Project 2
Social Network Mining

1. Facebook Network

We obtained our dataset from http://snap.stanford.edu/data/egonets-Facebook.html. The Facebook network data is unzipped from the edgelist file. And we created the Facebook network which was shown in the figure 1. And we also learn the connectivity and degree distribution in this part.

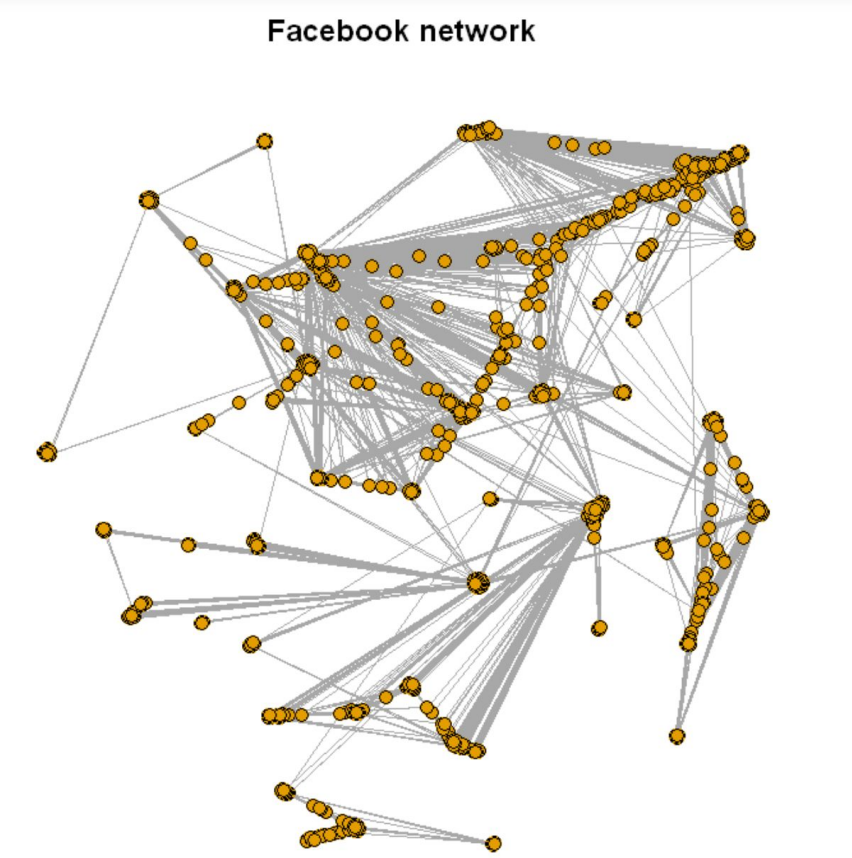


Figure 1. Facebook Network

Question 1

We used the command *is_connected* to test the network, and the result show that the network in connected.

Question 2

We used the command *diameter* to get that the diameter of the network is 8.

Question 3

We plotted the degree distribution in the figure 2.

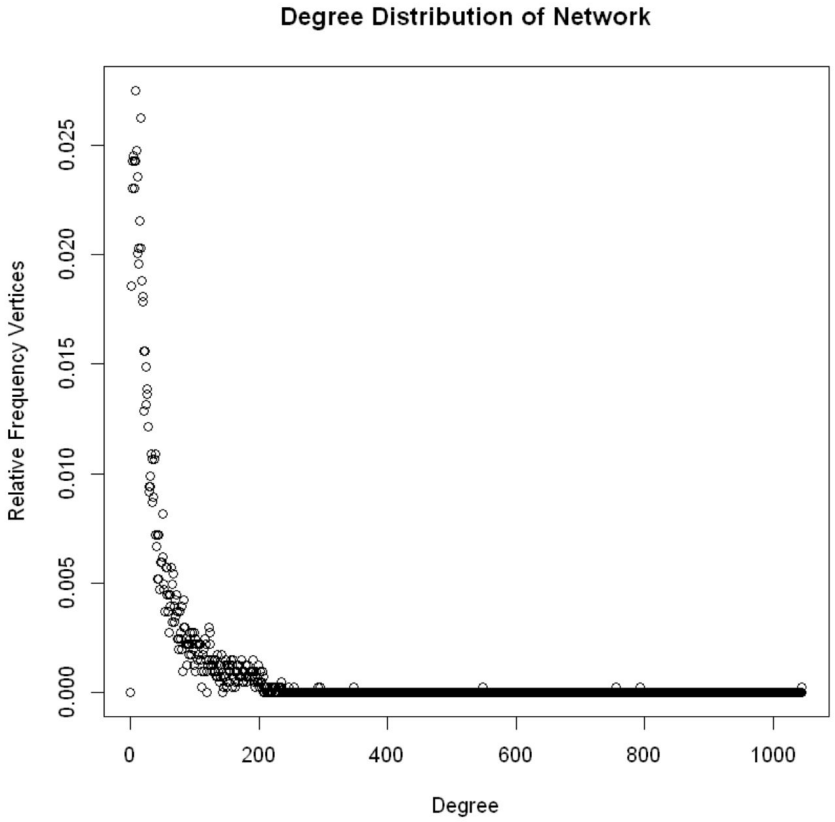


Figure 2. Degree Distribution of Network

From the above graph, we can see that the degree less than 200 corresponds to a relatively higher relative frequency vertices. And nodes are highest when the degree is close to 0. Such a distribution reflects that even though the network is connected as the test shows, people are only connected to their direct relationship like the direct friends of them or friends of their friends.

And the average degree is 522.5 .

Question 4

We plotted the log degree distribution in the figure 3.

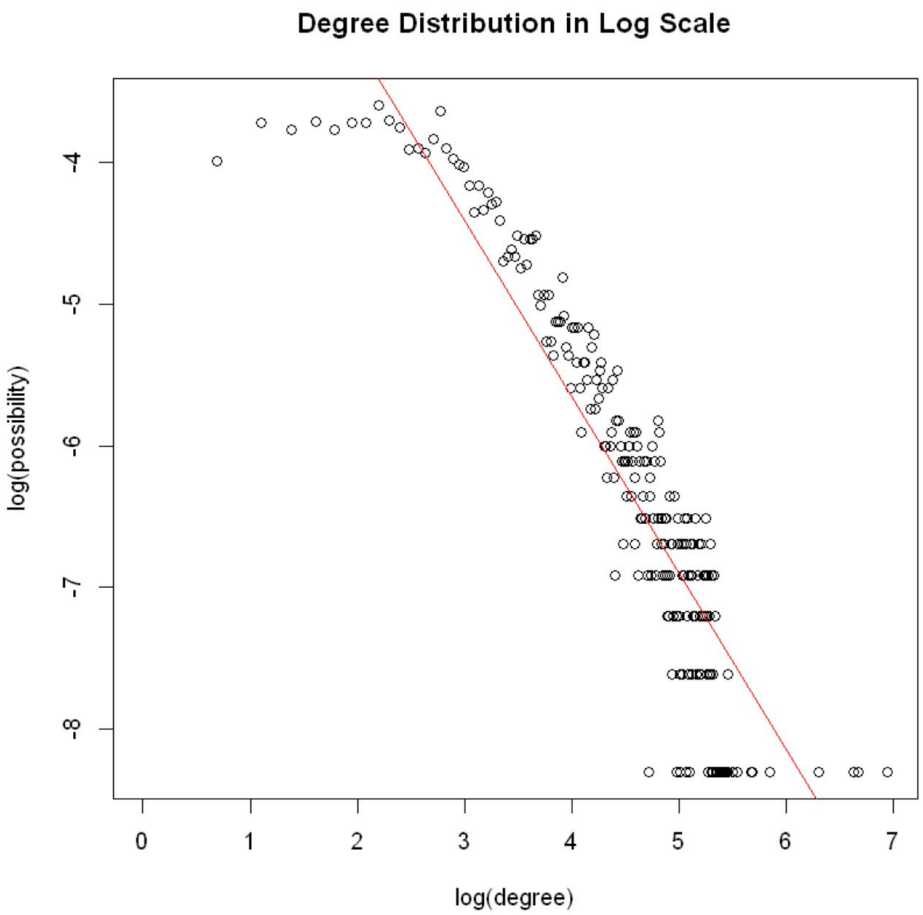


Figure 3. Log Degree Distribution of Network

In order to not get null values, we removes degrees with non-real numbers. And from the above graph, we estimated that the slope is around 1.25 .

2. Personalized network

A personalized network of an user vi is defined as the subgraph induced by vi and it's neighbors. In this part, we will study some of the structural properties of the personalized network of the user whose graph node ID is 1 (node ID in edgelist is 0). From this point onwards, whenever we are refering to a node ID we mean the graph node ID which is 1 + node ID in edgelist.

Question 5

We created a personalized network which is shown in the figure 4.

