

EE232E Project 2

Social Network Mining

Hongyan Gu – 205025476

Xin Liu – 505037053

Aoxuan (Douglas) Li - 905027231

Jiawei Du - 404943853

Part I Analysis of Facebook Network

In this part, we studied an undirected social network from Facebook and analyzed various characters of the network.

1.1 Structural properties of the facebook network

In this section, we looked into the connectivity and degree distribution of Facebook network.

Question 1

We constructed a network based on the edge list, and plotted the whole network in Fig. 1. We also verified by checking `is.connected()` function, the result indicates that the network is connected, and that there exists a path between every pair of vertices in this network. Every person in the network somehow has correlation with each other, either by knowing directly or introduced by other friends.

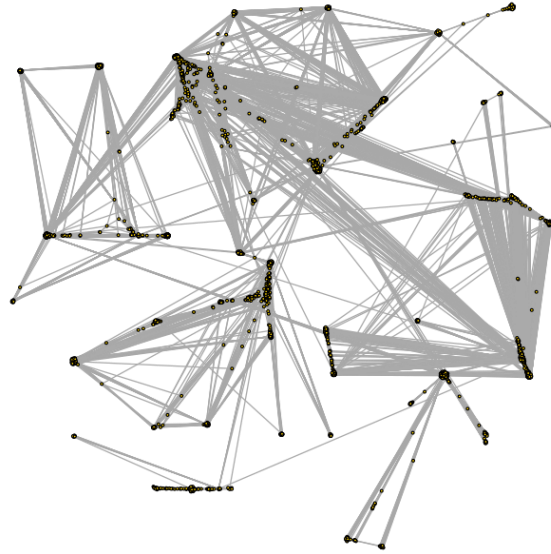


Fig. 1 Structure of Facebook network

Question 2

From problem 1, we have known that the network is connected, and the diameter of the network is 8. That is, the longest shortest path between each node pair (v_i, v_j) is 8.

Question 3

In this section, we examined degree distribution of the network. The degree distribution is shown in Fig. 2. And the average degree of the vertices of the network is 43.69, which means average friend number of the Facebook net is approximately 44.

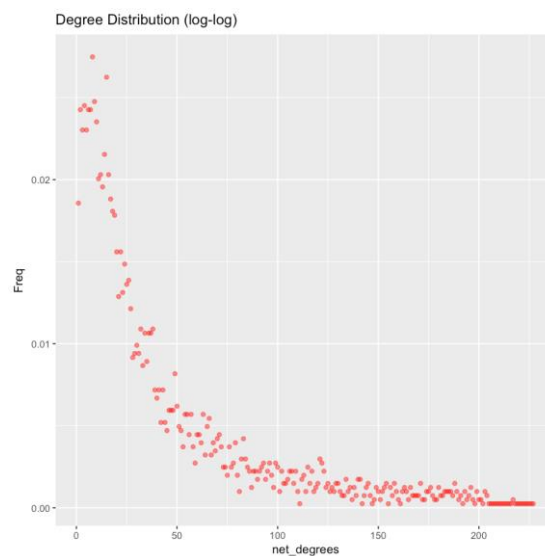


Fig. 2 Degree distribution of network

Question 4

We transformed the fig. into log-log scale and the distribution plot is shown in Fig. 3. Here we can fit a linear model into the log-transformed distribution, and the slope of the line is -2, which is in the $[-2,-3]$ region where power-law exponent usually lies.

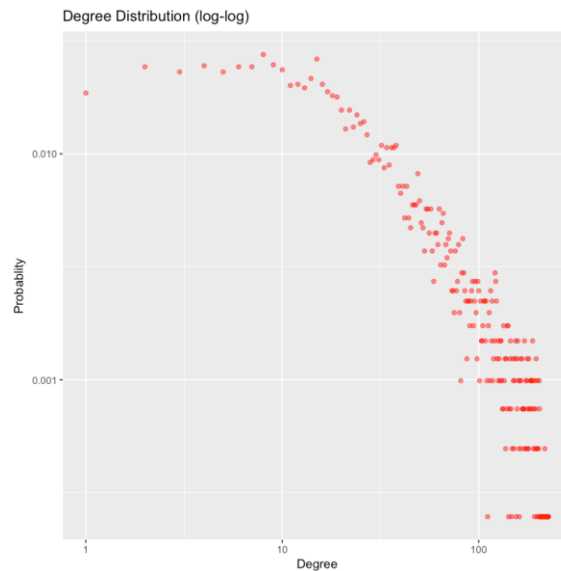


Fig. 3 Degree distribution of network (log-log)

1.2 Personalized network

In this section, we examined the properties of personalized network, which is generated from a core node, its neighbors and edges among them.

Question 5

We extracted personalized network of node 1# and plotted the network in Fig. 4. The network has 348 nodes and 2866 edges, which means node 1 # has 347 friends. The edges of the network is almost 8 times larger than nodes, indicating that many friends from node 1 # know each other. We can also verify this assertion by examining Fig. 4. In Fig. 4, we can see 5 communities that high connected inside, and this indicates that the node 1 # has connections to various communities. We will look into the structure of community later in Question 9.



Fig. 4 Personalized network with node 1 #

Question 6

The diameter of the network is 2. A trivial estimation of the diameter of the personalized network could be [1,2].

Question 7

A personalized network is composed of a vertex and its neighbors, which means, in the worst case, a path between a vertex pair (v_i, v_j) can be $v_i \rightarrow p \rightarrow v_j$, where v_i, v_j are neighbors in personalized network with vertex p . In this case, the diameter of the network is 2. This estimation gives an upper bound of the diameter. In the network, not all people in the network know each other, and paths between people should be established with the help of common friend p .

The lower bound of the network is 1, in this case, the personalized is fully connected, and for every pair of (v_i, v_j) , there exists an edge between them, and all friends in the network know each other.

1.3 Core node's personalized network

In this section, we studied the properties of personalized network with core nodes.

Question 8

A core node is defined as a node that has more than 200 neighbors. In the Facebook network, there are 40 core nodes, and the average degree of the core node set is 279.375.

1.3.1 Community structure of core node's personalized network

In this section, we studied the community structure of the following 5 core nodes:

- Node ID 1
- Node ID 10
- Node ID 349
- Node ID 484
- Node ID 1087

Question 9

We calculated the community structures with Fast-Greedy, Edge-Betweenness and Infomap community detection algorithms and compared modularity values. Table 1 shows the result.

Table 1. Modularity with different community detection algorithms

Vertex #	Fast-Greedy	Edge-Betweenness	Infomap
1	0.413	0.353	0.389
108	0.435	0.506	0.508
349	0.250	0.134	0.095
484	0.507	0.489	0.515
1087	0.145	0.028	0.026

From the Table 1, we can see that the Fast-Greedy algorithm has the highest modularity score among vertex 1, 349, 484 and 1087. In comparison to the other two, Fast-Greedy algorithm tends to have higher modularity score. This phenomenon can be explained by the process of the algorithm. Fast-Greedy first initializes as one vertex is one community. After that, the algorithm merges nodes recursively until the modularity does not increase. Fast-Greedy algorithm is a greedy method especially optimized for modularity, and thus it is no surprising that the algorithm tends to have higher modularity score.

We also find that Fast-Greedy does not work well in vertex 108#. Consider the scale of personalized network (shown in Table 2), it can be concluded that the precision of greedy method is not so good in large networks, since the modularity optimization problem has not been proved strictly convex, and the greedy method does not always work well. To sum up, Fast-Greedy is a computationally feasible algorithm for large networks and it is a good way to do a rough estimation which can be set as baseline.

The community structures of Edge-Betweenness and Infomap seem to be more precise (more communities with clearer illustration) in comparison to Fast-Greedy, shown in Table 3. Edge-Betweenness algorithm chooses community based on betweenness score of each edge. The algorithm would rank edges by decreasing betweenness score and remove those edges with highest score one by one. Edge-Betweenness algorithm bases on the assumption that if a path travels among various communities, it should pass edges that link between communities with higher probability. It is a top-down method in comparison with Fast-Greedy, and one of the drawbacks of the technique is that, the algorithm would update betweenness score every time it removes an edge, making it computationally intensive with large networks. Based on our observations, we also see that, the running time of the edge-betweenness algorithm is the longest.

