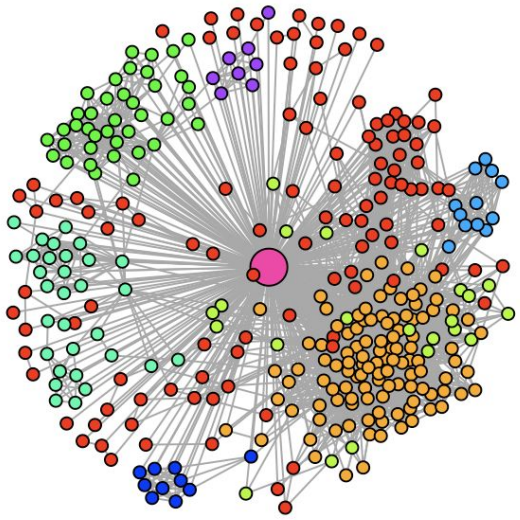


Figure 4. Visualization of result running fast-greedy on core node 1's personalized network with Modularity Q = 0.413

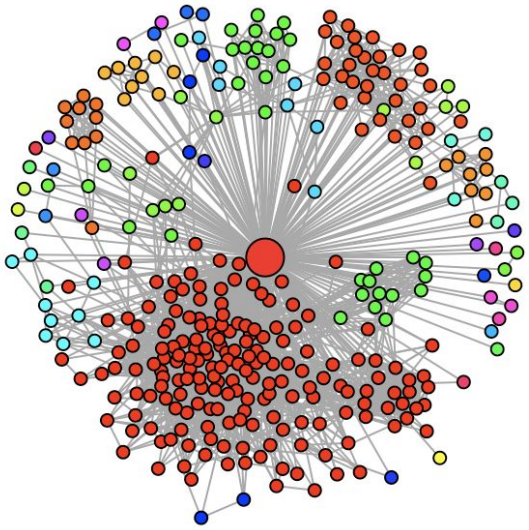


Figure 5. Visualization of result running Edge-Betweenness on core node 1's personalized network with Modularity Q = 0.353

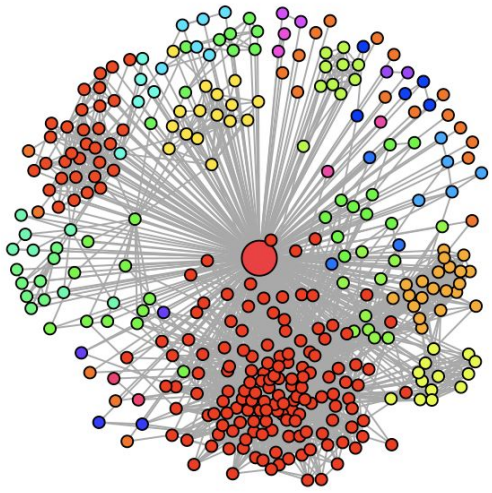


Figure 6. Visualization of result running Infomap on core node 1's personalized network with Modularity Q = 0.389

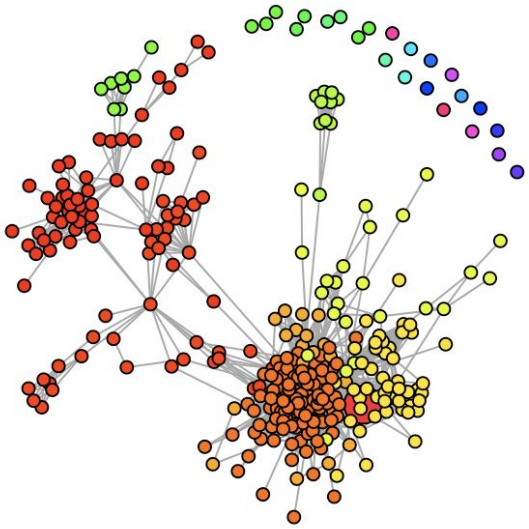


Figure 7. Visualization of result running fast-greedy on core node 1's personalized network with core removed. Modularity Q = 0.441

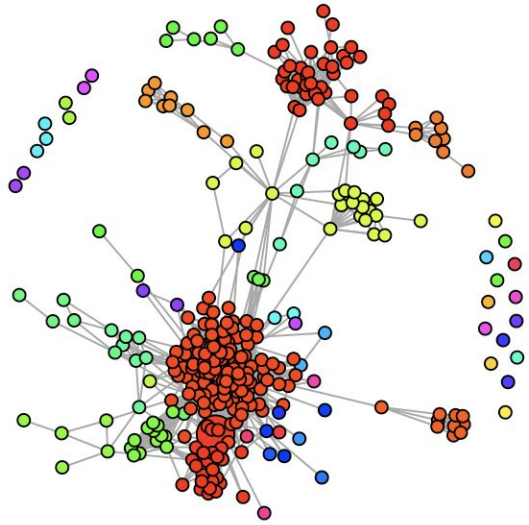


Figure 8. Visualization of result running Edge-Betweenness on core node 1's personalized network with core removed. Modularity Q = 0.416

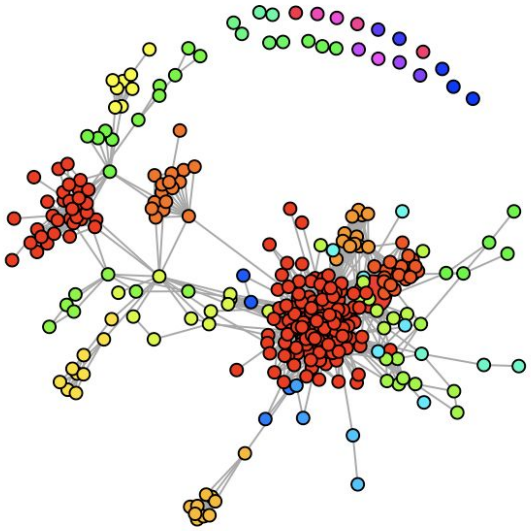


Figure 9. Visualization of result running Infomap on core node 1's personalized network with core removed. Modularity Q = 0.418

1.3.2 community detection on core node 108's personalized network

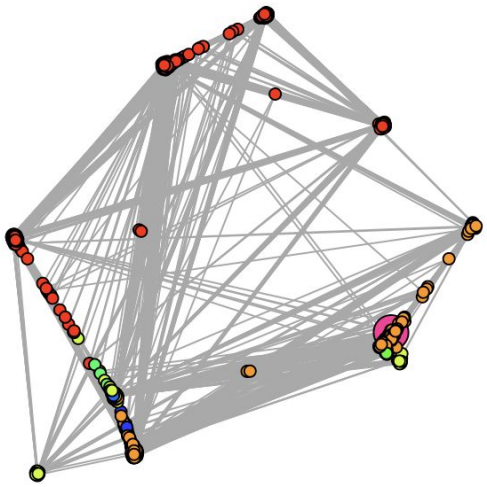


Figure 10. Visualization of result running fast-greedy on core node 108's personalized network with Modularity Q = 0.43592

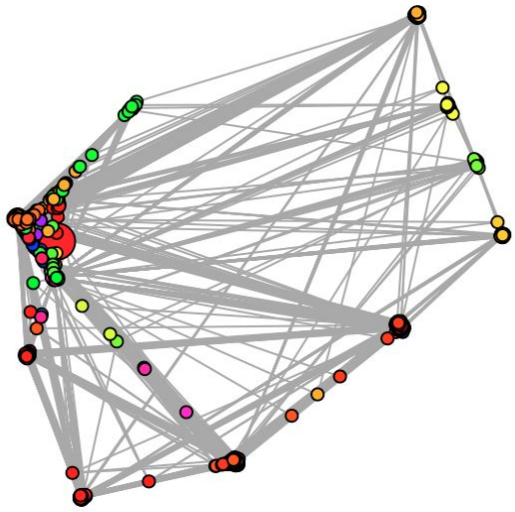


Figure 11. Visualization of result running Edge-Betweenness on core node 108's personalized network with Modularity Q = 0.506754916538902

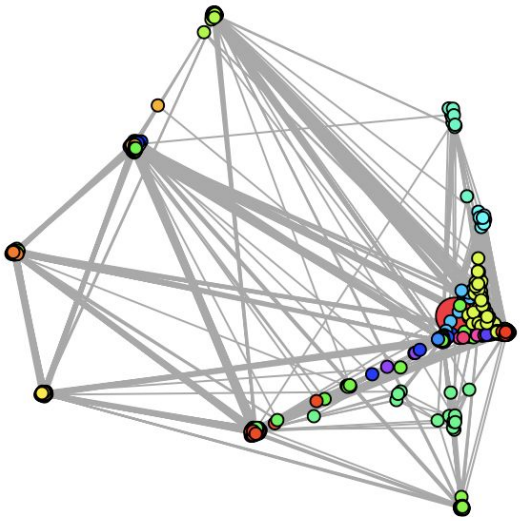


Figure 12. Visualization of result running Infomap on core node 108's personalized network with Modularity Q = 0.50866