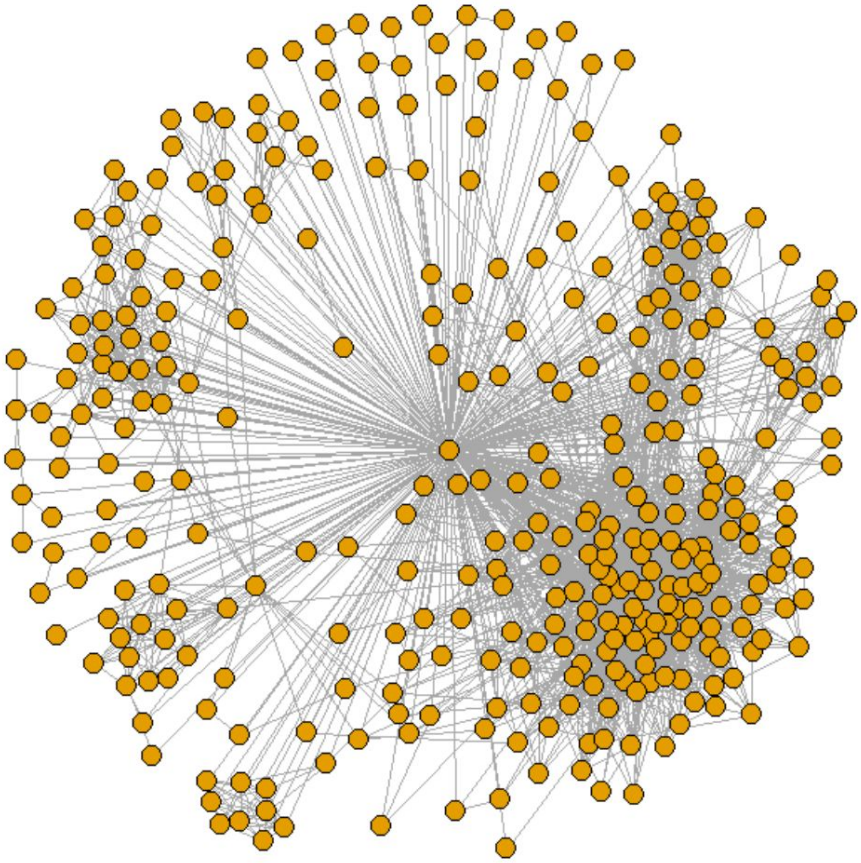


Figure 4. Personalized Network

This personalized network has 348 nodes and 2866 edges.

Question 6

The diameter of the personalized network is 2. A trivial upper bound for the diameter of the personalized network is 2 and a trivial lower bound is 1.

Question 7

A trivial upper bound which is equaling to 2 means a divergent network -some nodes connected to the core node but not all other nodes. In other words, a code can connect to a third node through the core code. And a trivial lower bound which is equaling to 1 means that the every node is directly connected to another node. In other words, people in this network knows each other.

3. Core node's personalized network

A core node is defined as the nodes that have more than 200 neighbors. For visualization purpose, we have displayed the personalized network of a core node below.

Question 8

There are 40 cored nodes and the average degree is 279.375.

3.1. Community structure of core node's personalized network

In this part, we study the community structure of the core node's personalized network. To be specific, we will study the community structure of the personalized network of the following core nodes:

- Node ID 1
- Node ID 108
- Node ID 349
- Node ID 484
- Node ID 1087

Question 9

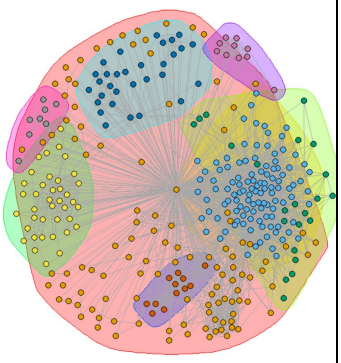
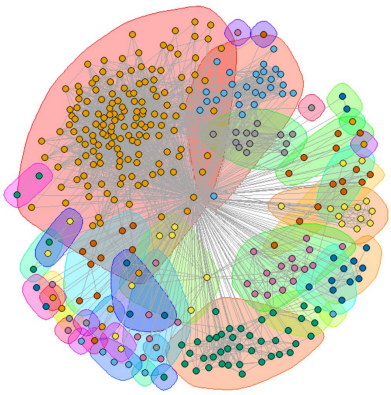
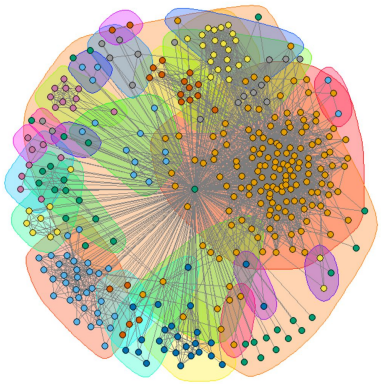
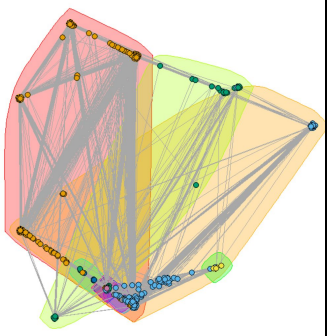
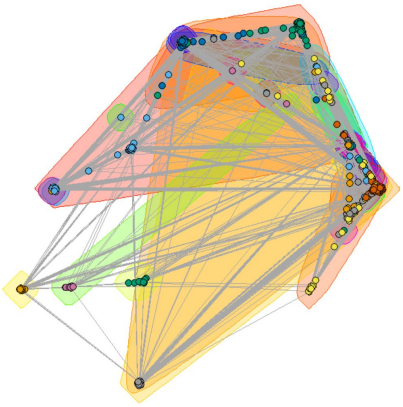
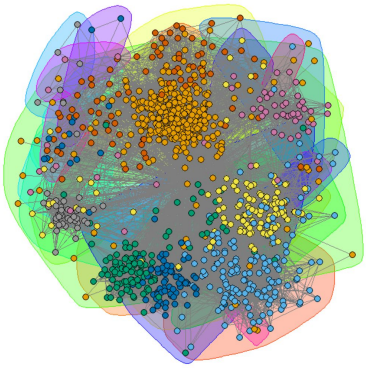
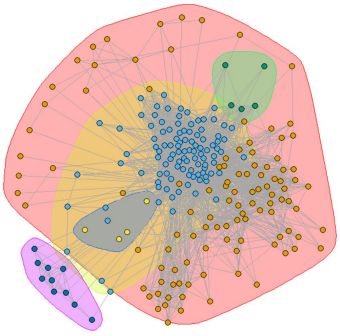
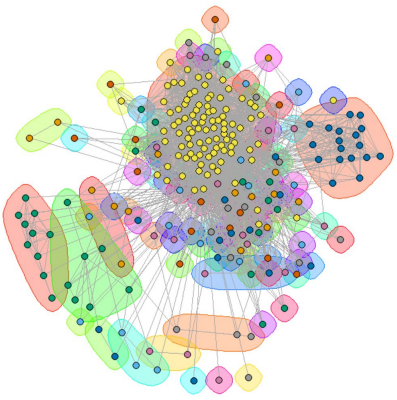
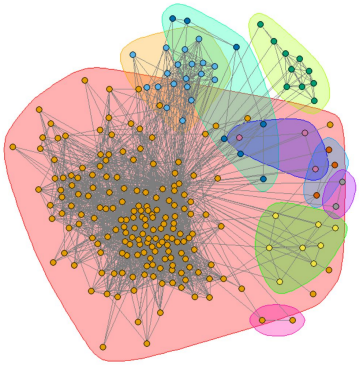
For each of the above core node's personalized network, find the community structure using Fast-Greedy, Edge-Betweenness, and Infomap community detection algorithms. Compare the modularity scores of the algorithms. For visualization purpose, display the community structure of the core node's personalized networks using colors. Nodes belonging to the same community should have the same color and nodes belonging to different communities should have different color. In this question, you should have 15 plots in total.

Fast greedy algorithm stops when the modularity can't increase anymore since this method assumes every vertex only belongs to a separate community.Edge-Betweenness is the number of shortest paths between pairs of vertices that run along them. And then, the algorithm will remove the highest one persistently until it doesn't exist. Infomap gets modularity with flows. More circles in a graph lead to a larger modularity scores since higher probability of a node goes back randomly.

The community structures are shown in the table 1 and the modularity scores are shown in the table 2.

Table 1: Community structures for core nodes using various community detection algorithms

ID	Fast-Greedy	Edge-Betweenness	Infomap

1	Fast-Greedy, core node 1 	Edge Betweenness, core node 1 	Infomap, core node 1 
108	Fast-Greedy, core node 108 	Edge Betweenness, core node 108 	Infomap, core node 108 
349	Fast-Greedy, core node 349 	Edge Betweenness, core node 349 	Infomap, core node 349 