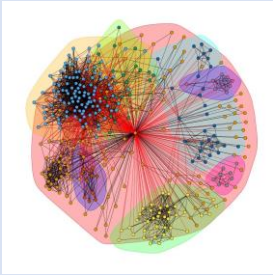
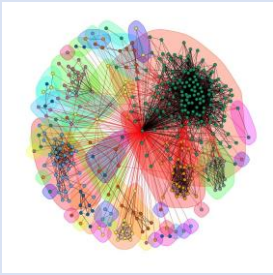
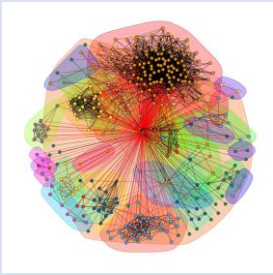
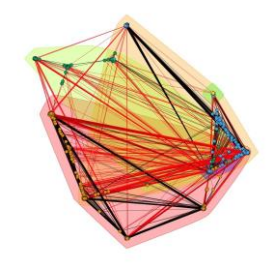
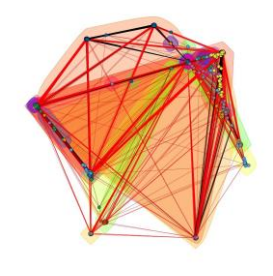
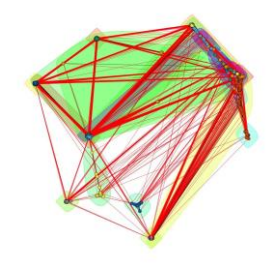
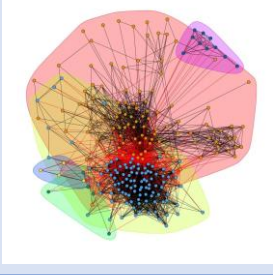
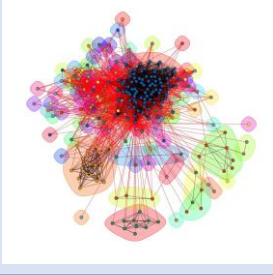
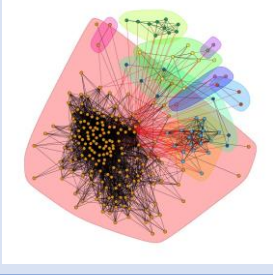
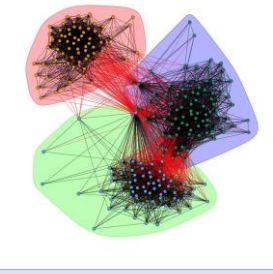
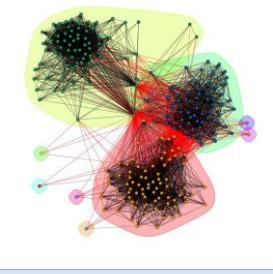
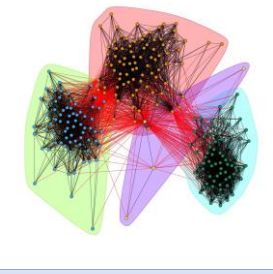
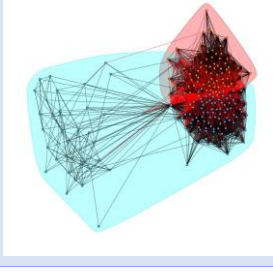
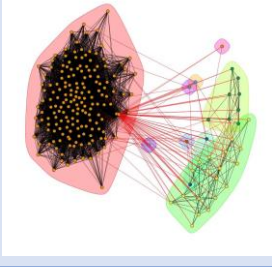
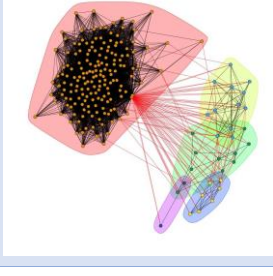
Infomap, however, optimizes the length of random-walk description by information coding, and it can be faster than Edge-Betweenness, and from Table 3, we can see that Infomap algorithm achieves similar performance to Edge-Betweenness.

Table 2 Basic characteristics of personalized network

Vertex #	Node number	Edge number
1	348	2866
108	1046	27795
349	230	3441
484	232	4525
1087	206	7409

Table 3 shows the community structure of personalized networks. Combined with Table 1, the community structure would be more clear with higher modularity scores. From Table 3, we can see that, community structures of 108# and 484# have are distinguishable, whereas those of 349# and 1087# is more ambiguous, and the structures yielded by different algorithms are not similar. In reality, the result may show that a person like vertex 349# or 1087# is likely to have friends from similar background (e.g. same primary school), and friends of person like vertex 108# or 484# are more diverse (e.g. home and abroad).

Table 3 Community structure of personalized networks

Vertex #	Fast-Greedy	Edge-Betweenness	Infomap
1			
108			
349			
484			
1087			

1.3.2 Community structure with the core node removed

In this section, we removed the core node of personalized network and studied the effect.

Question 10

In this question, we generated personalized networks as part 1.3.1 and removed the core nodes of those networks. We also calculated the modularity with different community detection algorithms and plotted community structures with core node removed, shown as Table 4 and Table 5.

Table 4 shows the modularity score with core removed and its difference with Table 3 ($\Delta = \text{Modularity_coreremoved} - \text{Modularity_withcore}$). From the Table 4, it can be justified modularity scores are slightly larger with core node removed. In the personalized network, the core node has edges connect with all other nodes in the network. If we assume the core node is in community A, and other nodes lies in community set $\{A, B, \dots, M\}$, and the edges between the core node and its neighbors who do not belong to community A would be counted in modularity. If core node is removed, those edges should be eliminated, and would contribute a higher modularity. However, from Table 2, it can also be found that the cardinality of “cross community” edges is relatively small in comparison to large edge amount in a graph. Hence, the increase modularity effect of core node removal is very small.

Table 4 Modularity with different community detection algorithms (core removed)

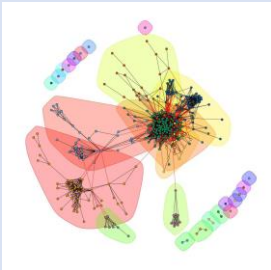
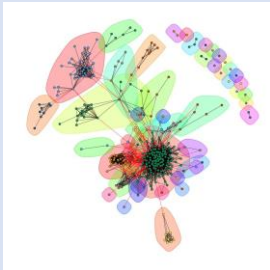
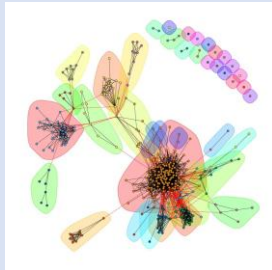
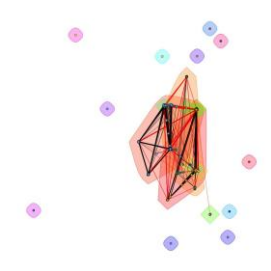
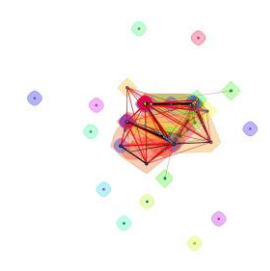
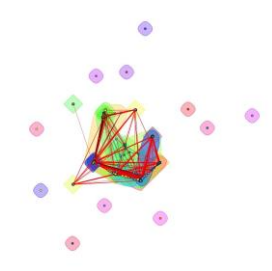
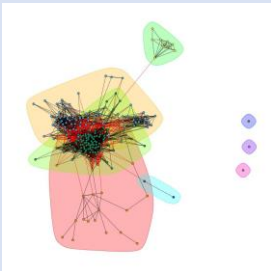
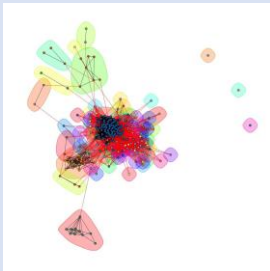
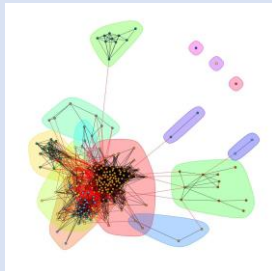
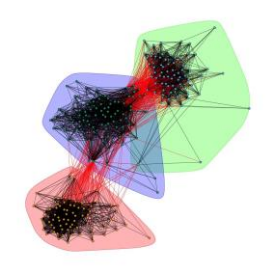
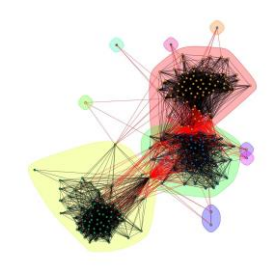
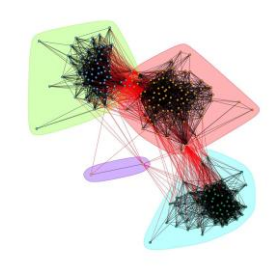
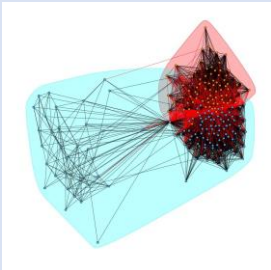
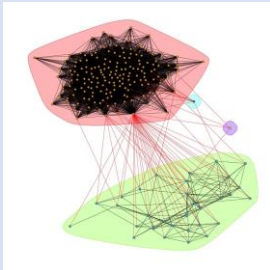
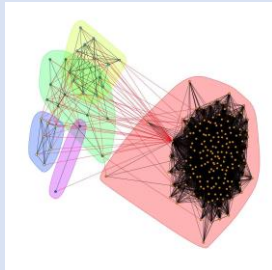
Vertex #	Fast-Greedy	Δ	Edge-Betweenness	Δ	Infomap	Δ
1	0.442	0.029	0.416	0.063	0.418	0.029
108	0.458	0.023	0.521	0.015	0.521	0.013
349	0.246	-0.004	0.151	0.017	0.247	0.152
484	0.534	0.027	0.515	0.026	0.543	0.028
1087	0.148	0.003	0.032	0.004	0.027	0.001