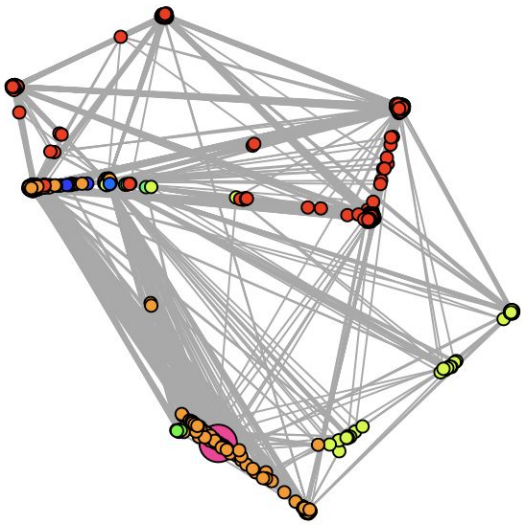


Figure 13. Visualization of result running fast-greedy on core node 108's personalized network with core removed. Modularity Q = 0.43595

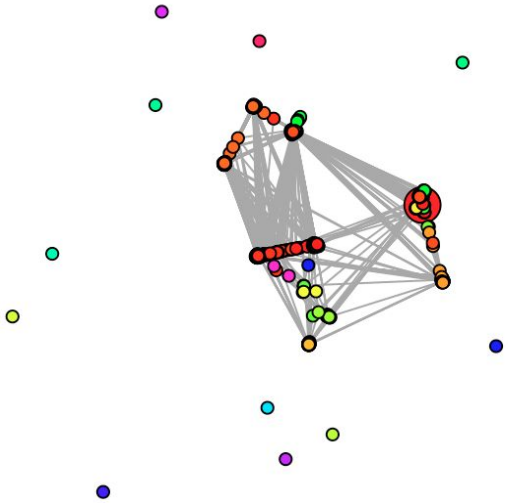


Figure 14. Visualization of result running Edge-Betweenness on core node 108's personalized network with core removed. Modularity Q = 0.521321576382217

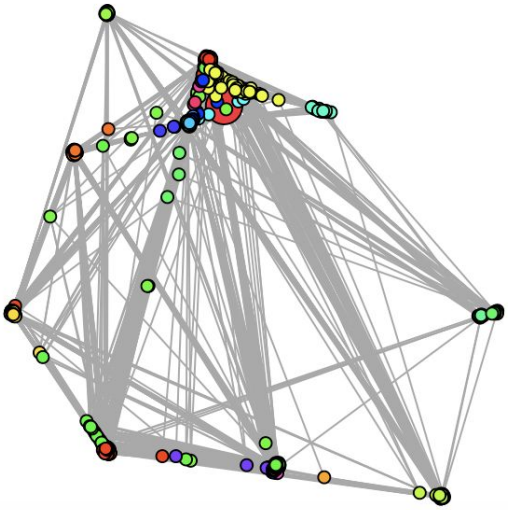


Figure 15. Visualization of result running fast-greedy on core node 1's personalized network with core removed. Modularity Q = 0.50819

1.3.3 community detection on core node 349's personalized network

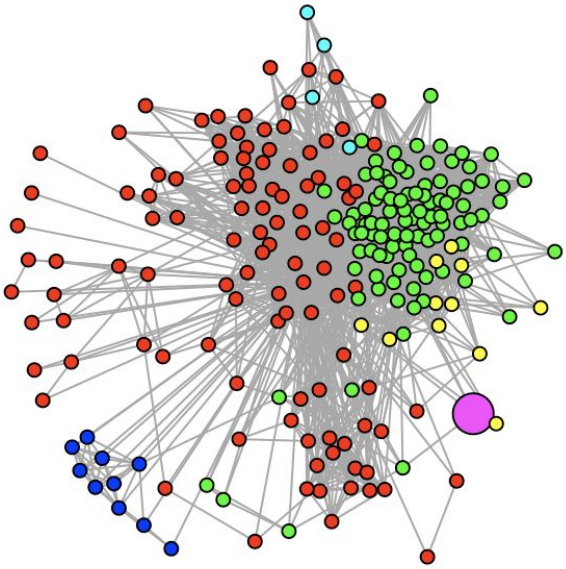


Figure 16. Visualization of result running fast-greedy on core node 349's personalized network with Modularity Q = 0.25171

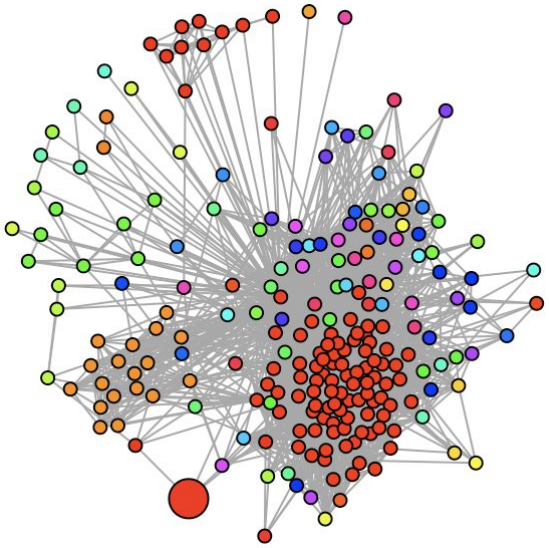


Figure 17. Visualization of result running Edge-Betweenness on core node 349's personalized network with Modularity Q = 0.133528

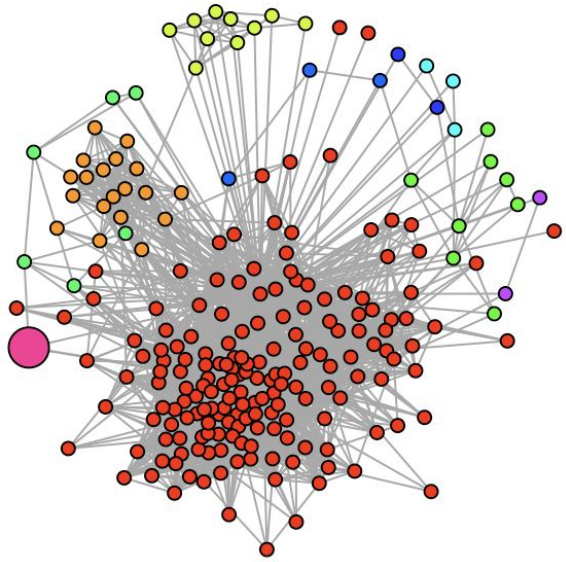


Figure 18. Visualization of result running Infomap on core node 349's personalized network with Modularity Q = 0.095464

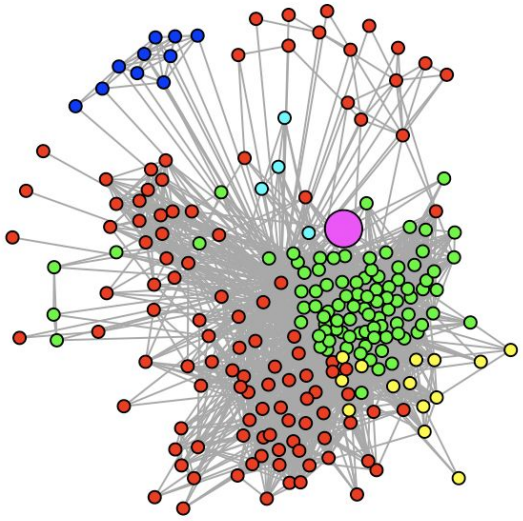


Figure 19. Visualization of result running fast-greedy on core node 349's personalized network with core removed. Modularity Q = 0.25749

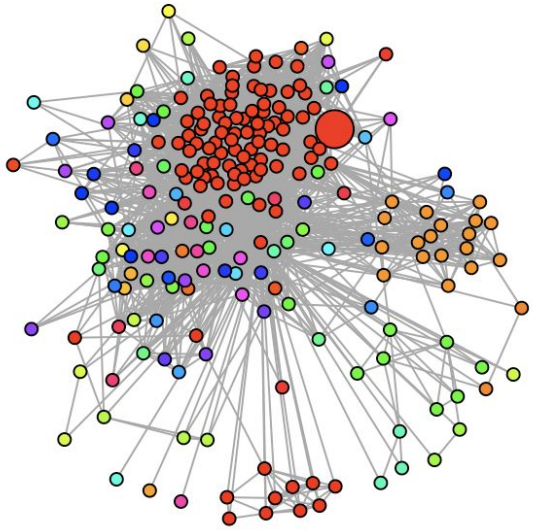


Figure 20. Visualization of result running Edge-Betweenness on core node 349's personalized network with core removed. Modularity Q = 0.13373