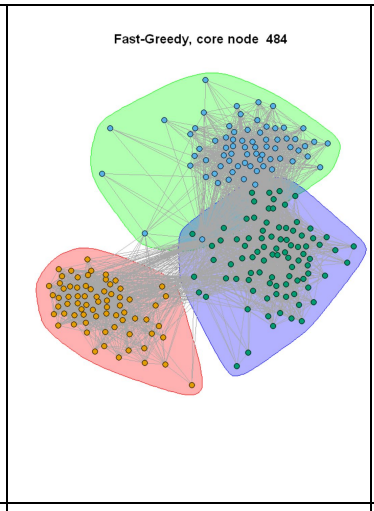
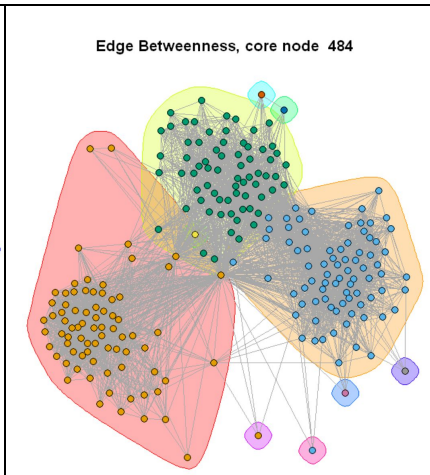
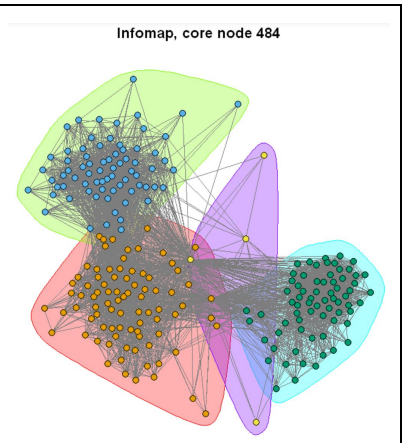
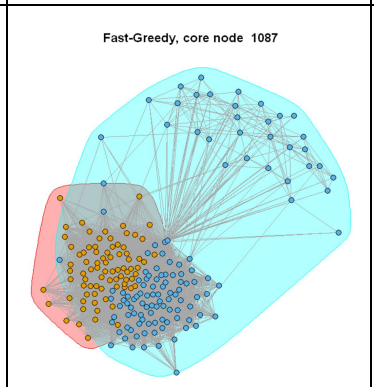
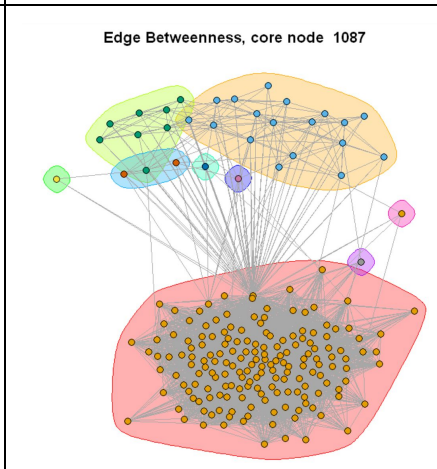
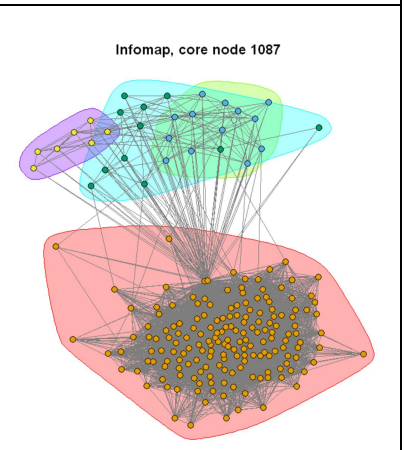
484			
1087			

Table 2: Modularity Scores for core nodes using various community detection algorithms

ID	Fast-Greedy	Edge-Betweenness	Infomap
1	0.4131	0.3533	0.3891
108	0.4360	0.5068	0.5082
349	0.2503	0.1335	0.0955
484	0.5070	0.4891	0.5153
1087	0.1455	0.0276	0.0270

From the above table, we can know that the Fast_Greedy has the highest modularity scores in different ID. And for fast-greedy algorithm, when ID is 484, the modularity score can reach 0.5070. Which means the it's easiest for Fast-Greedy algorithm to create community with

ID=484 and it's hardest to create community with ID=1087. And Edge-Bestweenness and Infomap;s modularity scores are close.

3.2. Community structure with the core node removed

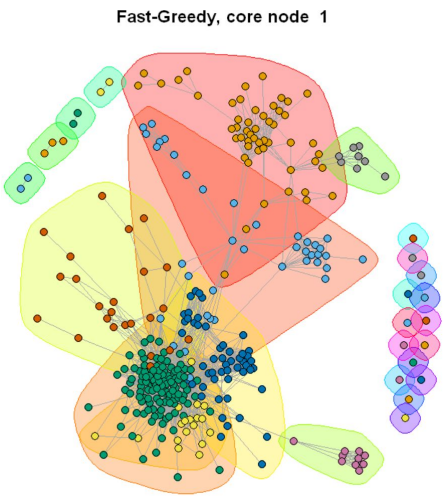
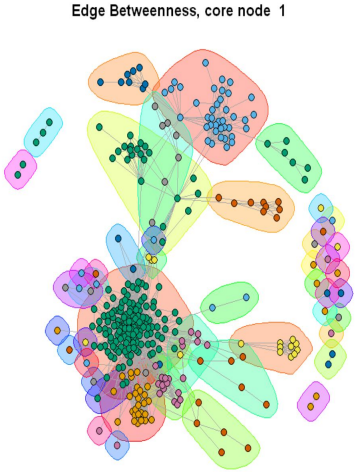
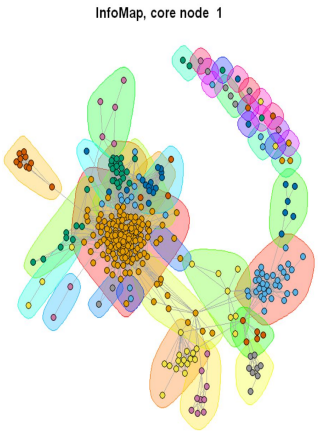
In this part, we will explore the effect on the community structure of a core node's personalized network when the core node itself is removed from the personalized network.

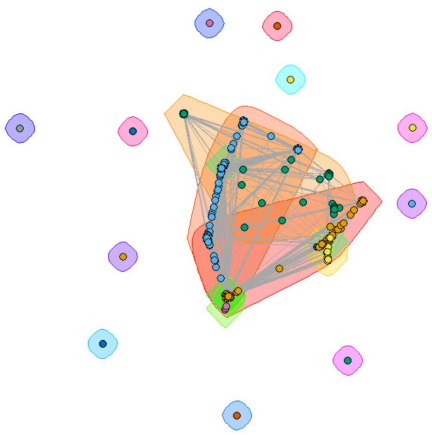
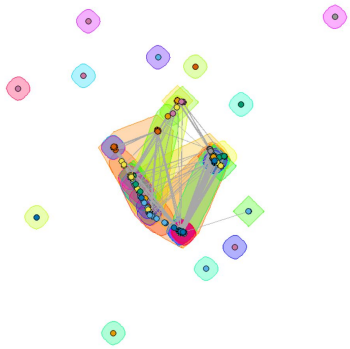
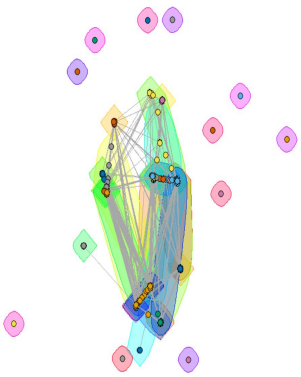
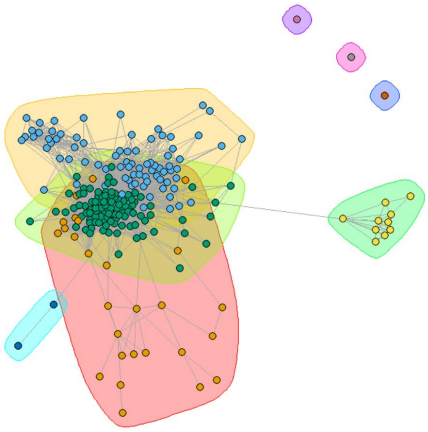
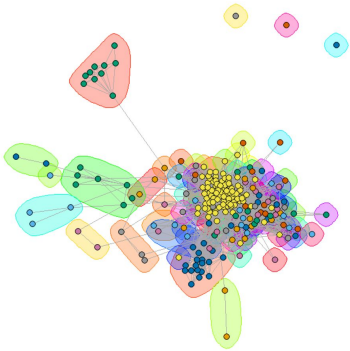
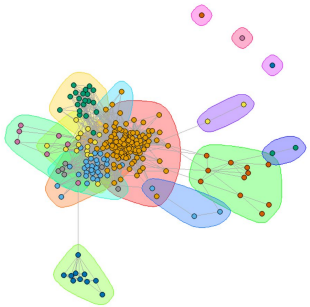
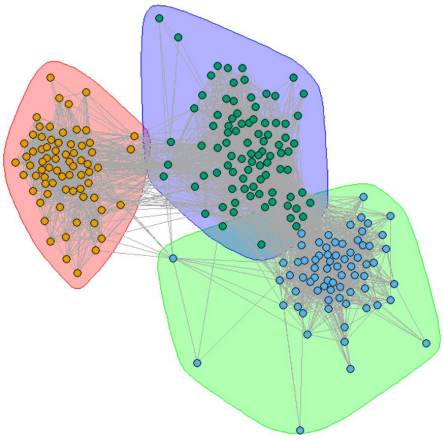
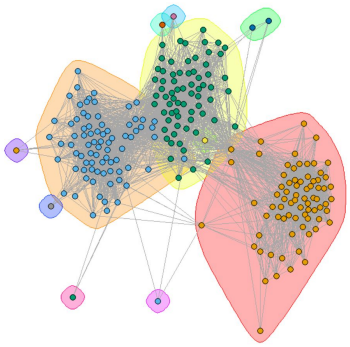
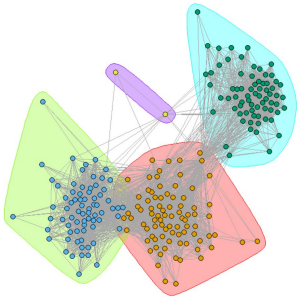
Question 10

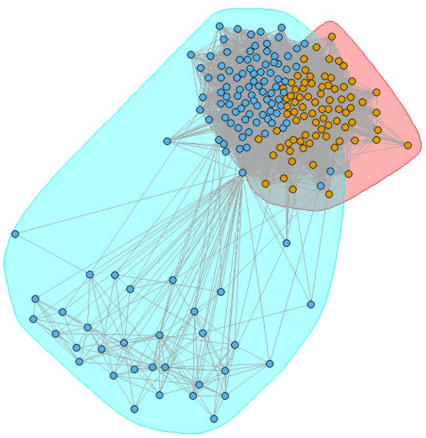
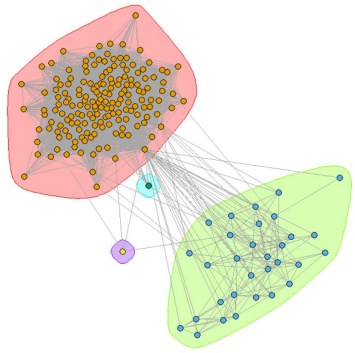
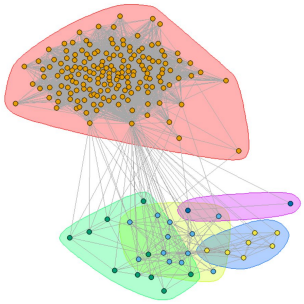
For each of the core node's personalized network (use same core nodes as Question 9), remove the core node from the personalized network and find the community structure of the modified personalized network. Use the same community detection algorithm as Question 9. Compare the modularity score of the community structure of the modified personalized network with the modularity score of the community structure of the personalized network of Question 9. For visualization purpose, display the community structure of the modified personalized network using colors. In this question, you should have 15 plots in total.

