ECE 232 Project 2 Report

July 24, 2024

Name: Kelly Furuya

1 Facebook network

1.1 Structural properties of the Facebook network

Q1. First look at the network

Q1.1. Report the number of nodes and number of edges of the Facebook network.
The network has 4039 nodes and 88234 edges.

Q1.2. Is the Facebook network connected? If not, find the giant connected component (GCC) of the network and report the size of the GCC.
Yes, the network is connected.

Q2. Find the diameter of the network. If the network is not connected, then find the diamter of the GCC.
The diameter of the network is 8.

Q3. Plot the degree distsribution of the facebook network and report the average degree.

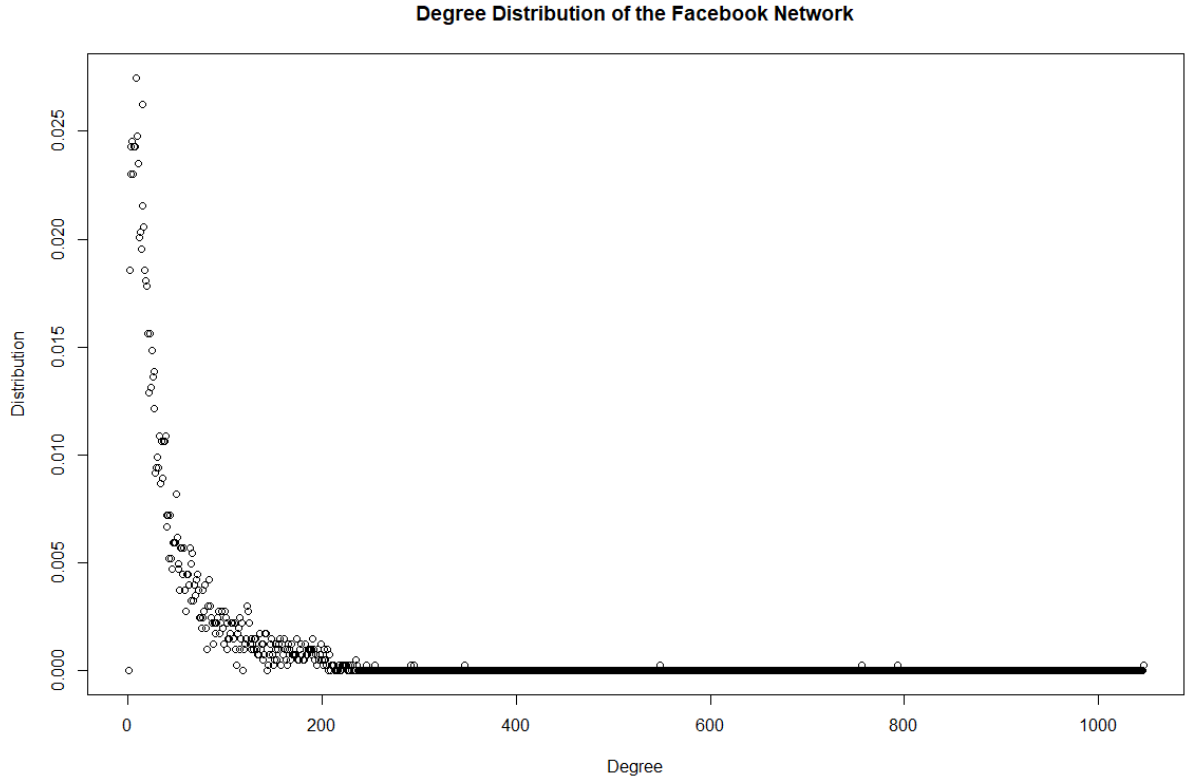


Figure 1: Degree distribution of the Facebook Network.

The degree distribution shows that the majority of the nodes have a degree below 200, and the graph appears to follow the Power Law.

The average degree is 43.69052.

Q4. Plot the degree distribution of Question 3 in a log-log scale. Try to fit a line to the plot and estimate the slope of the line.