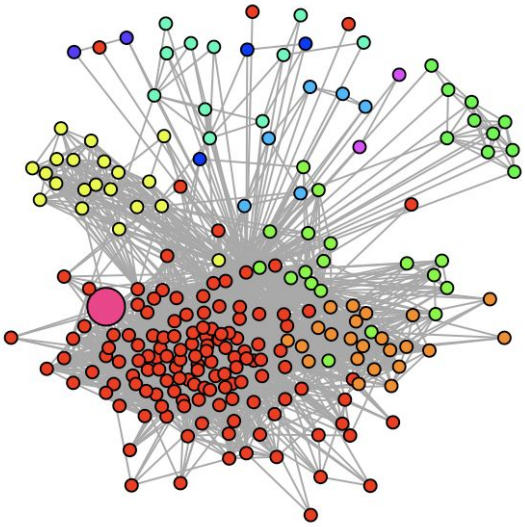


Figure 21. Visualization of result running Infomap on core node 349's personalized network with core removed. Modularity Q = 0.20365

1.3.4 community detection on core node 484's personalized network

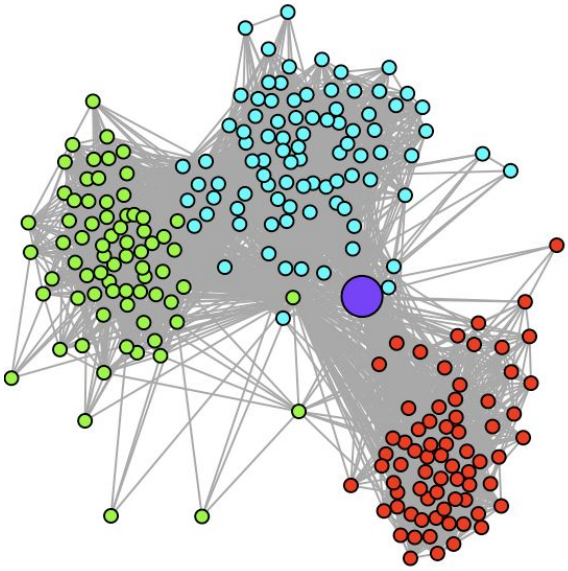


Figure 22. Visualization of result running fast-greedy on core node 484's personalized network with Modularity Q = 0.50700

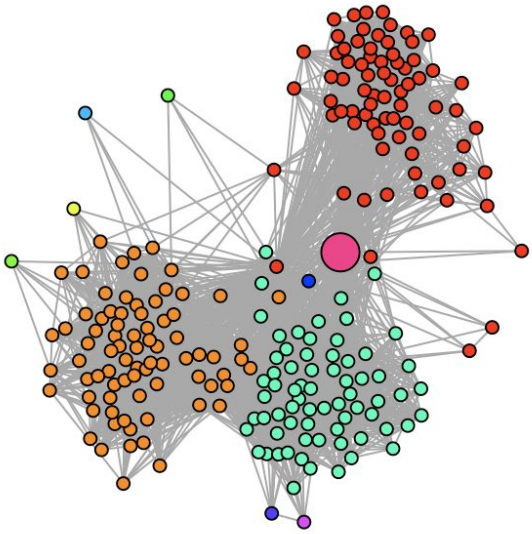


Figure 23. Visualization of result running Edge-Betweenness on core node 484's personalized network with Modularity Q = 0.48910

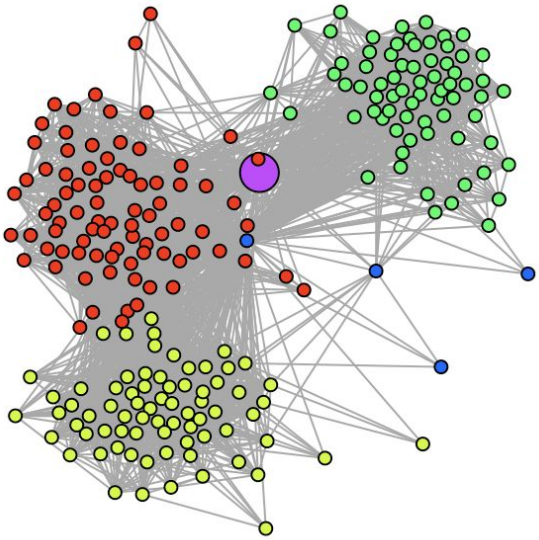


Figure 24. Visualization of result running Infomap on core node 484's personalized network with Modularity Q = 0.51528

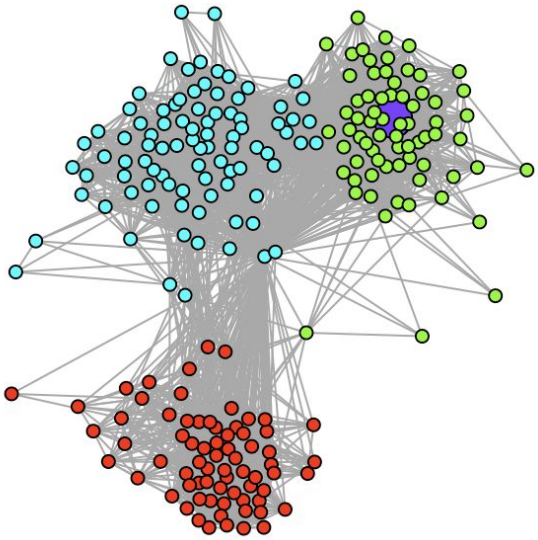


Figure 25. Visualization of result running fast-greedy on core node 484's personalized network with core removed. Modularity Q = 0.52113

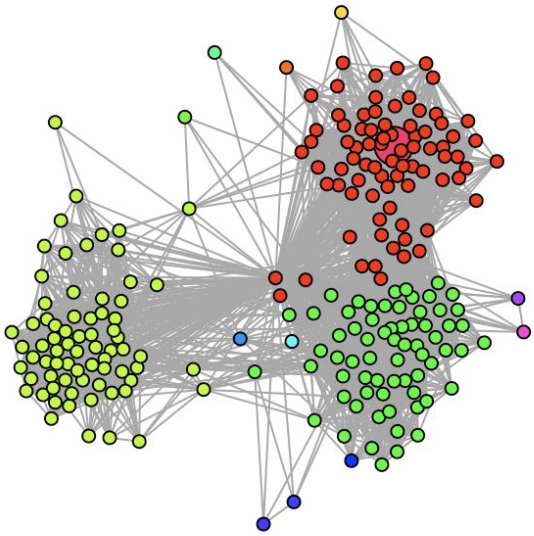


Figure 26. Visualization of result running Edge-Betweenness on core node 484's personalized network with core removed. Modularity Q = 0.49562

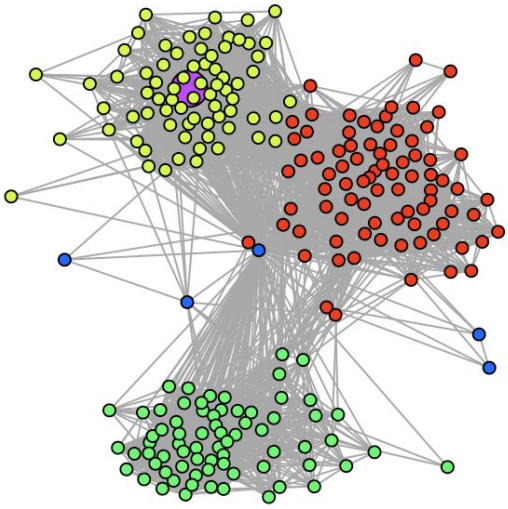


Figure 27. Visualization of result running Infomap on core node 484's personalized network with core removed. Modularity Q = 0.52915

1.3.5 community detection on core node 1087's personalized network

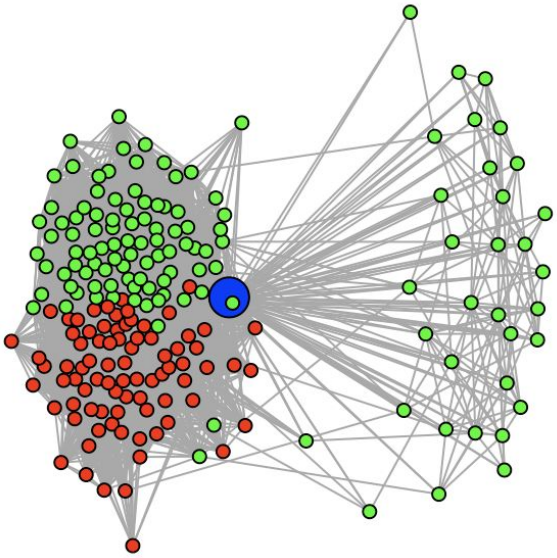


Figure 28. Visualization of result running fast-greedy on core node 1087's personalized network with Modularity Q = 0.14553