Table 5 illustrates the structure of community of personalized networks, with core removed. Although the graph is not connected with core removed, it is not partitioned into several pieces. Instead, only a small amount of nodes are disconnected, which is a very interesting result. In reality, the result shows the robustness of social networks, which means that in great chance, a person’s all friends may get contact with each other without the help of that person. (The phenomenon can be best matched with a well-known Chinese proverb: 没了谁地球都照样转)

If we ignore the isolated communities and look only into larger ones. We can still see a well-partitioned community structure in vertex 108# and 484#, and those

communities did not collapse as core node removed. Communities of vertex 349# and 1087# are still ambiguous.

Table 5 Community structure of personalized networks (core removed)

Vertex #	Fast-Greedy	Edge-Betweenness	Infomap
1			
108			
349			
484			
1087			

Question 11

In one's personalized network, the mutual friends of a node are its neighbors except the core node, and a node's neighbors number is equal to its degree; thus, we can conclude that $Embeddedness_{corenode}(node) = degree(node) - 1$.

Question 12

In this section, we measured the distribution of embeddedness and dispersion regarding to Node 1, 108, 384, 484, 1087, core nodes in facebook network. As core nodes, they all have over 200 neighbors; their personalized networks, however, have different characters in terms of embeddedness and dispersion.

For any node in a core node's personalized network, a high embeddedness means this node's personalized network strongly coincident with core node's, vice versa. Therefore, we can conclude that rare nodes share crowded mutual friends with node 1 and node 108, whose distribution of embeddedness $\approx O(x^{-1})$, while node 349, node 484, and node 1087 have more nodes in their personalized network which have a relatively larger set of mutual friends. Since classmates and colleagues have more probability to become pairs of friends in Facebook, it could be hypothesize that node 349, node 484, and node 1087 have more classmates and colleagues in their personalized networks than node 1 and node 108.

Another important measure for a personalized network is dispersion. In this project, it is calculated as the sum of shortest path, and infinity number is replaced by diameter of a personalized network +2. It can be found that for all core nodes, the most frequent dispersion is arbitrary near to 0. In addition, for node 1, node 108, and node 349, almost all other possible dispersions seldom occur, but, for node 484 and node 1087, there are also some possible dispersions have high frequency. This result is corresponded to the difference between community structures of those nodes: node 1, node 108, and node 349 have oblivious communities while 484 and node 1087 have more obvious structures.