We removed the original user in Question 9. The community structures can be seen below in Table 3.

Table 3. Community structures for core nodes using various community detection algorithms with code node removed

ID	Fast-Greedy	Edge-Betweenness	Infomap
1			

108	Fast-Greedy, core node 108 	Edge Betweenness, core node 108 	InfoMap, core node 108 
349	Fast-Greedy, core node 349 	Edge Betweenness, core node 349 	InfoMap, core node 349 
484	Fast-Greedy, core node 484 	Edge Betweenness, core node 484 	InfoMap, core node 484 

1087			

We recorded the modularity scores in Table 4.

Table 4. Modularity Scores for core nodes using various community detection algorithms with code node removed

ID	Fast-Greedy	Edge-Betweenness	Infomap
1	0.4419	0.4161	0.4180
108	0.4581	0.5213	0.5186
349	0.2457	0.1506	0.2448
484	0.5342	0.5154	0.5434
1087	0.1482	0.0325	0.0274

Comparing table 2 and table 4, we found the modularity scores with code node removed increased a little bit and the difference is small. It makes sense since the nodes connected only to the core nodes are separated with the rest of the network which makes themself communities. Thus, it's difficult to cleanly set nodes to communities if we classify the core nodes.

3.3 Characteristic of nodes in the personalized network

Question 11

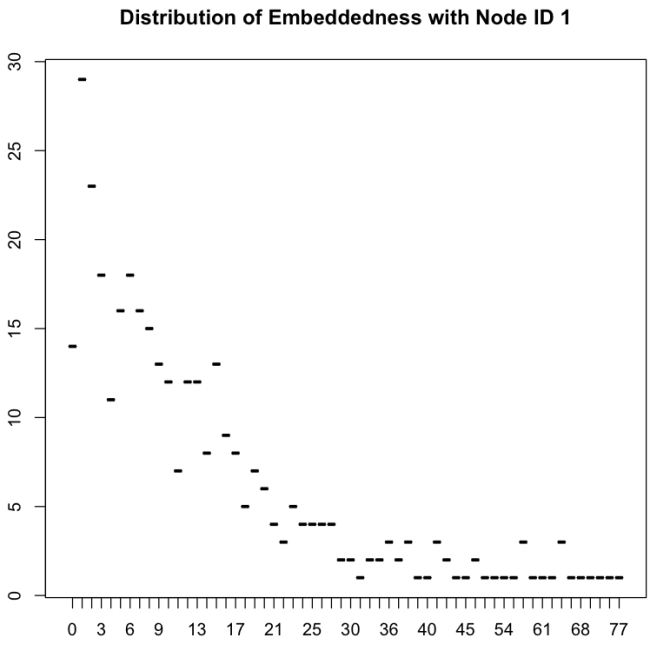
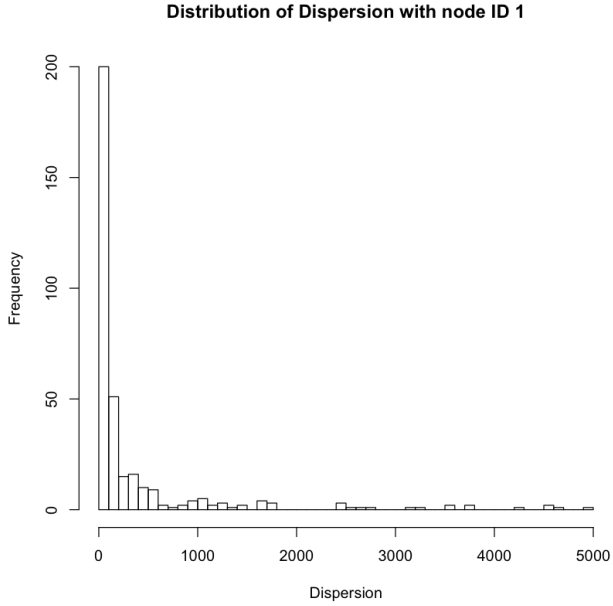
Embeddedness of a node is defined as the number of mutual friends a node shares with the core node. In the personalized network, for each core node, all other nodes are neighbors of this core node. Therefore, the number of mutual friends of a node with the core node includes all the neighbors except for the core node. As a result, the Embeddedness can be expressed as :

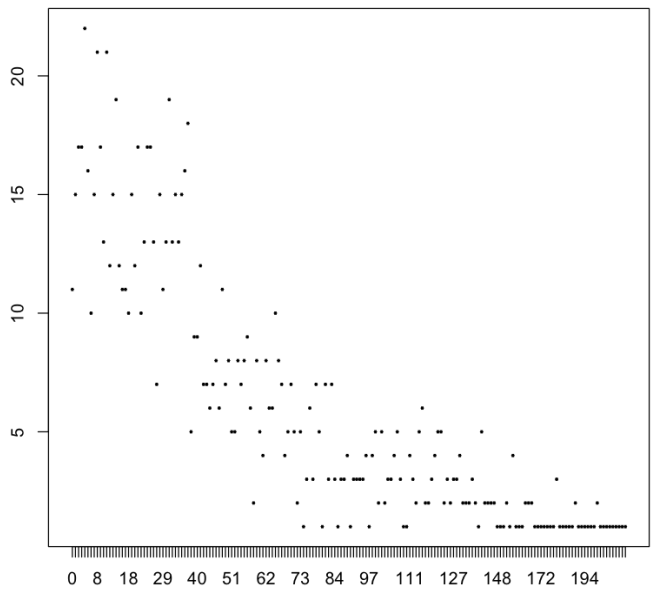
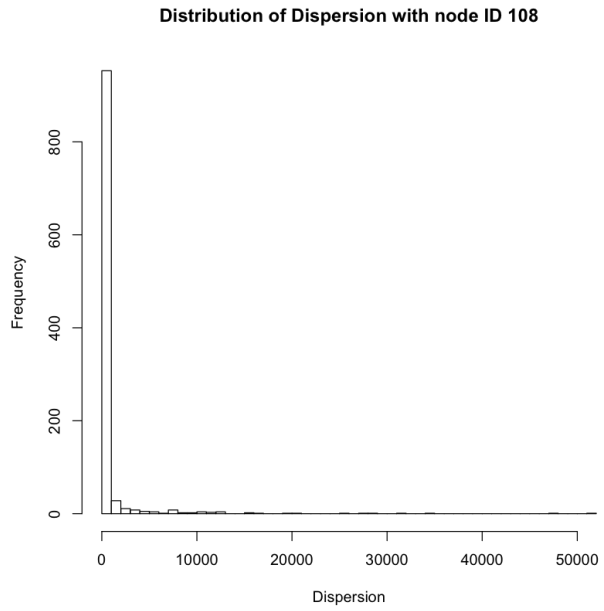
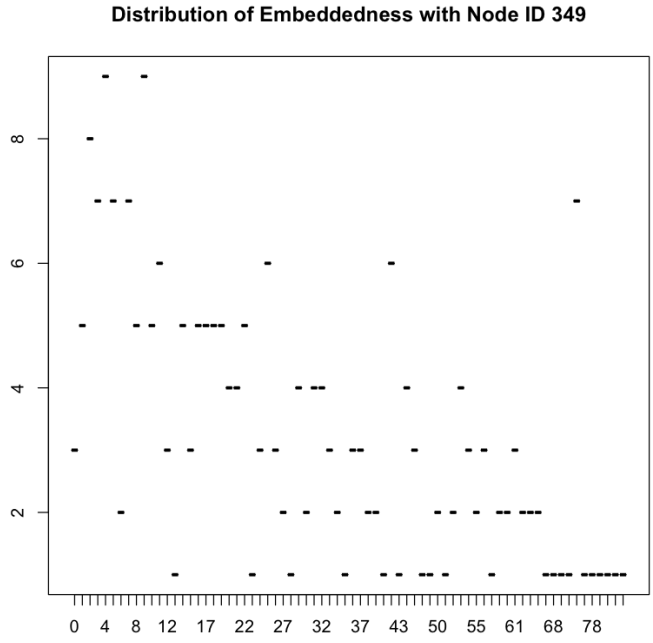
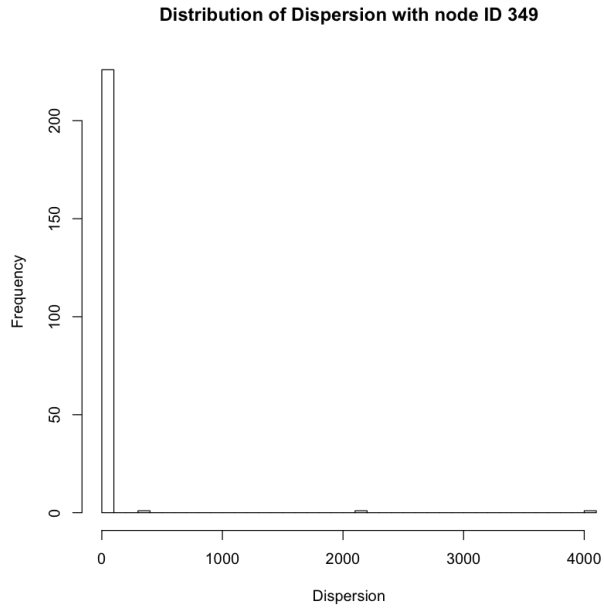
Embeddedness(graph, node) = degree(node) - 1