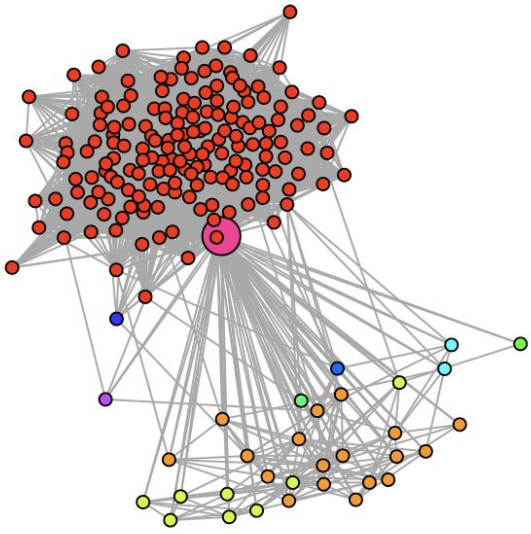


Figure 29. Visualization of result running Edge-Betweenness on core node 1087's personalized network with Modularity Q = 0.02762

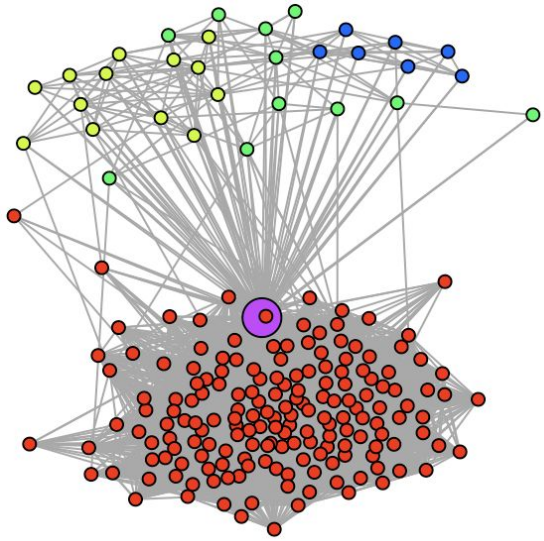


Figure 30. Visualization of result running Infomap on core node 1087's personalized network with Modularity Q = 0.02691

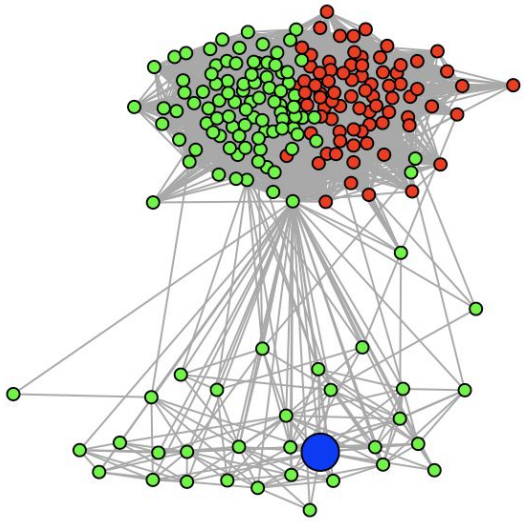


Figure 31. Visualization of result running fast-greedy on core node 1087's personalized network with core removed. Modularity Q = 0.14820

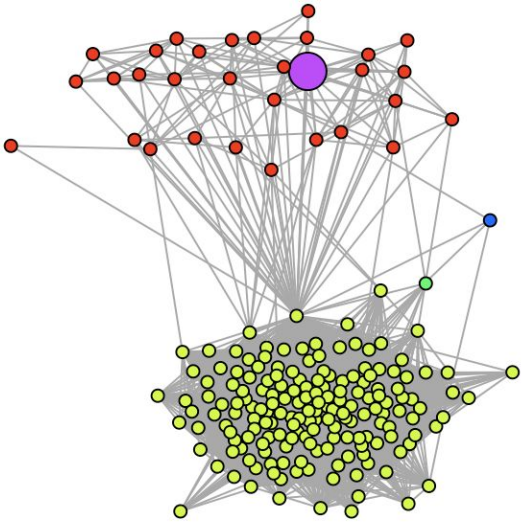


Figure 32. Visualization of result running Edge-Betweenness on core node 1087's personalized network with core removed. Modularity Q = 0.03250

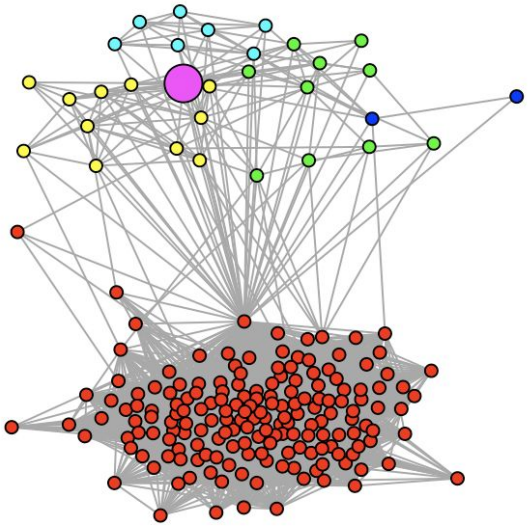


Figure 33. Visualization of result running Infomap on core node 1087's personalized network with core removed. Modularity Q = 0.02737

1.3.6 Discussion

Generally speaking, we have several findings:

1) once core node is removed from network, modularity of clustered community increases.

This can be explained by the fact that core nodes always share many edges with every community, bringing a very large a_i to the Q-modularity equation.

2) while considering visualization of network about node 484 and 1087, we believe, the even the sizes of communities be the more likely to see an increase in modularity if inter-connectivity between communities being roughly equal.

A possible explanation would be that: from figures related to node 1087, smaller community has some strong connectivity over few nodes in bigger community ; therefore while there is tense intra-communication for bigger community, it also cause many inter-connection for smaller communities. Thus the overall modularity is affected severely since cost of inter-connectivity is squared.

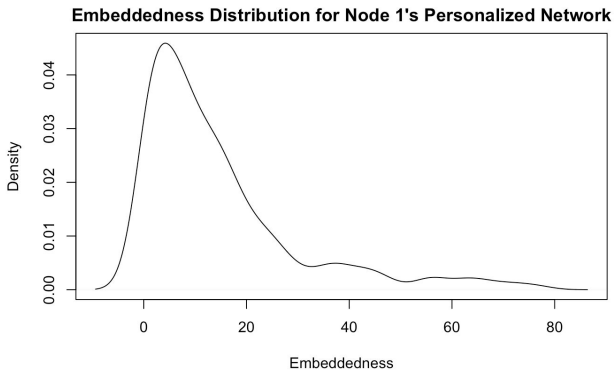
3) Edge-Betweenness is generally the slowest algorithm since each iteration it goes through graph and computes the edge betweenness again.

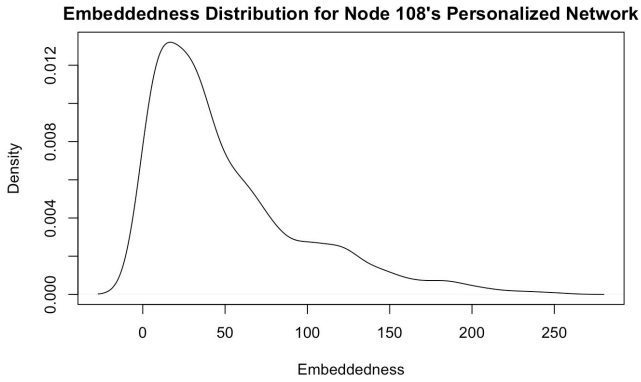
4) from graph related to 108 we noticed that nodes within community are the tightest connected, since layout of plotting is Fruchterman-Reingold, by which nodes share more connections are closer to each other. Therefore, with high intra-connectivity, modularity is relative high among all.

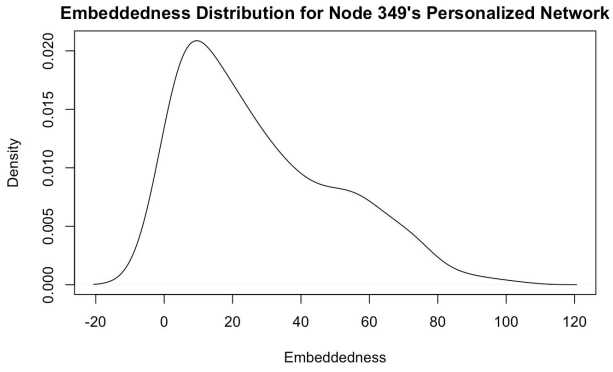
Q11:

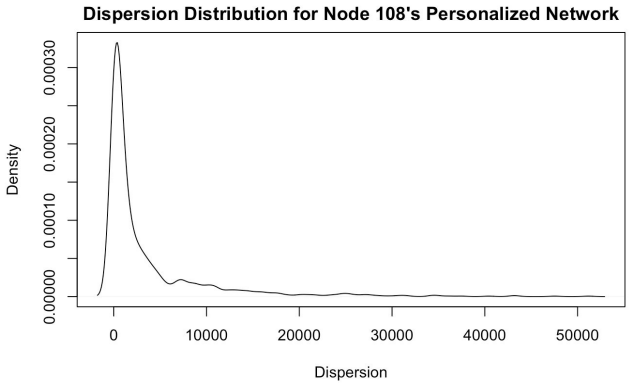
Typically, the embeddedness of a node is larger if its degree is higher. With higher degree, the node will have more neighbours, thus a higher probability to have more mutual connections with the core node.