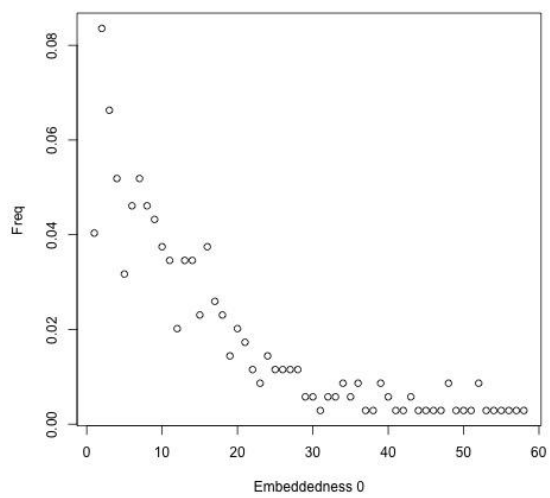


Fig. 5. Embeddedness Node 1

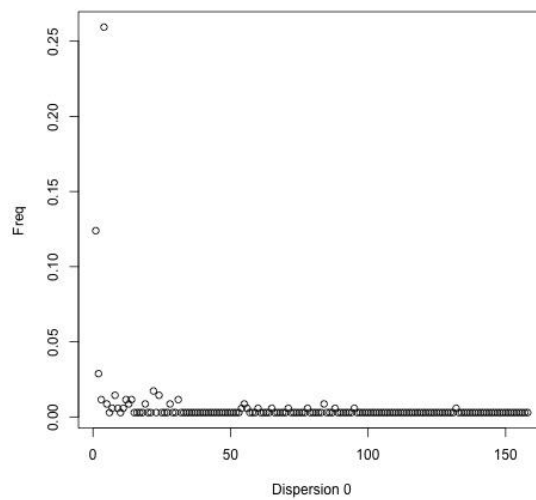


Fig. 6. Dispersion Node 1

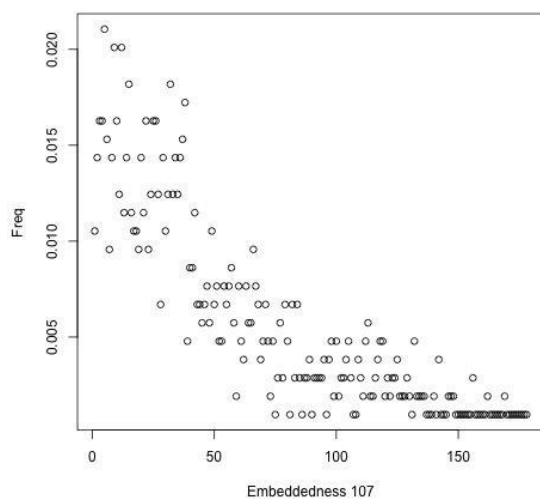


Fig. 7. Embeddedness Node 108

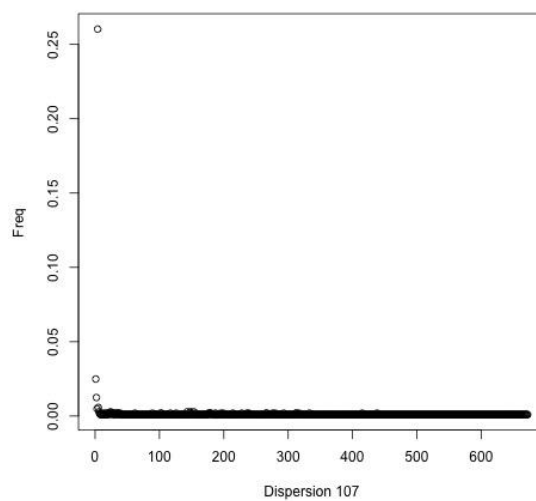


Fig. 8. Dispersion Node 108

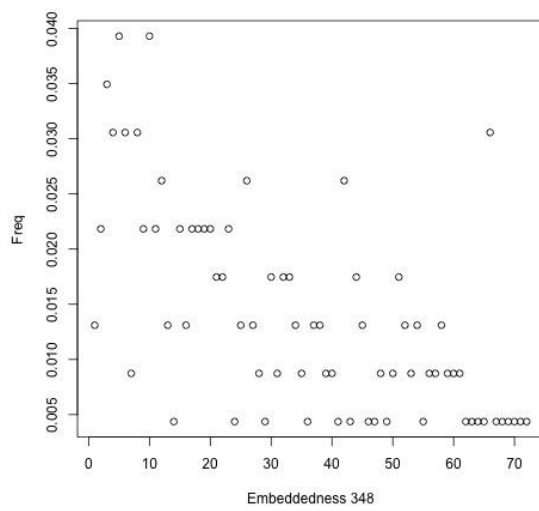


Fig. 9. Embeddedness Node 349

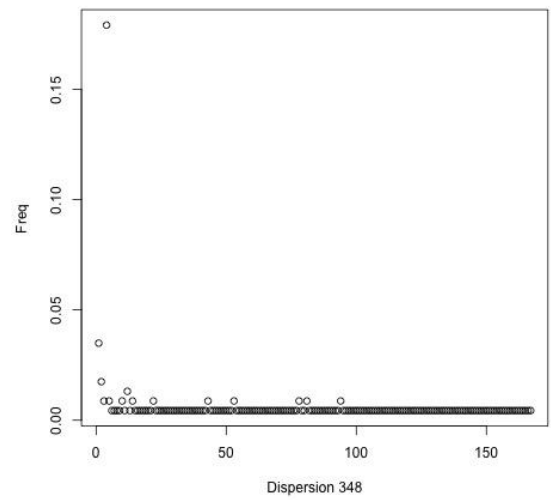


Fig. 10. Dispersion Node 349

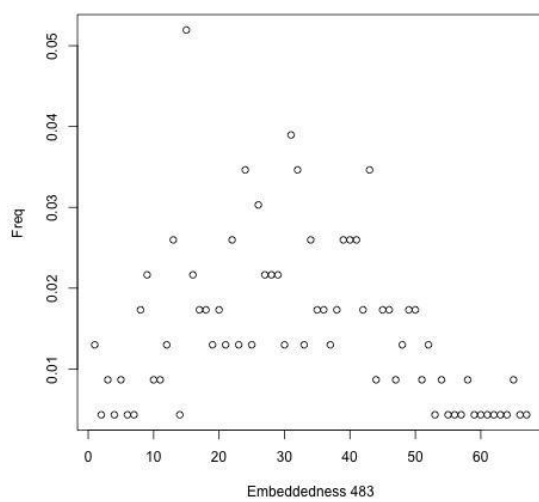


Fig. 11. Embeddedness Node 484

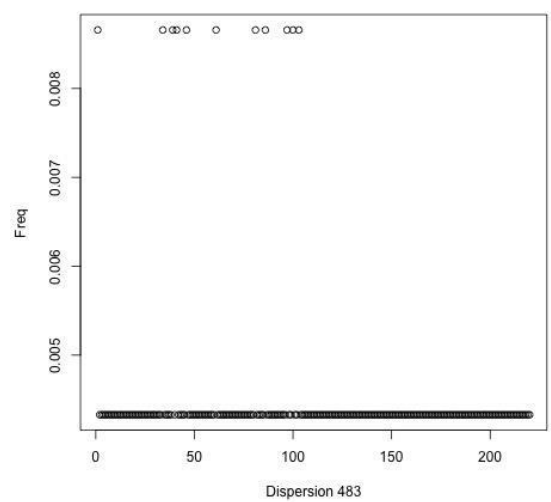


Fig. 12. Dispersion Node 484

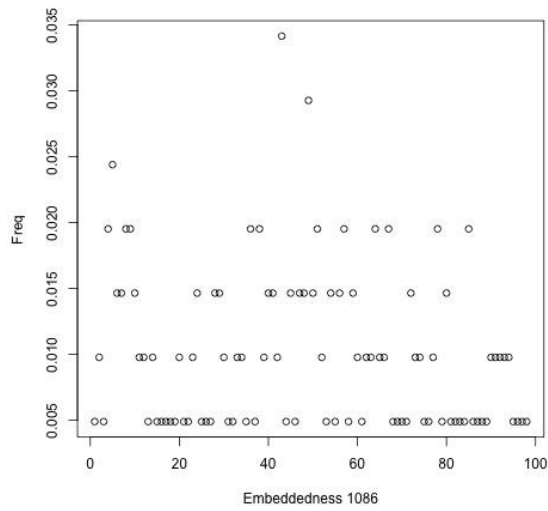


Fig. 13. Embeddedness Node 1087

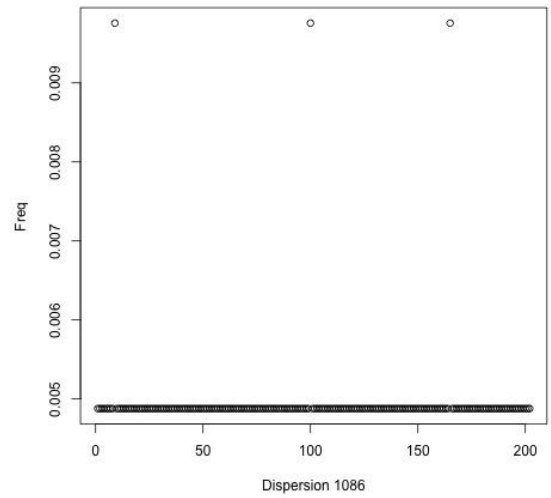


Fig. 14. Dispersion Node 1087

Question 13

In this section, different community structures of five nodes are illustrated. In those figures, the node with max dispersion and all incident edges are highlighted.