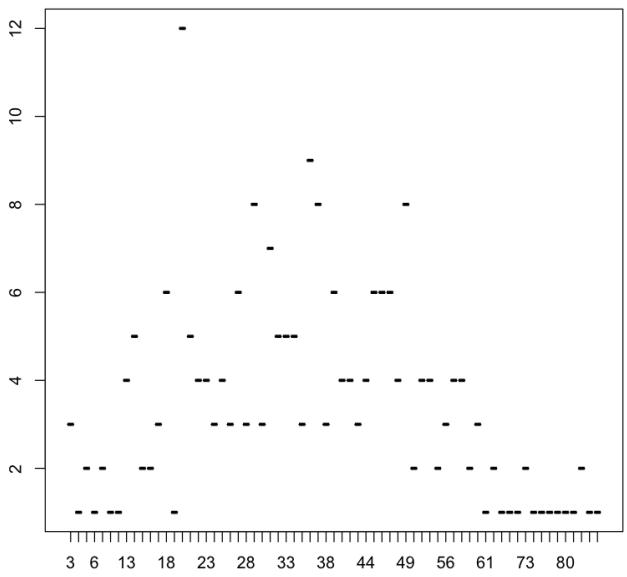
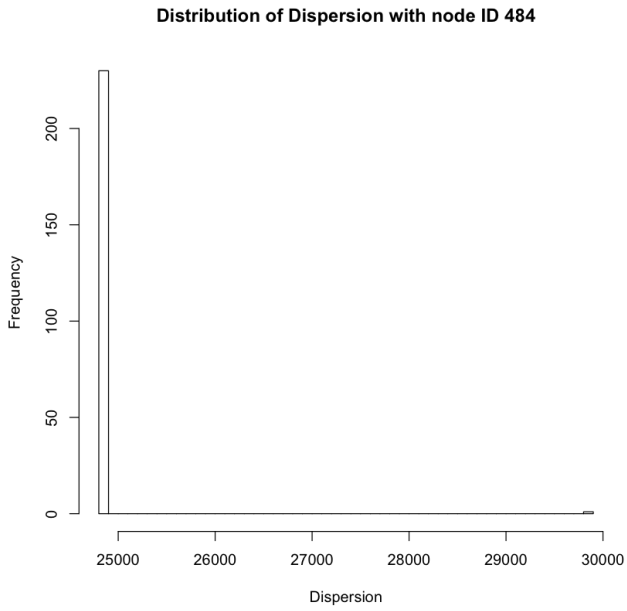
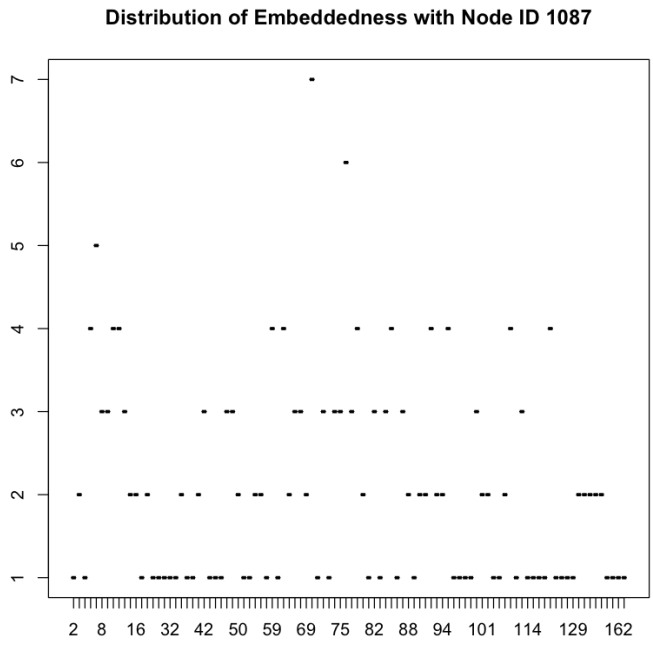
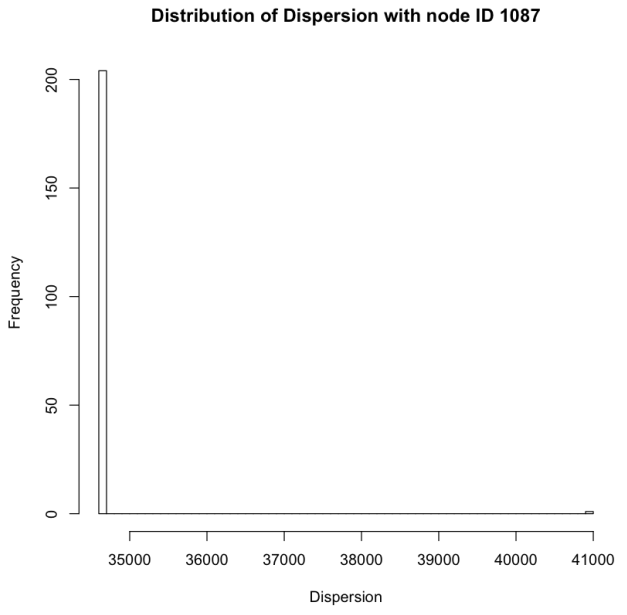
Question 12

For each core nodes (used in Question 9), we analyze its distribution of embeddedness and dispersion. We calculate embeddedness using the expression above. As for the dispersion, we need to consider several corner cases. When the number of mutual friends of a node is no more than 1, the dispersion is set to 0. Moreover, when the distance of two nodes in mutual friends not connected in the graph, with core node and current node deleted, is infinite, we set the bound with the diameter of current personalized network and a constant to avoid infinity dominating. As result, the plots of the distribution of embeddedness and dispersion with different core nodes are shown below:

Table 5. Distribution of Embeddedness and Dispersion with 5 core nodes

	Distribution of Embeddedness	Distribution of Dispersion
Node ID 1		

Node ID 108		
Node ID 349		

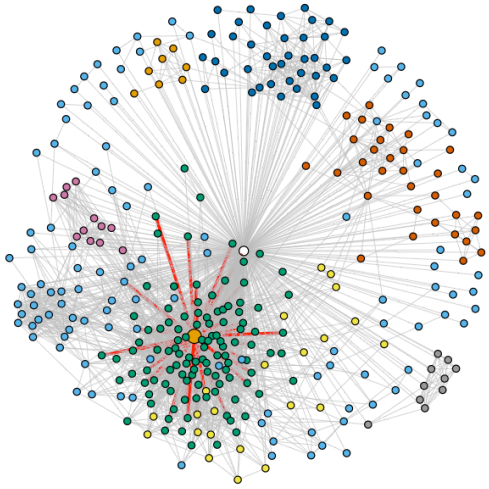
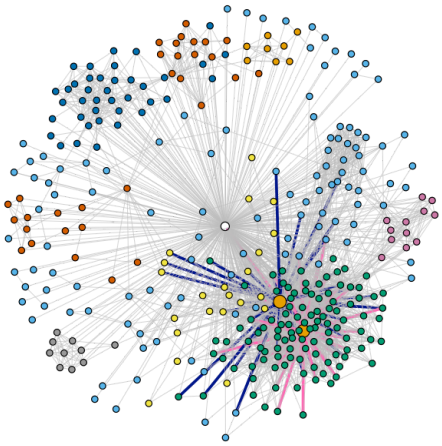
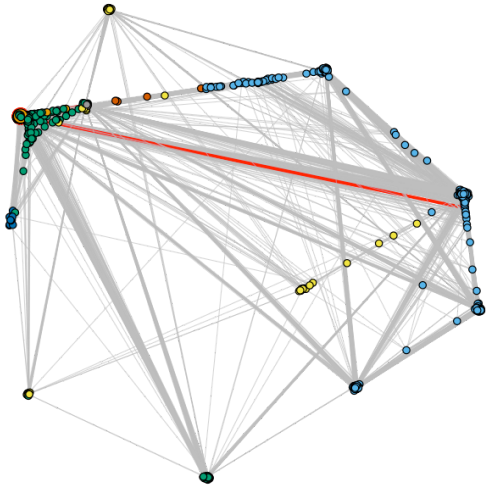
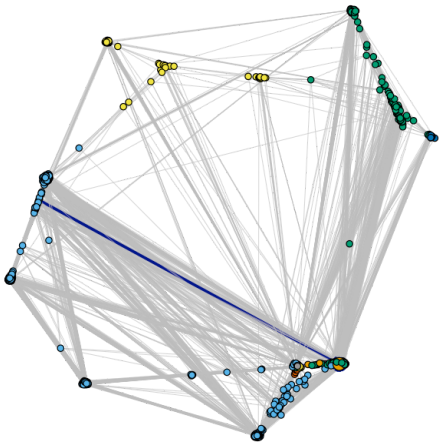
Node ID 484		
Node ID 1087		