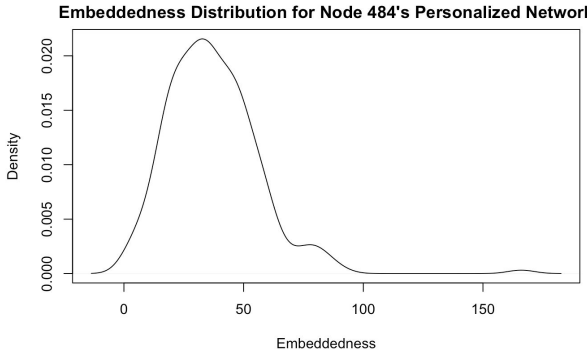
Q12: Distribution of Embeddedness and Dispersion

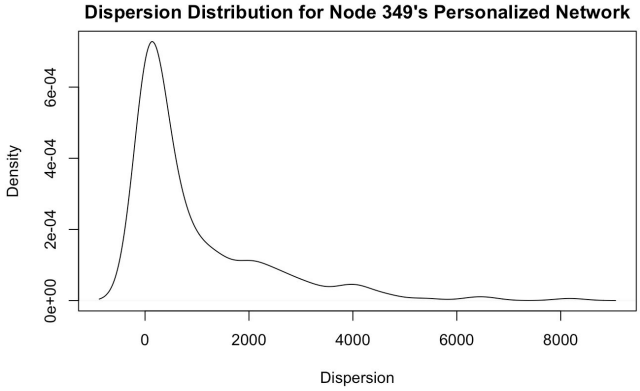


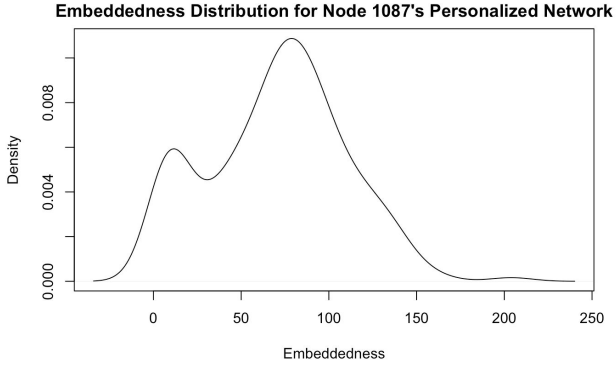


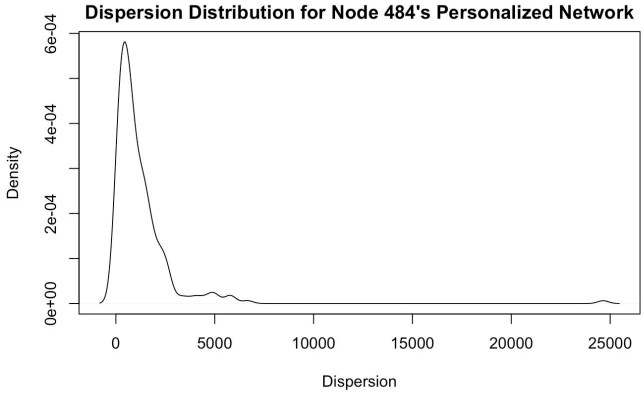

Embeddedness Distribution for Node 349's Personalized Network
Density
0.000
0.005
0.010
0.015
0.020
-20
0
20
40
60
80
100
120
Embeddedness

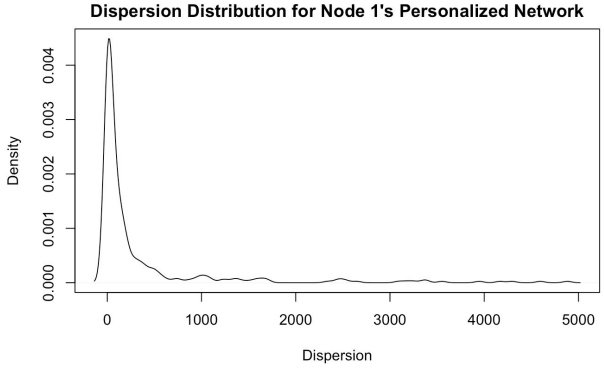

Dispersion Distribution for Node 108's Personalized Network
Density
0.00000
0.00010
0.00020
0.00030
0
10000
20000
30000
40000
50000
Dispersion

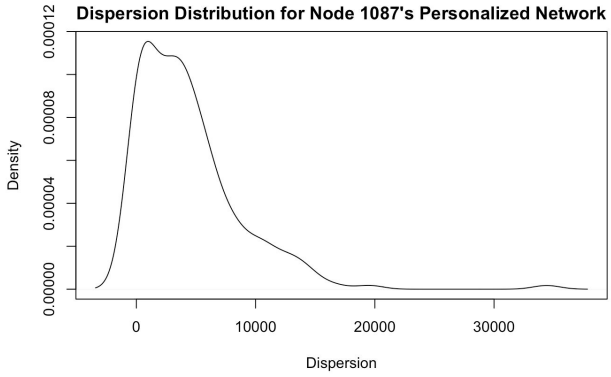

Embeddedness Distribution for Node 484's Personalized Network
Density
0.000
0.005
0.010
0.015
0.020
0
50
100
150
Embeddedness


Dispersion Distribution for Node 349's Personalized Network
Density
0e+00
2e-04
4e-04
6e-04
0
2000
4000
6000
8000
Dispersion


Embeddedness Distribution for Node 1087's Personalized Network
Density
0.000
0.004
0.008
0
50
100
150
200
250
Embeddedness

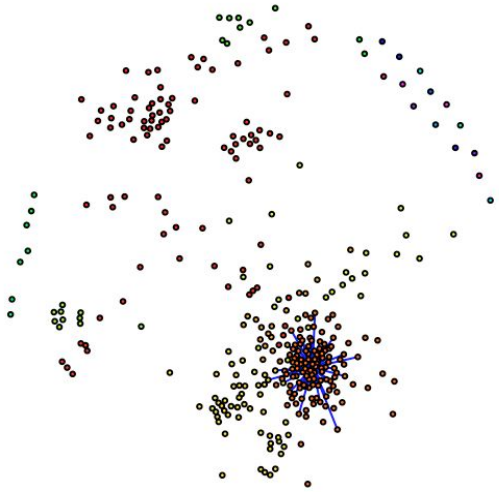

Dispersion Distribution for Node 484's Personalized Network
Density
0e+00
2e-04
4e-04
6e-04
0
5000
10000
15000
20000
25000
Dispersion


Dispersion Distribution for Node 1's Personalized Network
Density
0.000
0.001
0.002
0.003
0.004
0
1000
2000
3000
4000
5000
Dispersion



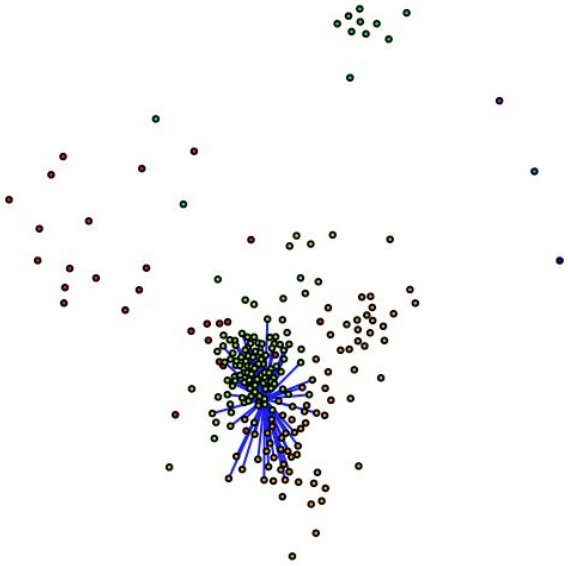
Q13 Max Dispersion with Node and Incident Edges Highlighted