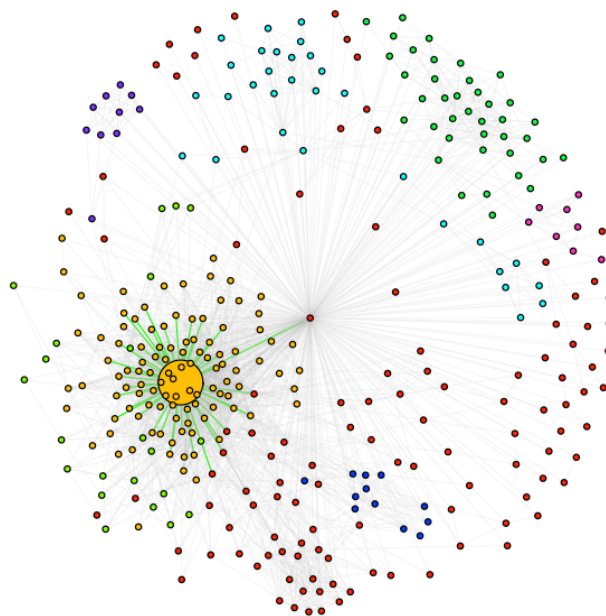


Fig. 15. Maximum Dispersion Node 1

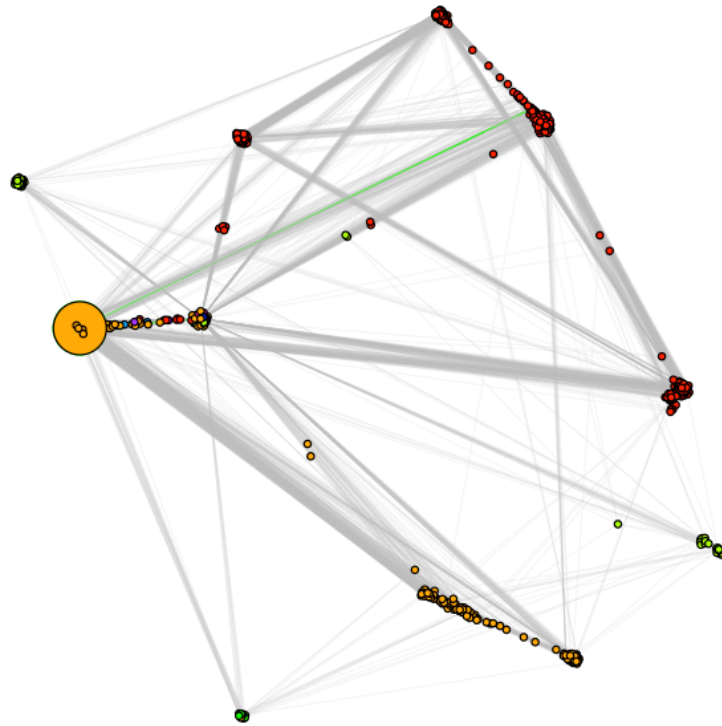


Fig. 16. Maximum Dispersion Node 108

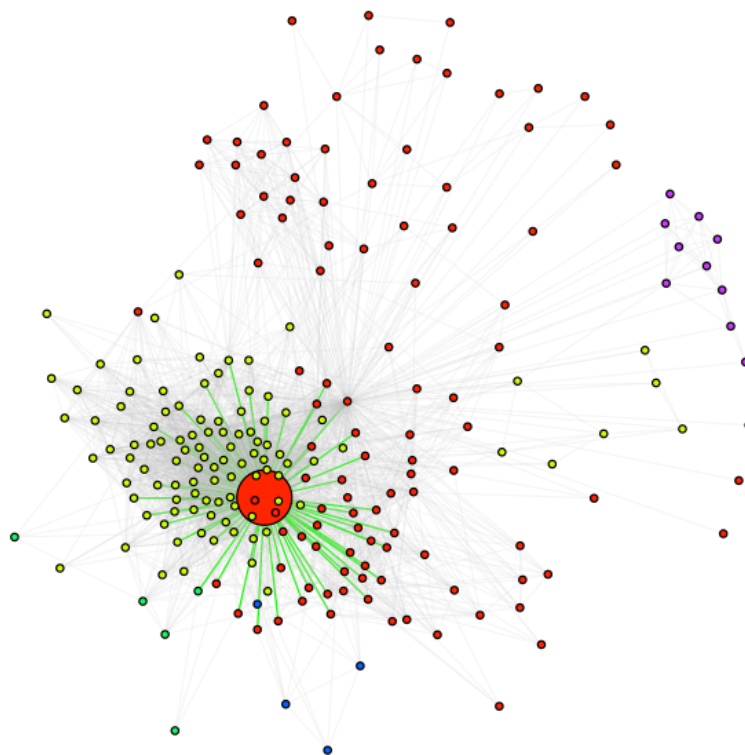


Fig. 17. Maximum Dispersion Node 349

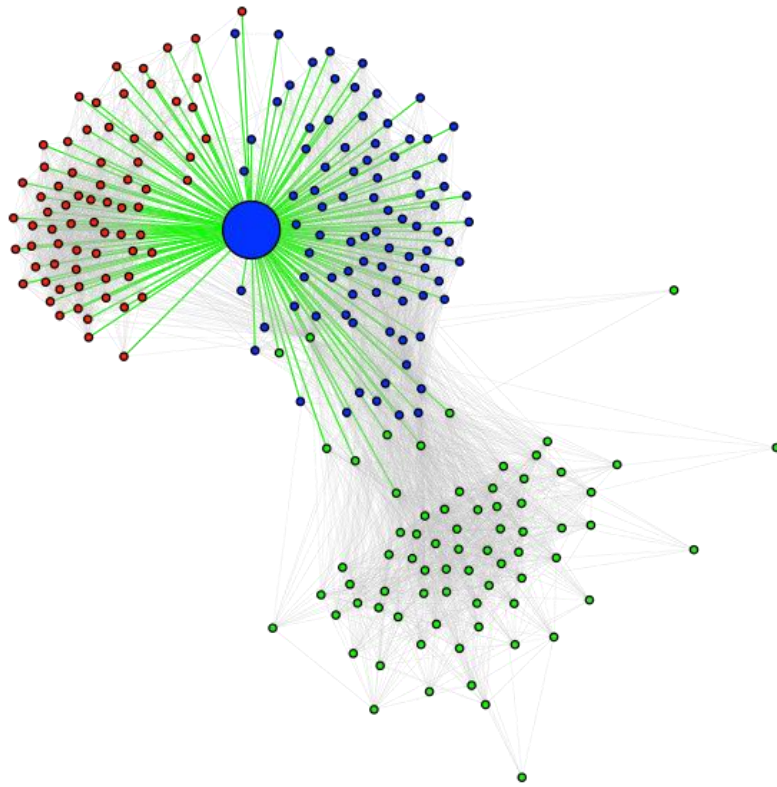


Fig. 18. Maximum Dispersion Node 484

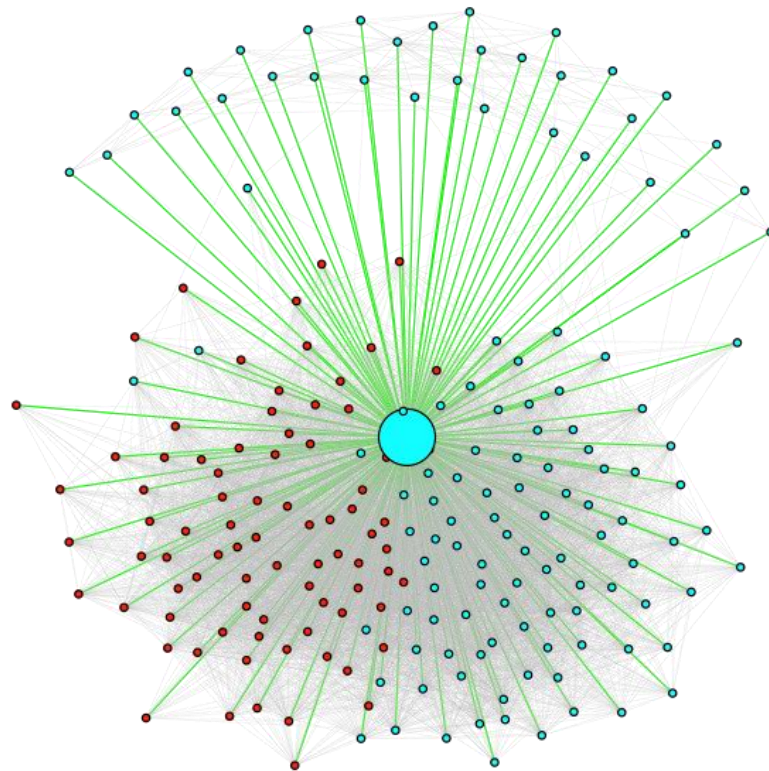


Fig. 19. Maximum Dispersion Node 1087

Question 14

In this section, different community structures of five nodes are still illustrated, but the node with max embeddedness and $\frac{Dispersion}{Embeddedness}$ and all incident edges are highlighted.