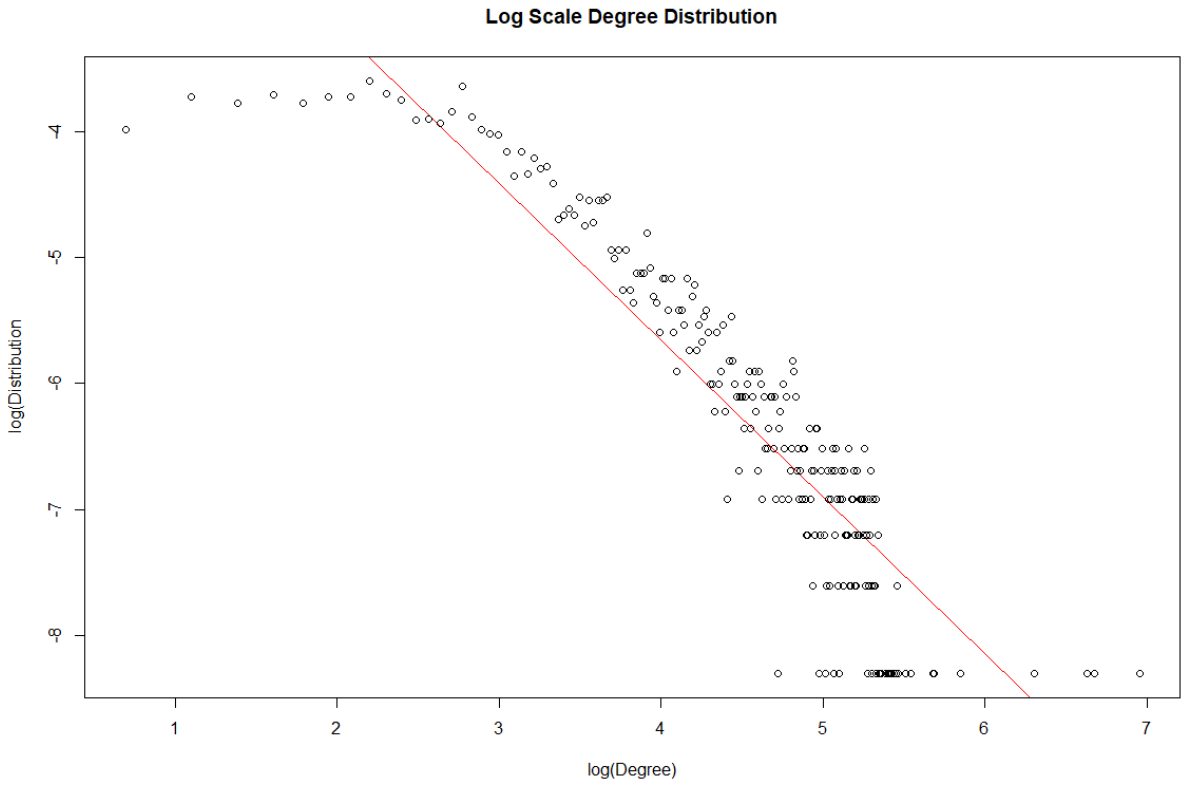


Figure 2: Log Scale Degree distribution of the Facebook Network.

The slope is -1.247539.

Since the log scale degree distribution is linear, we can safely assert that the degree distribution does follow the Power Law.

1.2 Personalized network

Q5. Create a personalized network of the user whose ID is 1. How many nodes and edges does this personalized network have?

We can use the functions *neighbors* and *induced_subgraph* to find the neighboring nodes of a user with the ID 1 which would be index 0. We have to be careful to remember to add back in the user's node before running *induced_subgraph*. There are 347 nodes and 2849 edges in the resulting network.

Q6. What is the diameter of the personalized network? Please state a trivial upper and lower bound for the diameter of the personalized network

The personalized network has a diameter of 2. A trivial upper and lower bound would be 2 and 1 respectively.

Q7. In the context of the personalized network, what is the meaning of the diameter of the personalized network to be equal to the upper bound you derived in Question 6. What is the meaning of the diameter of the personalized network to be equal to the lower bound you derived in Question 6 (assuming there are more than 3 nodes in the personalized network)?

The diameter indicates the smallest degree of separation between any two users in that subgraph (i.e. how many friends/friend of a friend would they have to go through to reach each other).