Node 1



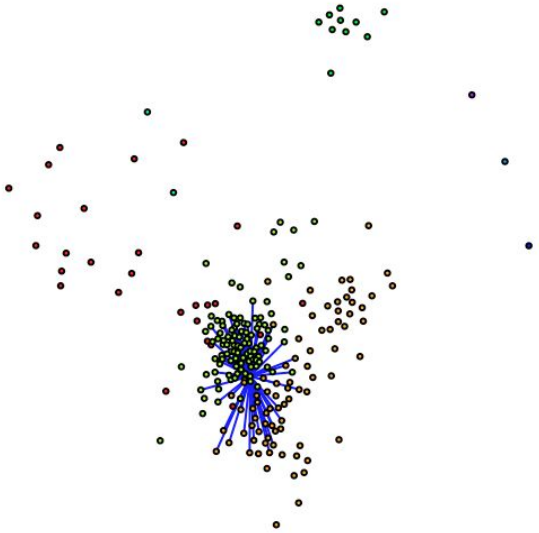
Node 108

Node 349

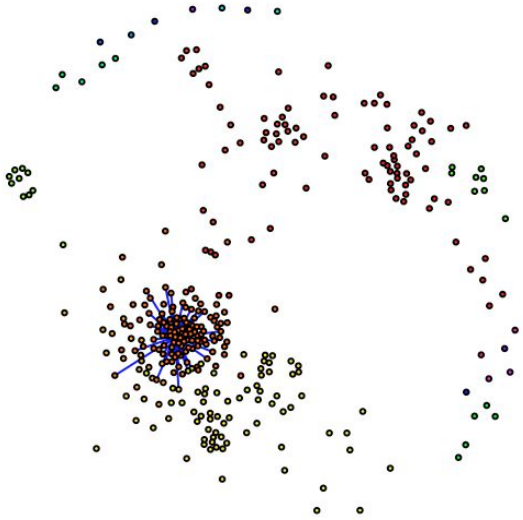


Q14.1 Max Embeddedness with Node and Incident Edges Highlighted

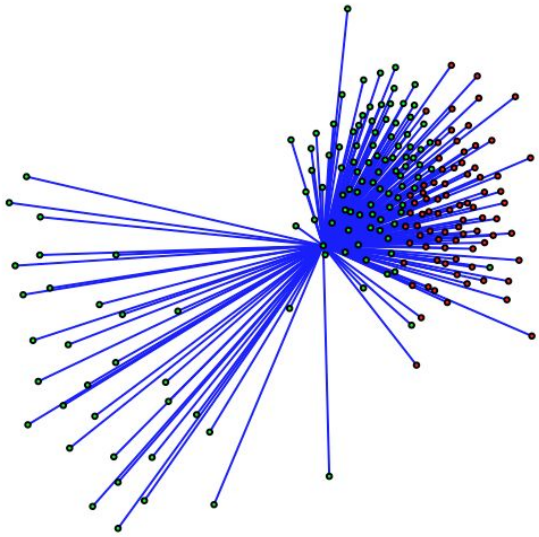
Node 484



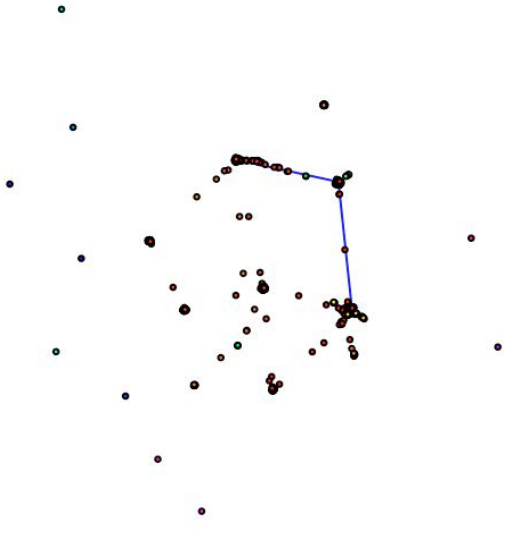
Node 1



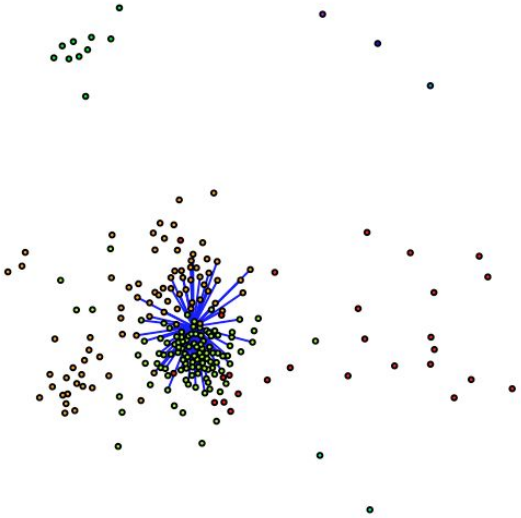
Node 1087



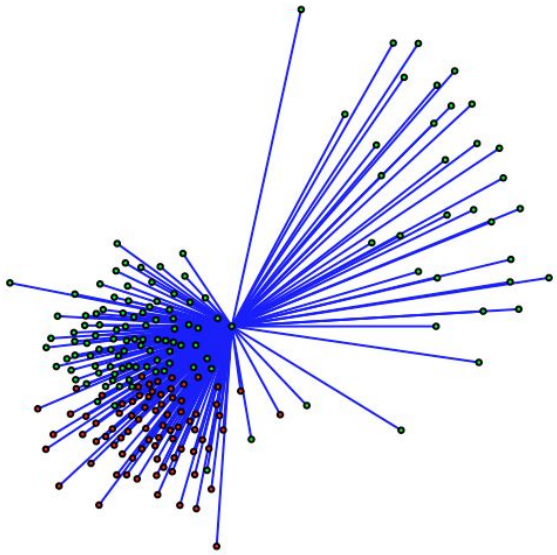
Node 108



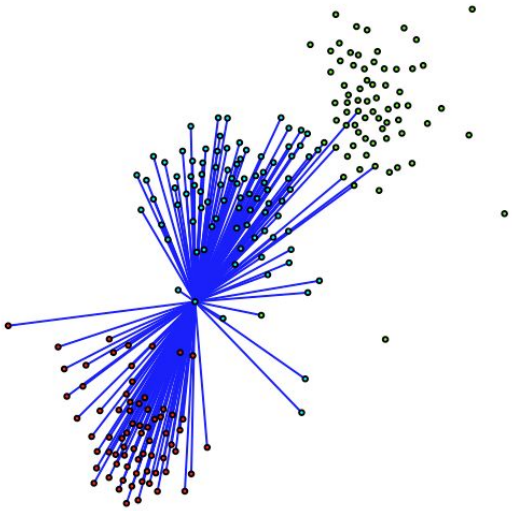
Node 349



Node 1087

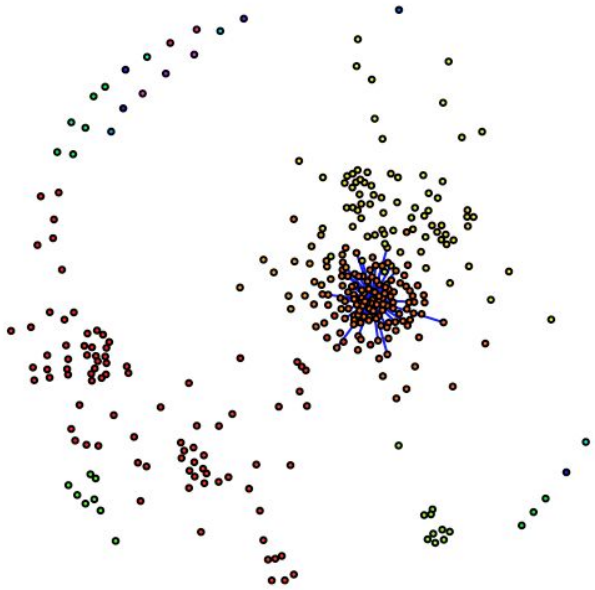


Node 484

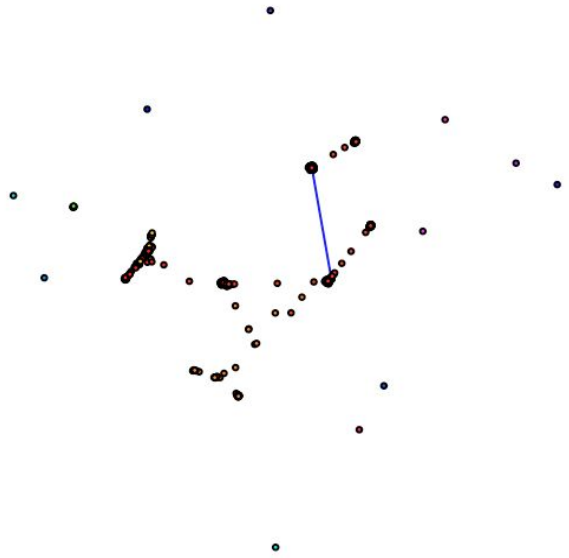


Q14.2 Max Dispersion/Embeddedness with Node and Incident Edges Highlighted

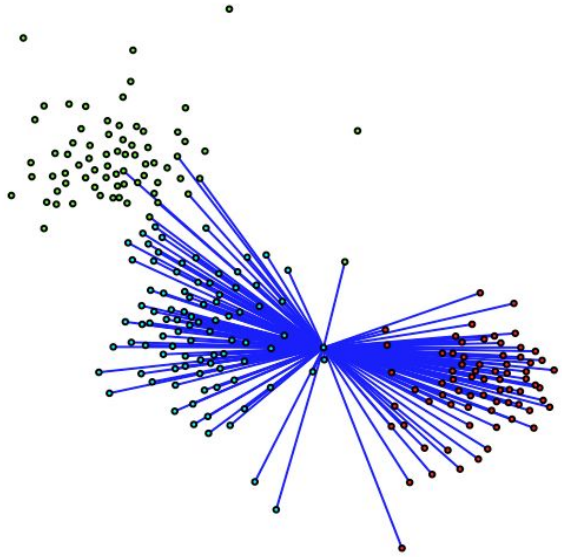
Node 1



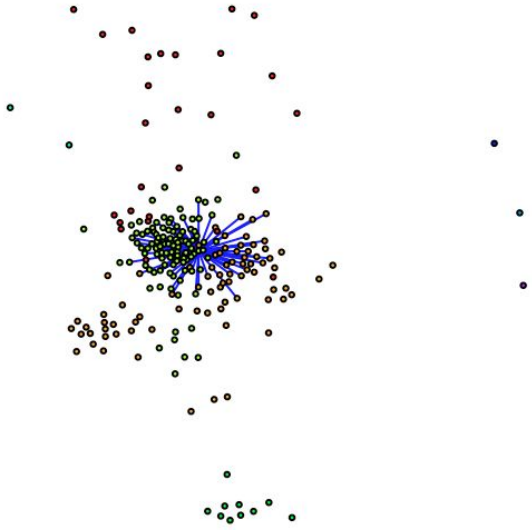
Node 108



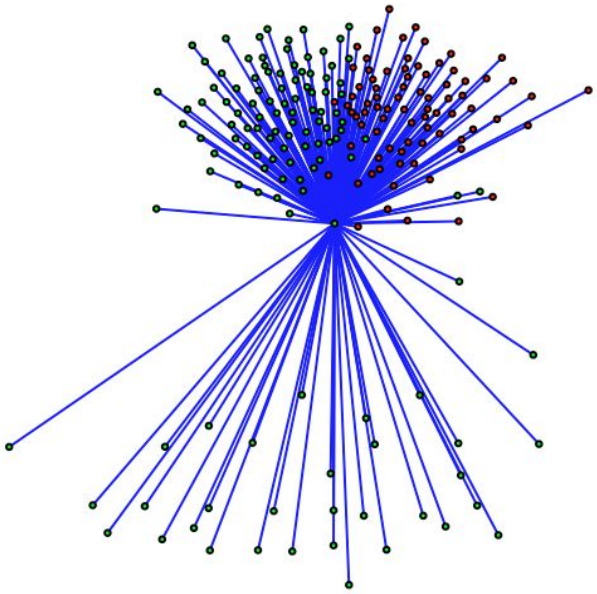
Node 484



Node 349



Node 1087



Q15 Characteristics of a node revealed by each of this measure

As can be seen in the figure of Q13 and Q14, we can see that the performance of different measurements finding the most relevant node to the core node is:

Dispersion/Embeddedness > Dispersion > Embeddedness

How do we judge whether the distinguishing performance of a measurement is better or not? If the nodes with best measurements (Dispersion/Embeddedness or Dispersion or Embeddedness) connect with nodes of as much different communities as possible, then we say its performance is good. This is very intuitive. If two students in UCLA have many mutual friends which are all UCLA students, they are not necessary need to know each other. But if they have mutual friend of both UCLA students, family members, colleagues, or even private doctor, they are very likely to have close relationship.

1.4 Friend recommendation in personalized networks

In this part of the question, we will investigate personal network with ID 415. Randomly cutting the edges in this network and try to recommend friends with some neighbourhood-based measures to these nodes whose edges have been cut. At last, we will calculate accuracy by checking how many recommended connections are the edges we deleted.

1.4.1 Neighborhood based measure