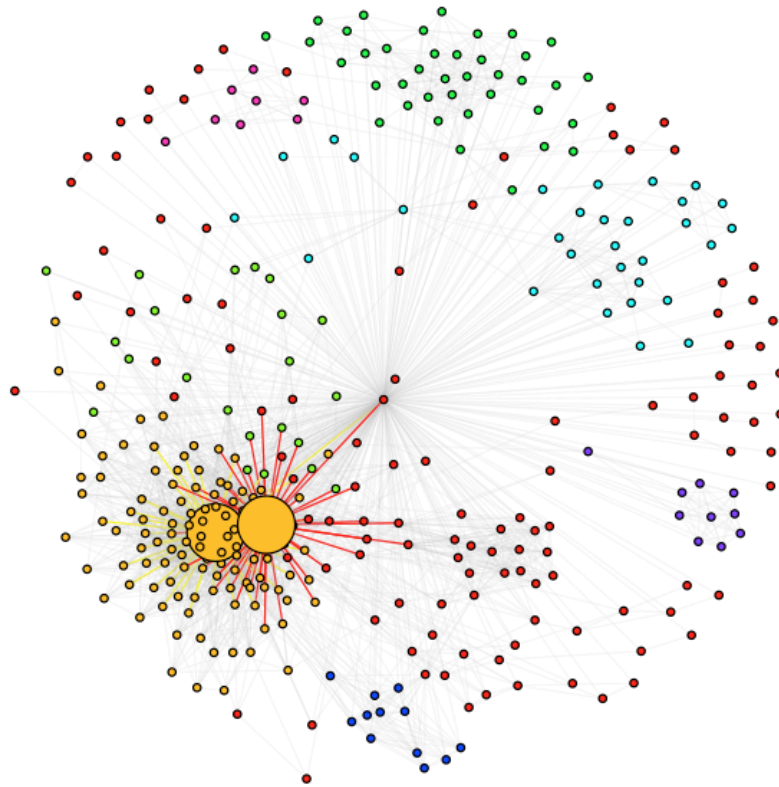


Fig. 20. Maximum Embeddedness and $\frac{Dispersion}{Embeddedness}$ Node 1

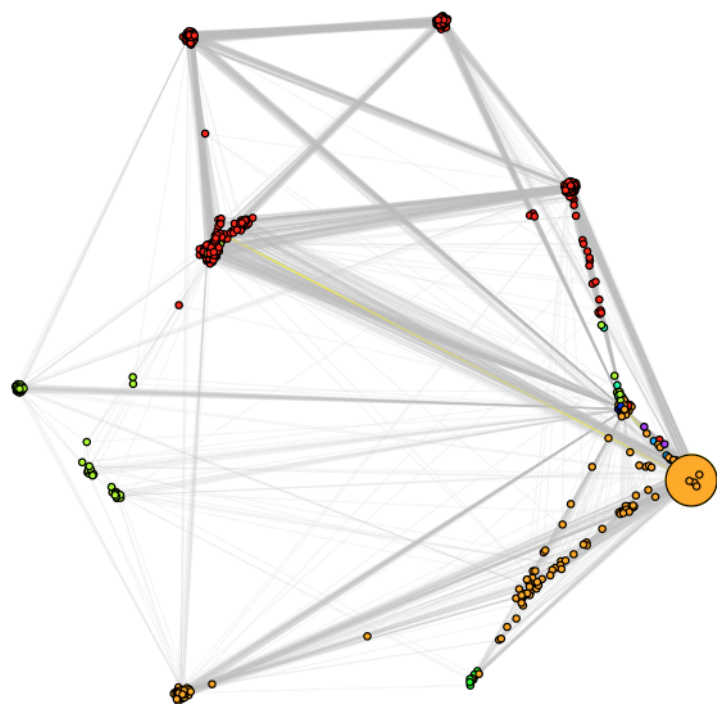


Fig. 21. Maximum Embeddedness and $\frac{Dispersion}{Embeddedness}$ Node 108

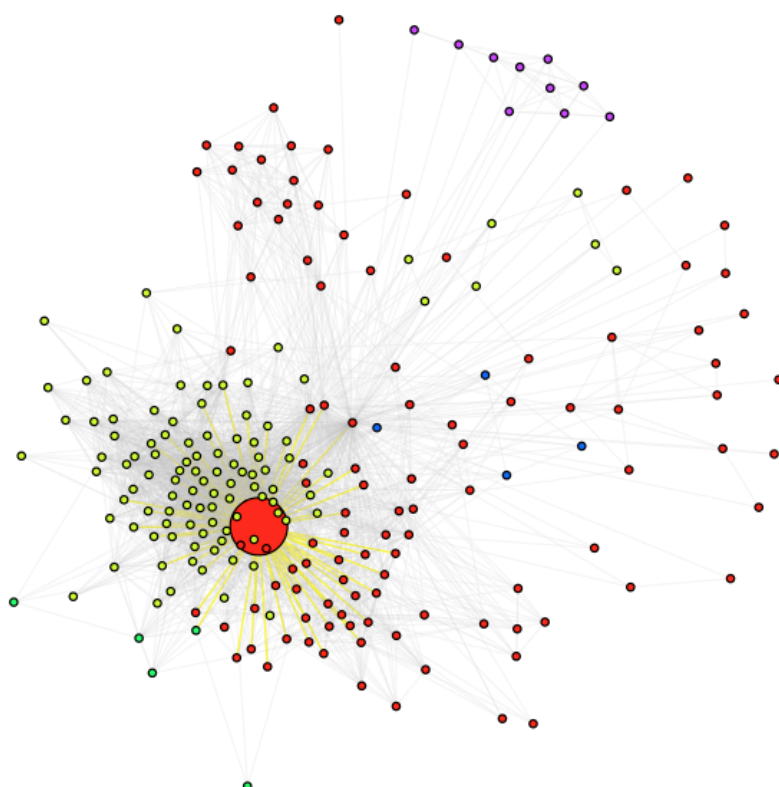


Fig. 22. Maximum Embeddedness and $\frac{Dispersion}{Embeddedness}$ Node 349

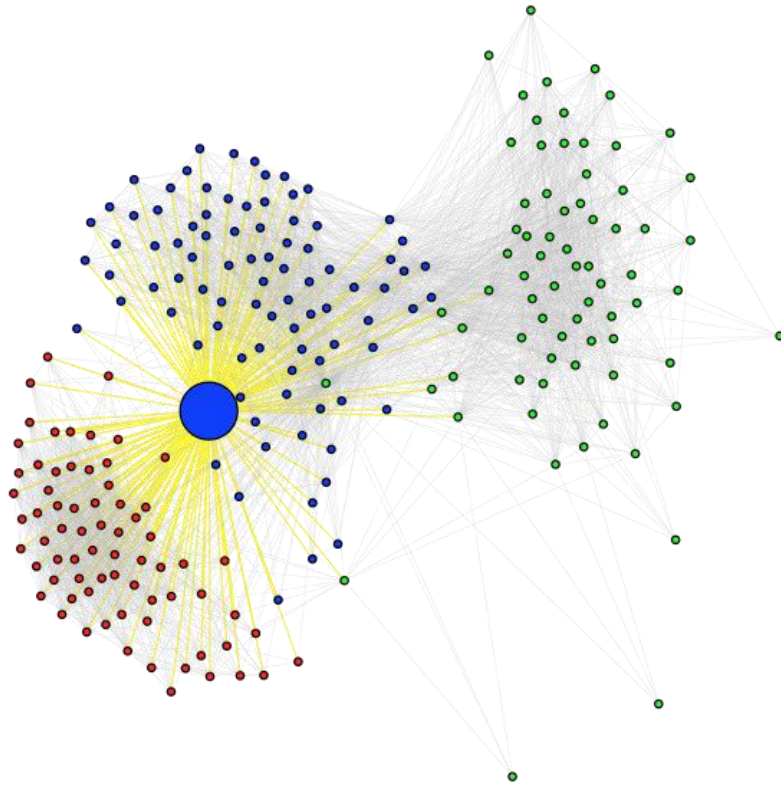


Fig. 23. Maximum Embeddedness and $\frac{Dispersion}{Embeddedness}$ Node 484

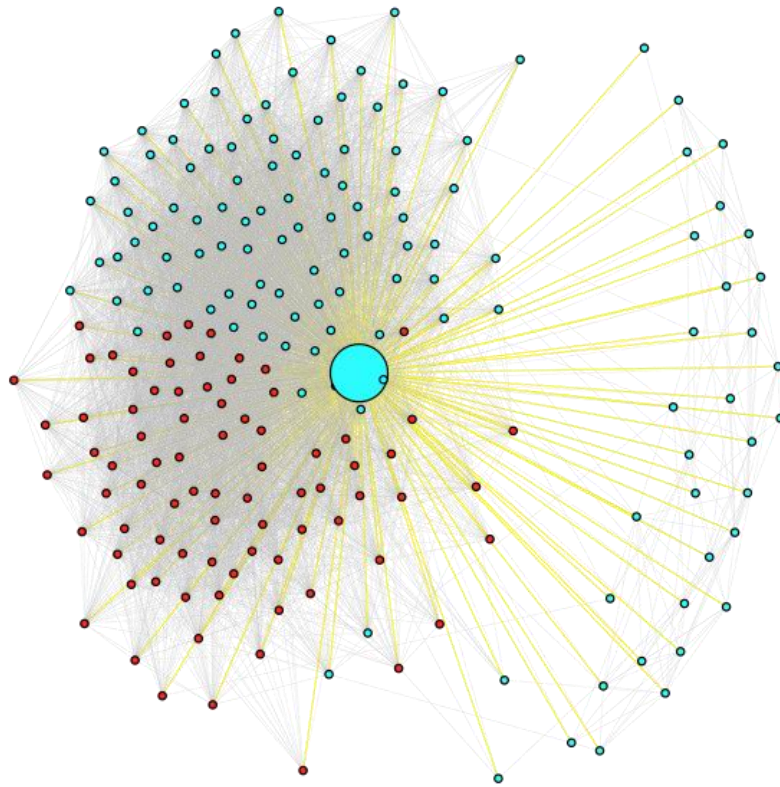


Fig. 24. Maximum Embeddedness and $\frac{Dispersion}{Embeddedness}$ Node 1087

Question 15

From the figures in question 13 and question 14, for any nodes, it can be conclude that:

- Dispersion: A node with high dispersion is likely to have more mutual friends, regarding to core node, from different communities. In real life, this means those two people have tie relationship, e.g. Alice and Bob are from the same secondary school and undergraduate school; thus, Alice and Bob share plentiful mutual friends from two communities: secondary schoolmate and undergraduate schoolmate. In this case, in Alice's personalized network, Bob should have high dispersion, and, in high probability, Bob could be the node with max dispersion.
- Embeddedness: Embeddedness is a measure of number of mutual friends between an arbitrary node and a core node. A node with high embeddedness is the same as it has many mutual friends with a core node. One important discovery is, for all five core nodes in this project, the node with max embeddedness is also the node with max dispersion. This discovery corresponds to the definition of dispersion, since the dispersion is the sum of all distance between each pair of mutual friends, and it is apparently that the dispersion is positive correlation to the number of mutual friends, which is also the embeddedness.
- $\frac{Dispersion}{Embeddedness}$: $\frac{Dispersion}{Embeddedness}$ is the normalization of dispersion. In this project, except the node 1, whose node with max embeddedness is not the same as the node with max $\frac{Dispersion}{Embeddedness}$, all others have the same node with max embeddedness and max $\frac{Dispersion}{Embeddedness}$. Been normalized on embeddedness, this measure implies the separation of mutual friends, i.e., if a node has high $\frac{Dispersion}{Embeddedness}$, this node is likely to have mutual friends from different contexts regardless of number of mutual friends.