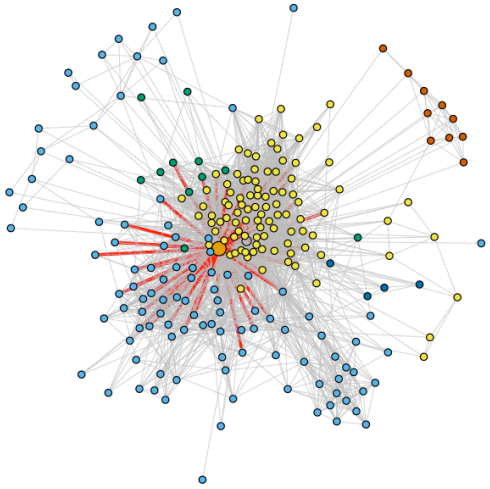
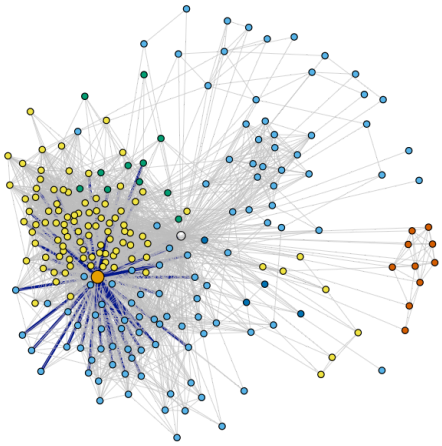
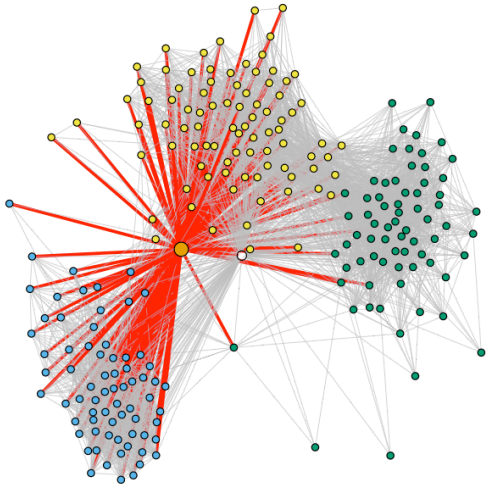
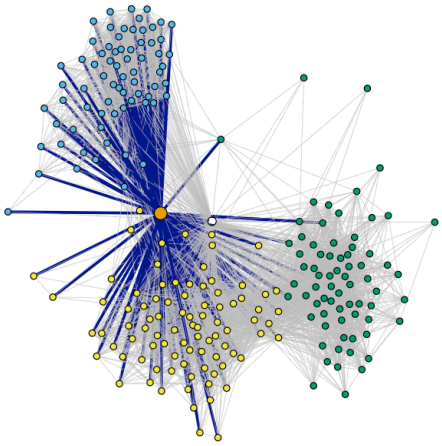
Question 13 & 14

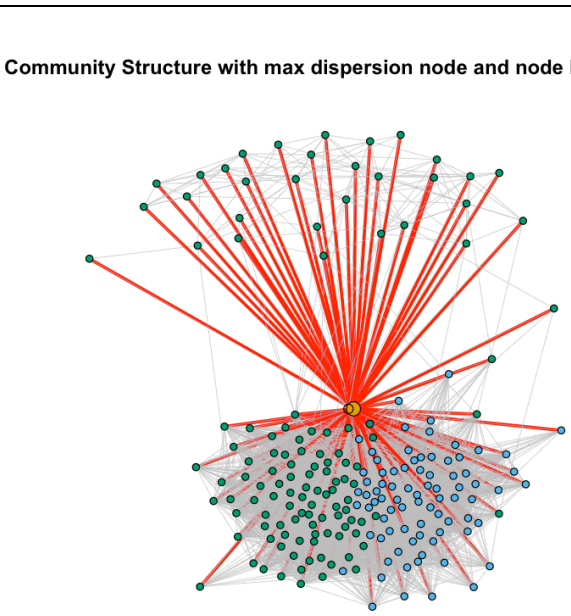
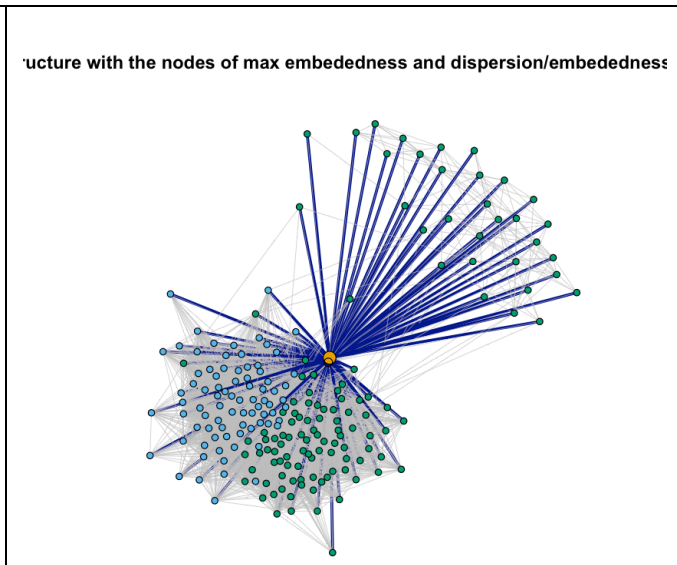
We required to plot community structure of the personalized network for each core node, using colors and highlight the node with maximum dispersion, maximum embeddedness, and maximum $\frac{dispersion}{embeddedness}$. To detect community structure, the Fast-Greedy algorithm is used. In the community structure with maximum dispersion node, this node is highlighted “orange” and its edges are “red” and the core node is “white”. In the community structure with maximum embeddedness node and maximum dispersion/embeddedness node, the node with maximum embeddedness is highlighted in orange with “hotpink” edges, and the node with maximum $\frac{dispersion}{embeddedness}$ node is highlighted in orange with “darkblue” edges.

Table 6. Community Structure with max dispersion node, embeddedness node and dispersion/embeddedness nodes

	Community Structure with max dispersion node	Community Structure with max embeddedness & $\frac{dispersion}{embeddedness}$ nodes

Node ID 1	Community Structure with max dispersion node and node ID 1 	tructure with the nodes of max embededness and dispersion/embededne 
Node ID 108	Community Structure with max dispersion node and node ID 108 	ructure with the nodes of max embededness and dispersion/embedednes 

Node ID 349	Community Structure with max dispersion node and node ID 349 	ructure with the nodes of max embededness and dispersion/embedednes 
Node ID 484	Community Structure with max dispersion node and node ID 484 	ructure with the nodes of max embededness and dispersion/embedednes 

Node ID 1087		

Question 15

Community Structure with max dispersion node:

Dispersion of a node is defined as the sum of distances between every pair of the mutual friends the node shares with the core node. The two nodes with high dispersion means that these two nodes share numerous mutual neighbors, that is, in real social life, two friends share a large number of mutual friends. However, the commo friends shared by the two nodes, the current target node and the core node in our case, have low connectivity from different circles.

Community Structure with max embeddedness node:

Embeddedness of a node is defined as the number of mutual friends a node shares with the core node. The node with maximum embeddedness has high strength with the core node owing to numerous mutual friends shared between them. In the personalized network, such node with maximum embeddedness is the node with maximum degree excluding the core node. Moreover, from the plots, we can also find that the node with maximum embeddedness usually from largest communities where the nodes have high connectivity.

Community Structure with max $\frac{dispersion}{embeddedness}$ nodes:

From the plots, we are able to observe that in most plots, the core node is the same node with maximum $\frac{dispersion}{embeddedness}$ node. Maximum $\frac{dispersion}{embeddedness}$ means high dispersion with low embeddedness. By normalizing embeddedness, this is better to indicate the strength between two people shared with mutual communities rather than mutual friends. Although certainly that share mutual friends, these friends may come from many different communities. In real social network, this value shows that two people not only know each other, but also share common groups with more common features to build a deeper relation, that is, to detect how likely two people can be in romantic relation. The higher this ratio is, the deeper romantic relation of the two nodes are.

Last but not least, from these plots of such three measurements, we can see that nodes with maximum dispersion, embeddedness and $\frac{dispersion}{embeddedness}$ are generally with maximum degree. The results can be shown in Node ID 108, 349, 484 and 1087. As for Node ID 1, the distance distribution of mutual friends for each node in the modified network is different, which may lead to the node with maximum embeddedness and the one with maximum $\frac{dispersion}{embeddedness}$ are different, but such nodes still keep high degree.

4. Friend recommendation in personalized networks

4.3. Creating the list of users

Question 16

We are required to apply friend recommendation procedure to the personalized network of Node ID 415. First of all, we create a subgraph with all the neighbor nodes of Node ID 415, and then build the list of users who we want to recommend new friends to, picking all nodes with degree 24 of course. As a result, Nr is the user list, and **|Nr| = 11** is its length.

4.4. Average accuracy of friend recommendation algorithm

Question 17

we need to compute the average accuracy for user i in the list Nr, that is, compute it by iterating over the following steps 10 times and then taking the average:

1. Remove each edge of node i at random with probability 0.25. In this context, it is equivalent to deleting some friends of node i. Let's denote the list of friends deleted as Ri

2. Use one of the three neighborhood based measures to recommend |Ri| new friends to the user i. Let's denote the list of friends recommended as Pi

3. The accuracy for the user i for this iteration is given by $\frac{|Pi \cap Ri|}{|Ri|}$

By iterating over the above steps for 10 times and then taking the average gives us the average accuracy of user i. In this manner, we compute the average accuracy for each user in the list Nr. Once we have computed them, then we can take the mean of the average accuracies of the users in the list Nr. The mean value will be the average accuracy of the friend recommendation algorithm.

The accuracy results of these three algorithms are shown below:

Table 7. Accuracy results of three friend recommendation algorithms

	Common Neighbors measure	Jaccard measure	Adamic Adar measure
Accuracy	0.848678669815033	0.850299422799423	0.850590217862945

According to the table, we can observe that all the three algorithm have high average accuracies, and Adamic Adar does best.