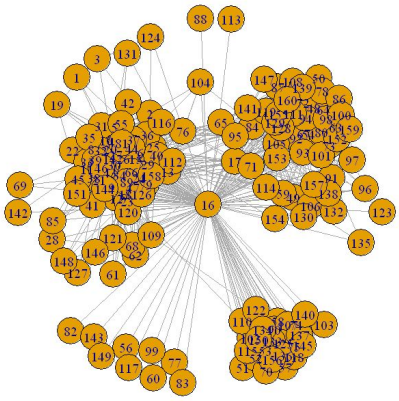
In this section, three measures are introduced. They are: Common neighbor measure, Jaccard measure and Adamic-Adar measure These three measures will be used in later friend recommendation.

1.4.2 Friend recommendation using neighborhood based measures

In this section, the method of recommending friends is introduced. First step is for each node that is not the neighbor of target user, compute the similarity measure between node and target node. Then rank these nodes and recommend the top k nodes as potential friends to target user.

1.4.3 Creating the list of users

In this section, we are going the reduce the original graph to personal network of node ID 415.



Then we shall find the neighbors of node ID 415 whose degree is 24 as the list of users that we want to recommend new friends to. After programming the personal network, we find out the number of degree 24 is :

Q16: $|Nr| = 11$

1.4.4 Average accuracy of friend recommendation algorithm

We will apply three neighbor measures to recommend friends to our degree 24 user list. First of all, remove edge of node in user list with a probability of 0.25. It is same as deleting friends of these nodes. Then, use one of the three measurements to rank all the nodes in personal network and recommend the top k (k is equal to the number of edges deleting for each node). After that, we calculate the accuracy of each user by comparing how many common nodes between the deleted nodes and recommended nodes. Iterating this process ten times and get average of each candidate in the degree 24 list. At last, compute the average score of degree 24 list. Finally, we reach a conclusion presented as the following table:

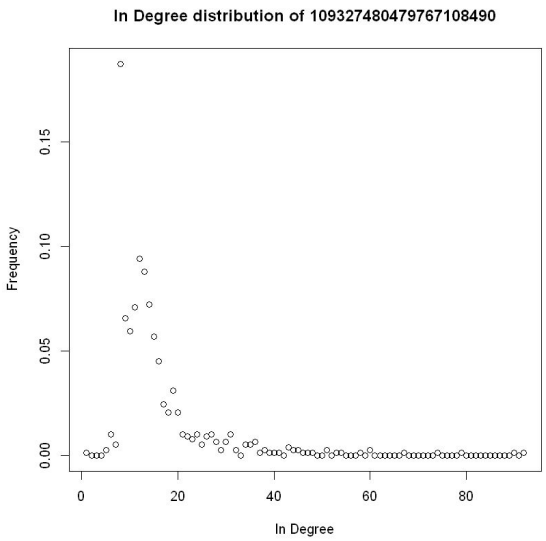
Q17:

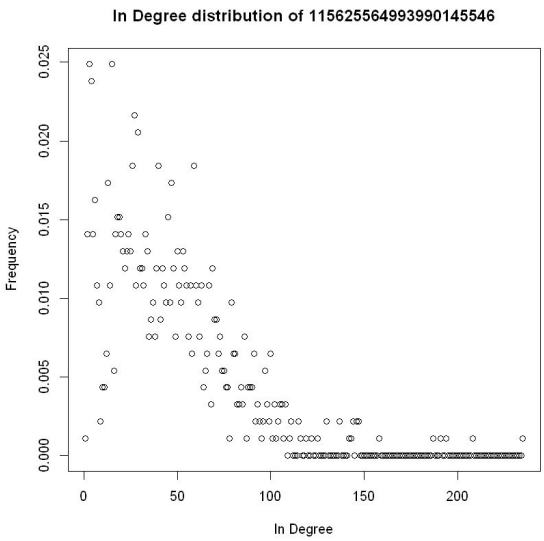
	Common Neighbors measure	Jaccard measure	Adamic Adar measure
Average accuracy	0.83718	0.80353	0.82812

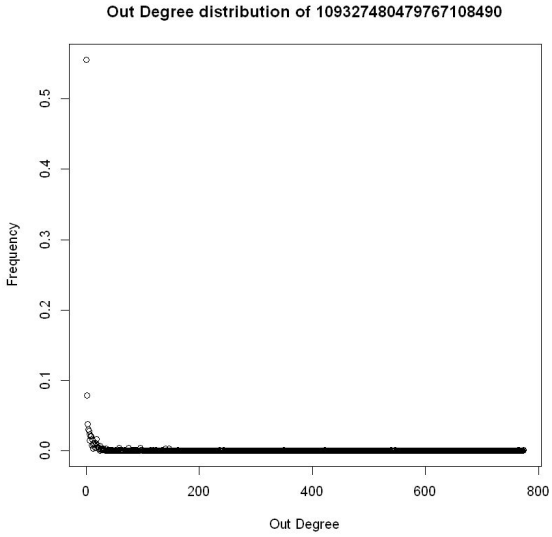
All three methods return good results and common neighbors measure performs the best in this particular case.