In conclusion, dispersion is a better measure for the tie strength of a core node and its friends. The dispersion can measure both the number of mutual friends and number of spanned contexts.

1.4 Friend recommendation in personalized networks

Predicting future links between pairs of nodes in the network can be applied as recommending friends to users. In this section, we will explore some neighborhood-based measures for friends recommendation. The network that we use for this part is the personalized network of node with ID 415.

Question 16

We first create the list of users who we want to recommend new friends to. We create this list by picking all nodes in this personalized network with degree 24. The list is denoted as Nr. $|Nr|$ is 11.

Question 17

We use three different neighborhood-based measures to recommend friends to each user in Nr. In addition, we compute the average accuracy of each algorithm and the result is shown in Table 6.

Table. 6 Accuracy Comparison between Three Neighborhood-based Measures

Neighborhood-based Measures	Average Accuracy
Common Neighbors measure	0.729
Jaccard measure	0.808
Adamic Adar measure	0.726

As experiments of many times demonstrate, the effect of common neighbors measure is on a par with that of adamic adar measure. Usually, the accuracy algorithm using common neighbors measure is slightly better than that of using adamic adar measure.

By contrast, the supremacy of Jaccard measure-based recommendation is distinct among three measures. Thus, friend recommendation algorithm using Jaccard measure has the best performance in this case.

Part II Analysis of Google+ Network

Question 18:

There are 132 ego-nodes in the file, and 57 personal networks who have more than two circles.

Question 19:

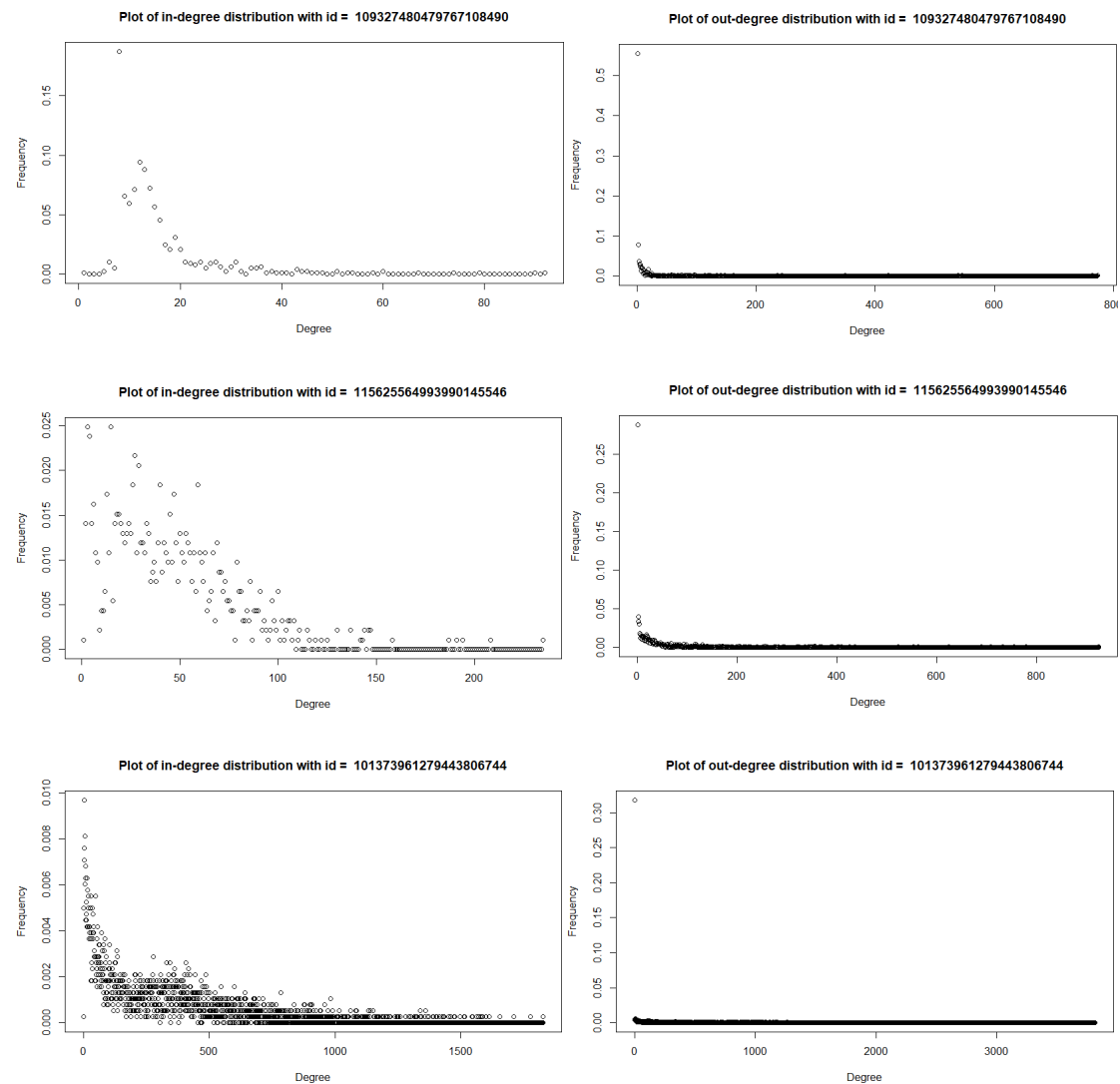


Fig. 25. In-degree and out-degree distribution for three ego-nodes

From the six plots, we could see that the out-degree distribution for the three nodes are almost similar, but the in-degree distributions are different. This may be due to the difference of their network structures. Elaborated analysis would be implemented in following questions.

Question 20:

In this question, we used the walk-trap detection algorithm to detect the

communities for the three users, the plots are shown below.

Communities with id = 109327480479767108490

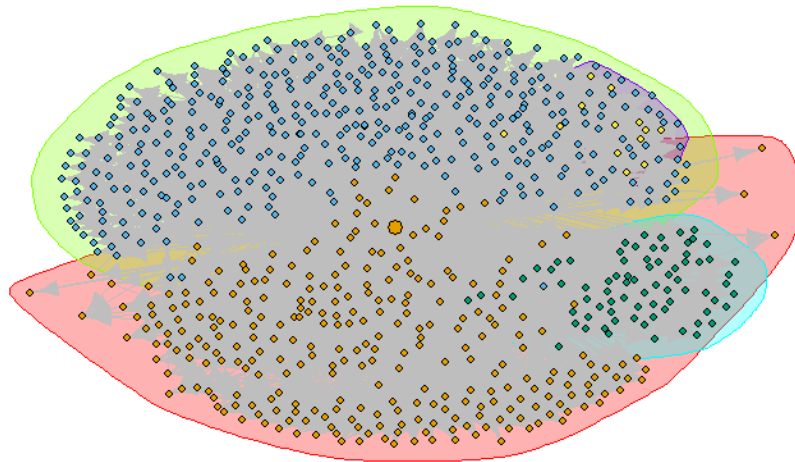


Fig. 26. Community plot of id = 109327480479767108490

Communities with id = 115625564993990145546

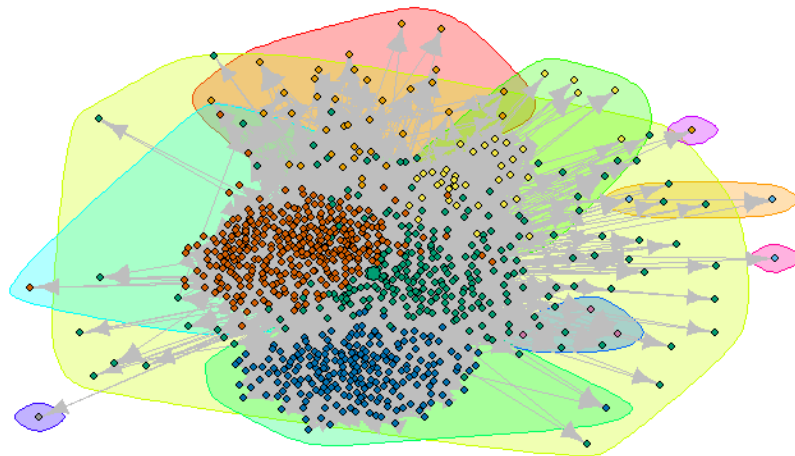


Fig. 27. Community plot of id = 115625564993990145546

Communities with id = 101373961279443806744

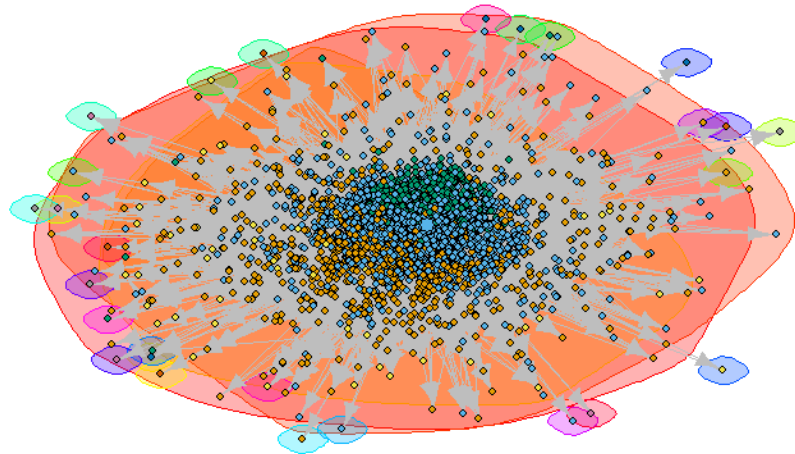


Fig. 28. Community plot of id = 101373961279443806744

The modularity scores for the three communities are:

Table. 7. Modularity scores for the three communities

id	Modularity Score
109327480479767108490	0.25276535939251
115625564993990145546	0.319472554647349
101373961279443806744	0.191090282684037

The modularity scores for three ego nodes are not similar. The modularity score for the third node is smaller than the first two nodes, which means the strength of the community structure is weaker than the previous two, and the quality of the community partitioning is also lower than the previous two.

Question 21:

From a clustering perspective, a clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class. A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster.

By our example, big homogeneity score means less users in different circles are contained in the same community, and big completeness score means less numbers of users in the same circle belong to different communities. Usually, these two numbers are negatively correlated.

Question 22:

Table. 8. Homogeneity scores and completeness scores for the three nodes

id	Homogeneity	Completeness
109327480479767108490	0.851885115440867	0.329873913536689
115625564993990145546	0.451890303032235	-3.4239623491117
101373961279443806744	0.00386670698130509	-1.5042383879479

From the results, we could see that the homogeneity and completeness scores for the three nodes are different. The difference may be due to the distribution of users in communities and circles for every of these three networks. We could explain this phenomenon in detail by combining the results in previous questions.

a.

For the first network, whose id is 109327480479767108490, the homogeneity scores and completeness are relatively high. The structure for this node is that it has four communities and three circles, as shown in Fig. 26. The high homogeneity score means that for every community, it may contain very few users that belong to different circles. The high completeness score means that for every user in one circle, they may largely possible to be contained in the same community. The results can be obviously retrieved from Fig. 26. : The yellow and blue point community users are almost all contained in the light green circle. Orange and green ones are almost all contained in pink circle. This proves that the homogeneity score could be high; For every circle (light-green, pink, light-blue), it contains only one or two colors of points which stand for different communities. This proves the completeness score could be high. The result for the first user may indicate that his/her social networks are well-clustered that friends from different circles/communities are almost not overlapped.

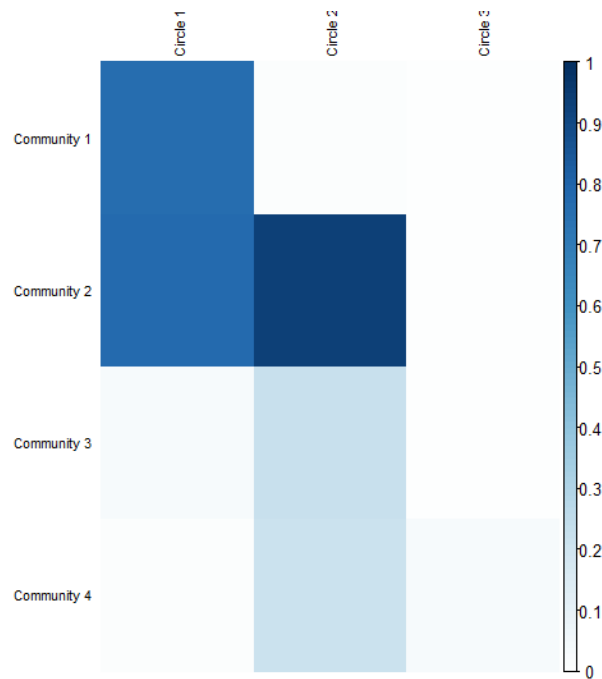


Fig. 29. Correplot between communities and circles for node 1.

Fig. 29. is a correlation plot of communities and circles which indicate the number of nodes belonging to a certain community and circle. From the plot, we could also see the similar results that it has relatively high homogeneity and completeness.

b.

For the second network, whose id is 115625564993990145546, the homogeneity score is relatively high, but the completeness score is low. It has 10 communities and 31 circles. The distribution is shown in Fig. 30.

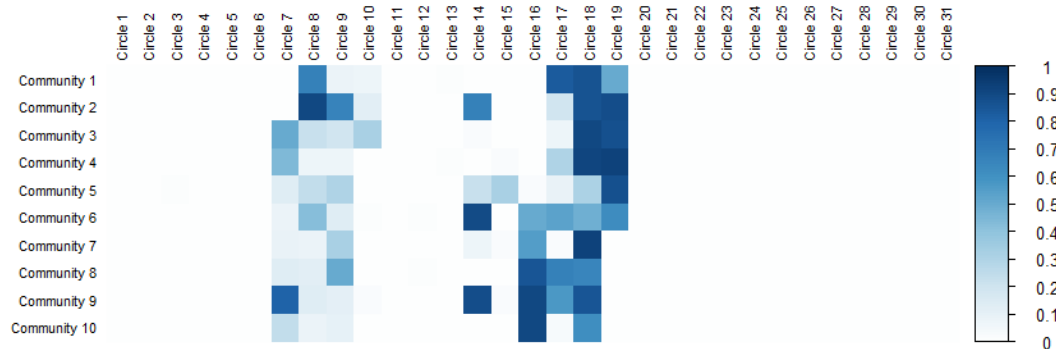


Fig. 30. Correplot between communities and circles for node 2.

From the correplot, we could see that for each community, it may contain only a few numbers of circles. In contrast, for each circle with community information, they would be distributed in every community. This may give rises to a relatively high homogeneity and pretty low completeness. This means, for this ego-node network

c.

For the third network, whose id is 101373961279443806744, both the homogeneity score and the completeness score are low. It has 31 communities and 3 circles.

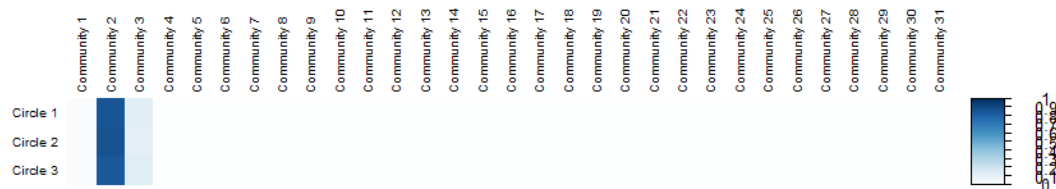


Fig. 31. Correplot between communities and circles for node 3.

From the correplot, we could see that the C matrix for this node is extremely sparse. The nodes in the circle only show information of a small number of communities, and from the correplot, we could also see that the nodes for

communities 1, 2 and 3 are almost evenly distributed. This may give rise to both entropy and conditional entropy reasonably high so that the completeness and homogeneity are both low accordingly. This means, for this ego-node network, the communities and circles are almost mixed with each other. They are not well-separated.

Appendix: The calculated data are shown below:

Table. 9. Modularity scores for the three communities

id	109327480479767108490	115625564993990145546	101373961279443806744
N	764	727	521
H(C)	1.51595415381635	12.2126251379311	0.554456498907601
H(k)	1.45020885763114	1.55982884141566	0.711725628565447
H(C K)	0.224535374489447	6.69385826353235	0.552312578092445
H(K C)	0.971822786318783	6.90062406548138	1.78233064073994
Homo	0.851885115440867	0.451890303032235	0.00386670698130509
Comp	0.329873913536689	-3.4239623491117	-1.5042383879479

Last, we are still confused about why completeness scores are calculated negative in this part (actually, we've done a lot of researches and still not found a place for completeness to be negative). We now believe that the problem is generated from matrix C, where we could have repetitive numbers of nodes calculated by definition in the statement, and in the meantime, b and N are calculated using unique nodes. However, this does not influence the conclusion for the relationship between circles and communities. Just our personal ideas.