Common neighbors captures the idea that two strangers who have a friend in common are more likely to be introduced than those who don't have any friends in common. Consequently, the accuracy can be low when there are many famous nodes in the personalized network. An unknown user may be recommended to another user because of similar famous nodes they are following

Jaccard algorithm measures similarity between two neighbor sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. The Jaccard algorithm can work out the similarity between two neighbor sets, and it reduces the influence of many famous nodes in a personalized network.

Adamic Adar algorithm is a measure used to compute the closeness of nodes based on their shared neighbors. It is based on the concept that common elements with very large neighbourhoods are lesser significant when predicting a connection between two nodes compared with elements shared between a small number of nodes. Therefore, in our personalized network case, this algorithm is the best.

Part 2

Question 18

In this problem, we checked all the ".circle" files and find the files with more than two lines. After coding, we found that the total number of personal network is 57.

Question 19

For the 3 personal networks (node ID given below), plot the in-degree and out- degree distribution of these personal networks. The results are shown in the following figures.

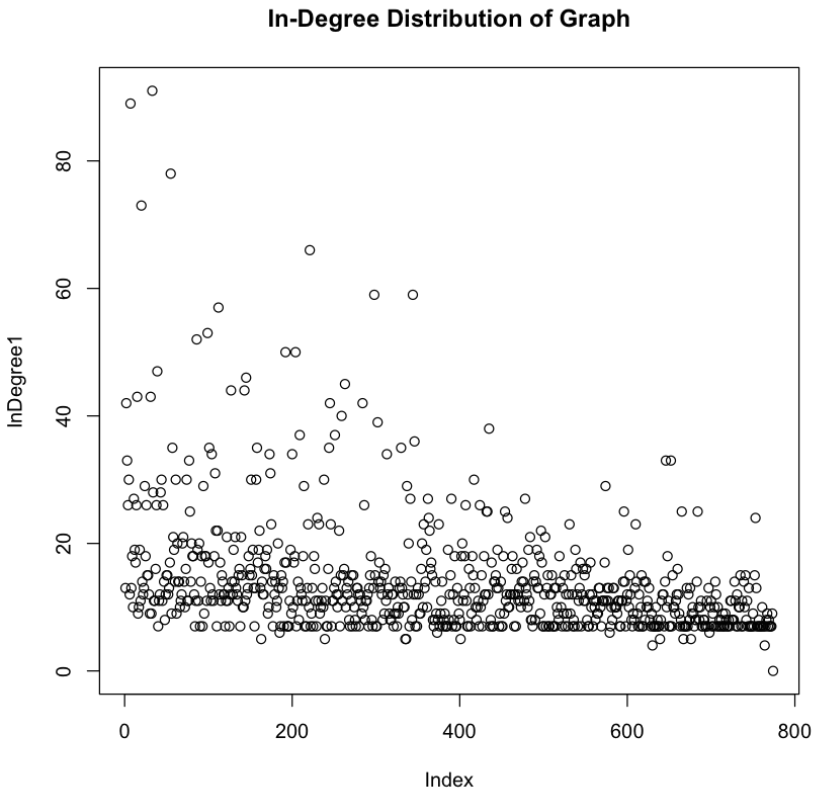


Figure 5. Indegree Distribution for node "109327480479767108490"

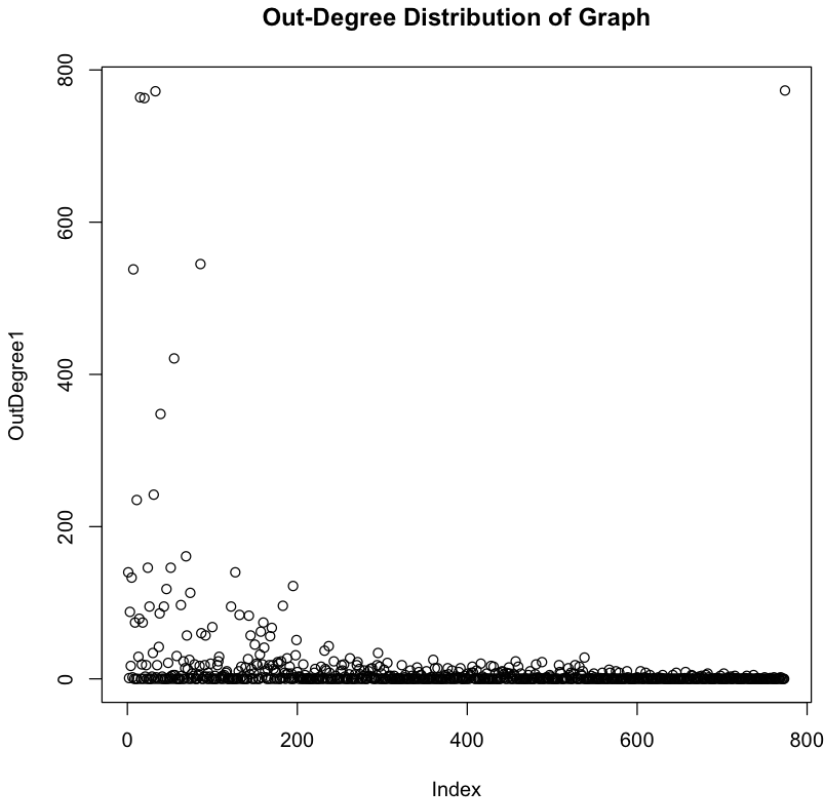


Figure 6. Outdegree Distribution for node "109327480479767108490"

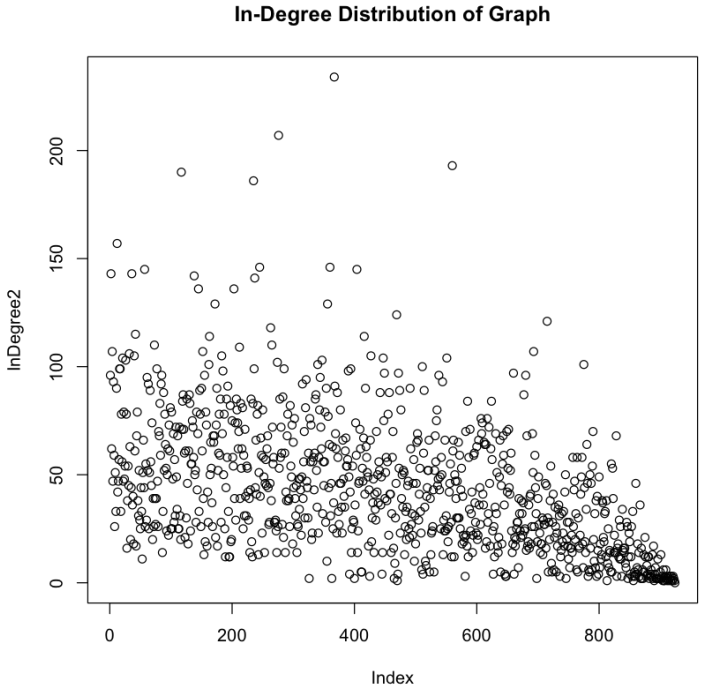


Figure 7. In-degree Distribution for node "115625564993990145546"

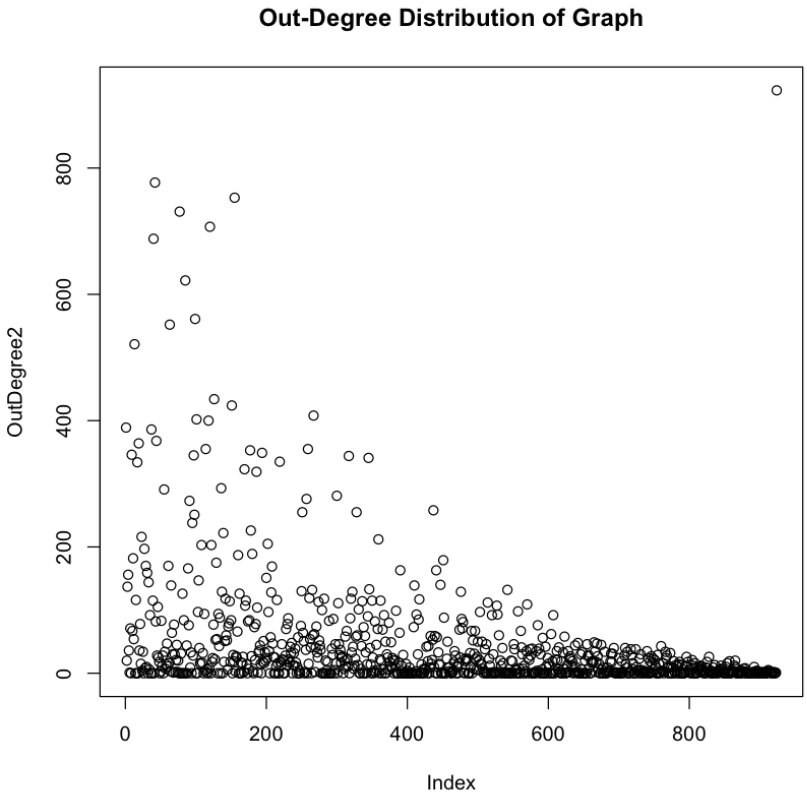


Figure 8. Out-degree Distribution for node "115625564993990145546"