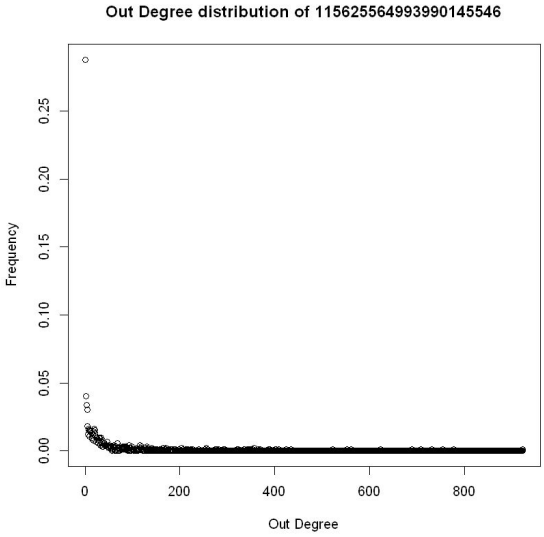
Part2 Google+ network

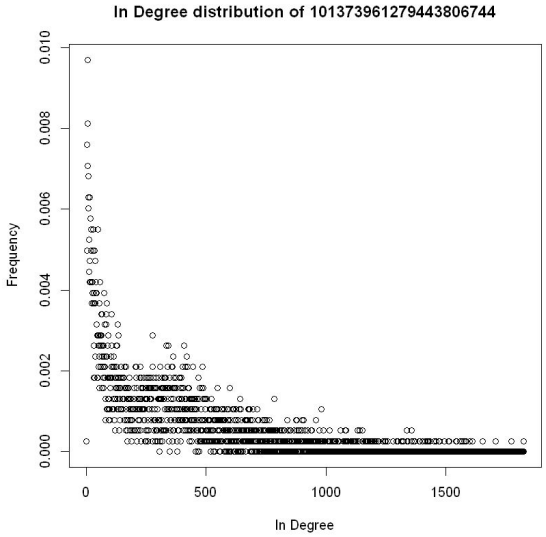
In this part, we are going to introduce directed network and consider the properties of this directed graph. First, we will consider personal network who have more than 2 circles. After counting, we find out there are Q18:57 personal networks that have more than 2 circles. Then we plot the in-degree distribution and out-degree distribution for three nodes: 109327480479767108490, 115625564993990145546 and 101373961279443806744. Q19: Plots are presented below:

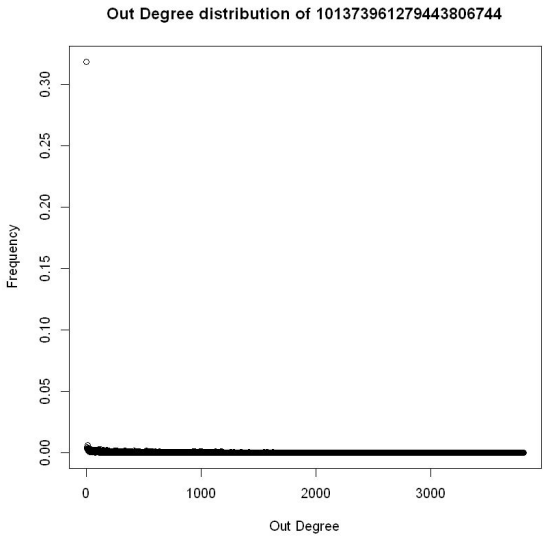

In Degree distribution of 109327480479767108490
Frequency
0.00
0.05
0.10
0.15
0
20
40
60
80
In Degree


In Degree distribution of 115625564993990145546
Frequency
0.000
0.005
0.010
0.015
0.020
0.025
0
50
100
150
200
In Degree


Out Degree distribution of 109327480479767108490
Frequency
0.0
0.1
0.2
0.3
0.4
0.5
0
200
400
600
800
Out Degree


Out Degree distribution of 115625564993990145546
Frequency
0.00
0.05
0.10
0.15
0.20
0.25
0
200
400
600
800
Out Degree





We can observe that these nodes have a similar in-degree and out-degree distributions. Most of the nodes have a low in-degree and only a small portion of the nodes have higher out degree.

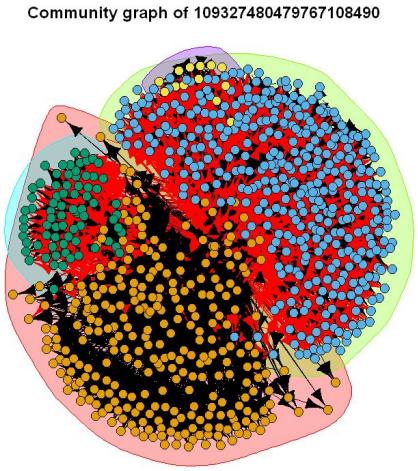
2.1 Community structure of personal networks

In this section, we will explore the community structure of the personal networks of three nodes above. We will first use Walk-trap community detection algorithm to find communities of each node.
Q20: Modularity score are presented in the following table:

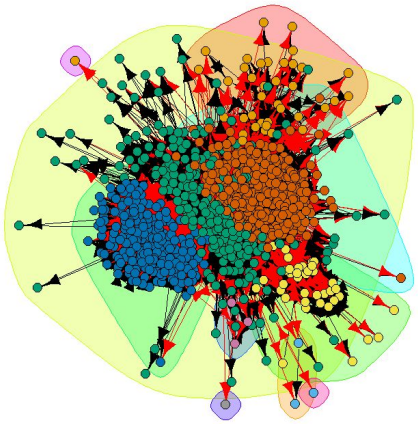
Node ID	N1	N2	N3
score	0.25276	0.31947	0.19109

N1: 109327480479767108490
N2: 115625564993990145546
N3: 101373961279443806744

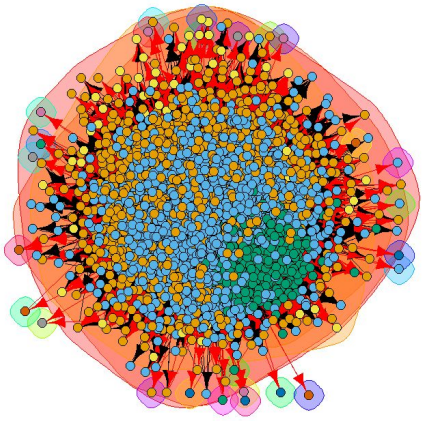
The scores are not similar to each other. Community plots are presented below:



Community graph of 115625564993990145546



Community graph of 101373961279443806744



2.1 Community structure of personal networks

Q21:

homogeneity: the proportion of the difference of the entropy with every circles and the entropy with every circles given its community, and the entropy with every circles. In a nutshell, it measures the score that each community contains only people of a single circle.

completeness: the proportion of the difference of the entropy with every communities and the entropy with every communities given its circles, and the entropy with every communities. In a nutshell, it measures the score that all people of a given circle are assigned to the same community.

Q22:

In this question, we calculate homogeneity and completeness score of nodes we mentioned above.
The results are as follows:
Node ID: 109327480479767108490
Homogeneity Score: 0.851885
Completeness Score: 0.32987

Node ID: 115625564993990145546
Homogeneity Score: 0.45189
Completeness Score: -3.423962

Node ID: 101373961279443806744
Homogeneity Score: 0.0038
Completeness Score: -1.504238

We can see from the result that, in the first node, which id is 109327480479767108490, the homogeneity is high and the completeness is also high and positive compared with the other two nodes. This is because there's only 3 circles and 4 communities in this network, and the node in the social network's circle has more chance to be signed to one single community,which means there's less overlap in this network. Hence this Node network has higher homogeneity and completeness score.
For other two nodes, which ids are 115625564993990145546 and

10137396127944380674, we can see that their homogeneity and completeness decrease sharply even to negative value. This is because the circles and communities are both very big(bigger than the first node), and there's more change that more overlap occur with this circle information. The negative completeness is because the overlap is great and the circle information is not sufficient to cover all the person. Especially, the homogeneity in third node is relatively low, which means there are many circles into one community, hence decrease the homogeneity.