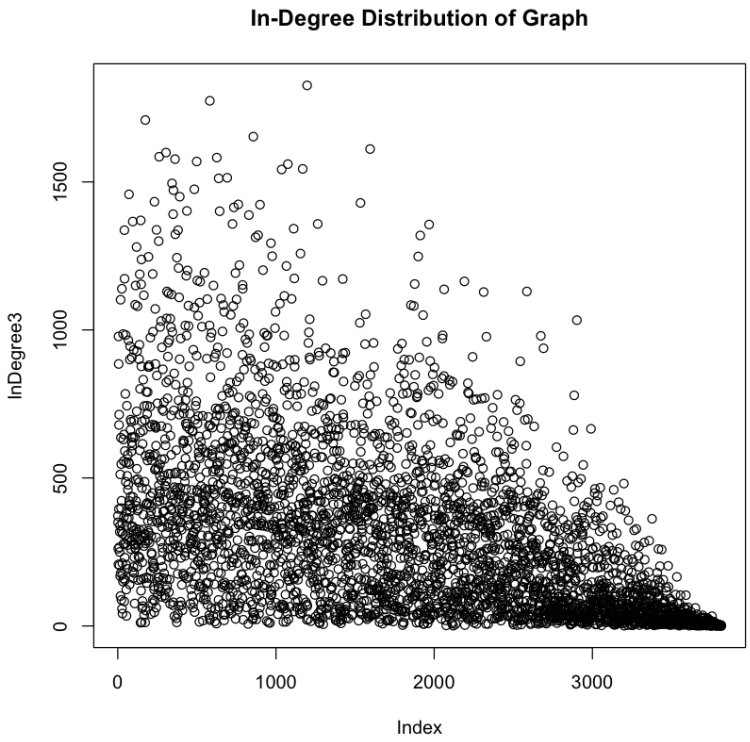


Figure 9.In-degree Distribution for node "10137396127944380674 4"

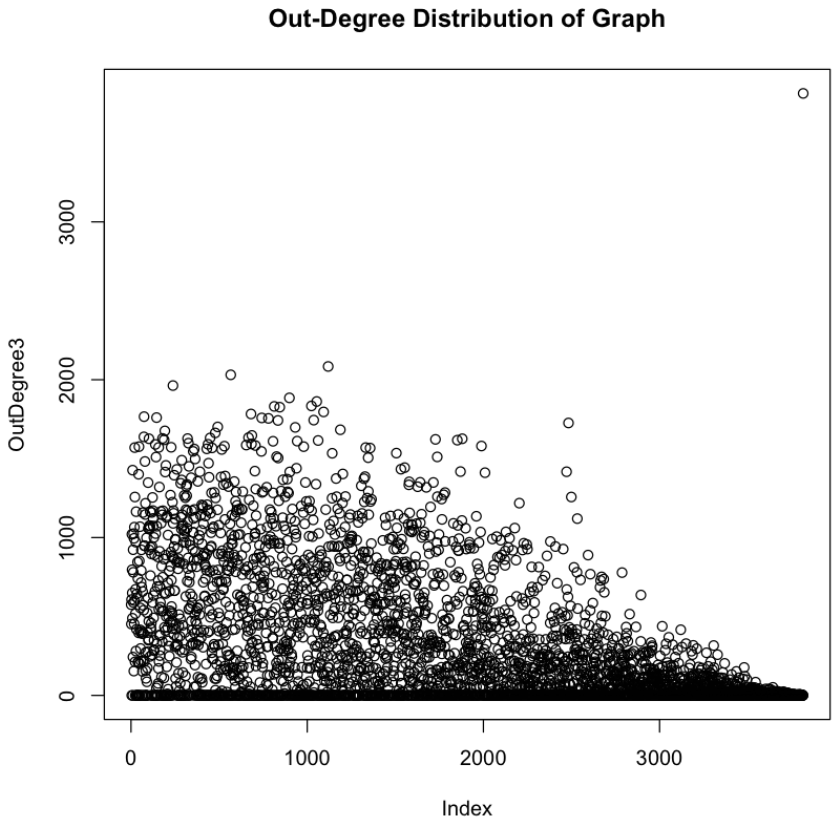


Figure 10. Out-degree Distribution for node "101373961279443806744"

From the above 6 figures, we can find that the nodes in the third network have the largest indegree and outdegree values. The second maximum is the second network. And the first network has the lowest number on the indegree and outdegree values. Therefore, the third graph is the most complex one with strong connections. And the first graph has the least strong connection. Also, these three personal networks have similar out-degree distribution. As for the in-degree distribution, we find that in node 1 and node 2, only small number of member have high in-degrees. And the in-degree changes more like exponentially. However, for node 3, the change is much slower. It is like linear. And there is a large number of members in node 3 with high in-degree value, indicating that users are closer.

Question 20

For the 3 personal networks picked in Question 19, we extracted the community structure of each personal network using Walktrap community detection algorithm. The results for the modularity score is:

node "109327480479767108490": 0.252765387296677
node "115625564993990145546": 0.319472551345825
node "101373961279443806744": 0.191090270876884

From the above results, we found that there is no similarity between modularity scores.

Then we plotted the communities using colors shown in Figure xx to Figure xx.

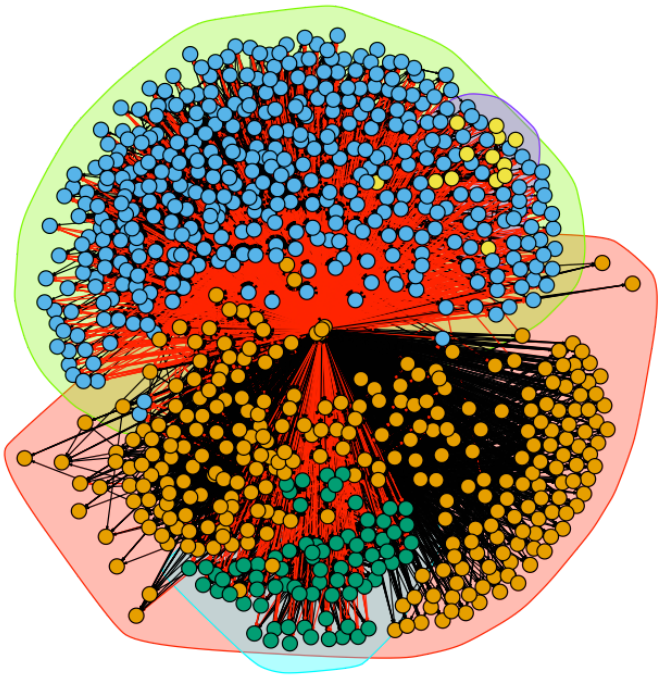


Figure 11. Community in node "109327480479767108490"

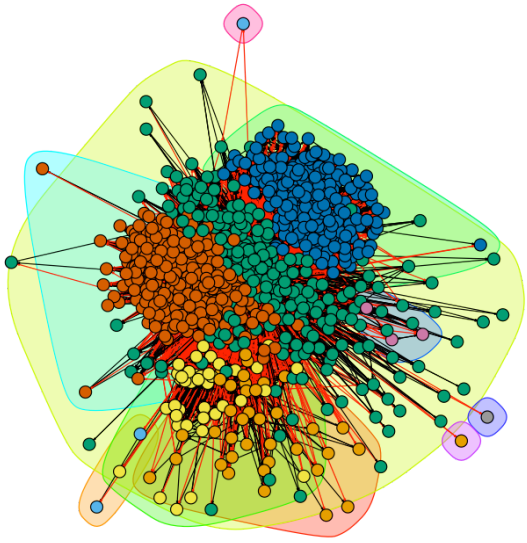


Figure 12. Community in node "115625564993990145546"

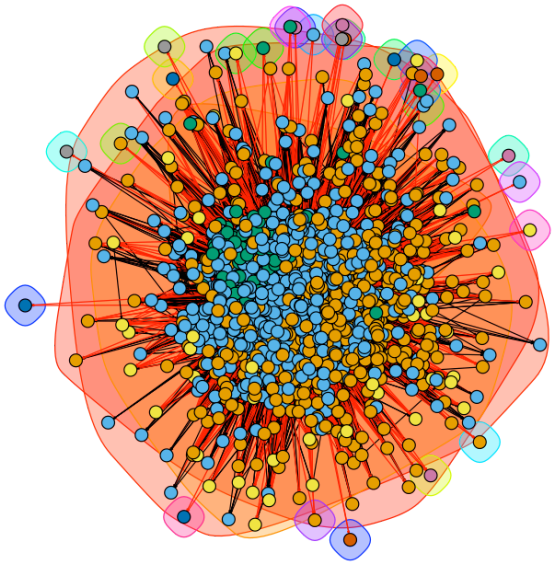


Figure 13. Community in node "101373961279443806744"

Question 21

In our understanding, homogeneity is used to measure the variance of users in terms of the circles they belong to in the community. If all the members in the community come from same circle, then the homogeneity is high. However, if users are equally distributed in all circles, then the value of homogeneity is low.

Completeness is for describing the variance of users in the same circle. If all users in the same circle are in the same community, then the completeness score is high. If users in the circle are from all different communities uniformly, then the score is low.

Question 22

We computed homogeneity (h) and completeness (c) for all three nodes. And here are the results:

Node	h	c
"109327480479767108490"	0.851885115440867	0.329873913536689
"115625564993990145546"	0.451890303032235	-3.4239623491117
"101373961279443806744"	0.0038667069813052	-1.5042383879479

Node 109327480479767108490 has the highest homogeneity values, and node 101373961279443806744 has the lowest. It indicates that in each community of Node 109327480479767108490, there is almost only one label in the cluster. For node 115625564993990145546, the members belong to multiple circle. As for node 101373961279443806744, the value is very close to 0, indicating that the users in the community are from many different circles.

As for completeness, these two nodes, node 115625564993990145546 and node 101373961279443806744 have negative completeness value. This is because a node can belongs to more than one circles, leading to large punishment of the score more than rewarding. Node 109327480479767108490 has small positive completeness. The nodes from the same circle form different communities.