A trivial upper bound would be 2 as that's the largest diameter that a graph generated in this manner can have. Any given pair of nodes that doesn't include the user's node will have that user node in common, meaning if there is not an edge connecting the pair, then the shortest path will always be one edge to the user node and then one more to the end node. 2 only works if we look at only neighbors 1 edge away from the user. A trivial lower bound would be 1 as 1 will always be the shortest the diameter can be (assuming a path exists).

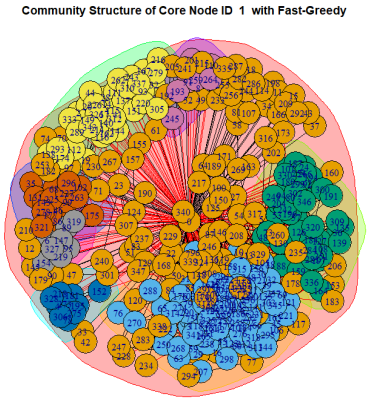
1.3 Core node's personallized networks

Q8. How many core nodes are there in the Facebook network. What is the average degree of the core nodes?

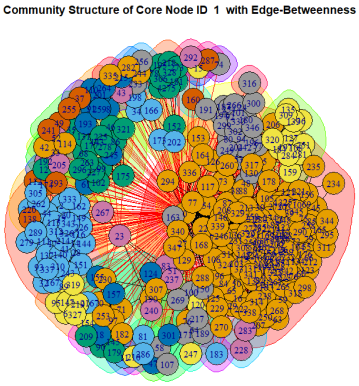
There are 40 core nodes and the average degree is 279.35.

1.3.1 Community structure of core node's personalized network

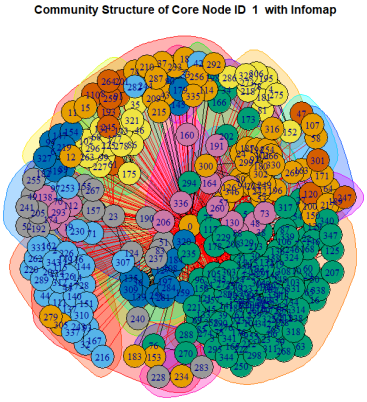
Q9. For each of the above core node‘s personalized network, find the community structure using Fast-Greedy, Edge-Betweenness, and Infomap community detection algorithms. Compare the modularity scores of the algorithms. For visualization purpose, display the community structure of the core node's personalized 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.



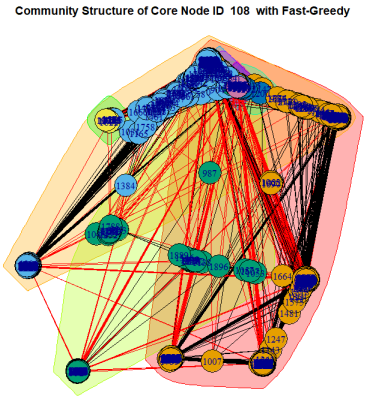
(a) Community structure for Node ID 1 with Fast-Greedy Algorithm.



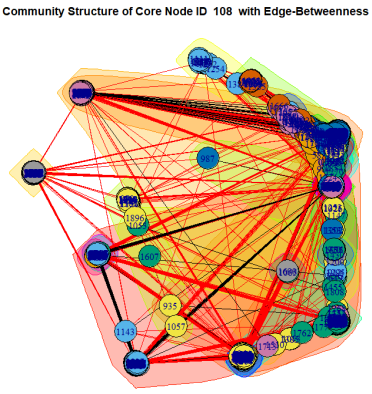
(b) Community structure for Node ID 1 with Edge-Betweenness Algorithm.



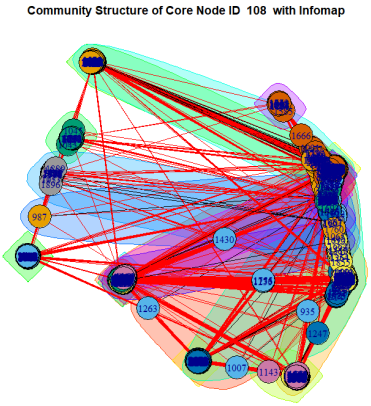
(c) Community structure for Node ID 1 with Infomap Algorithm.



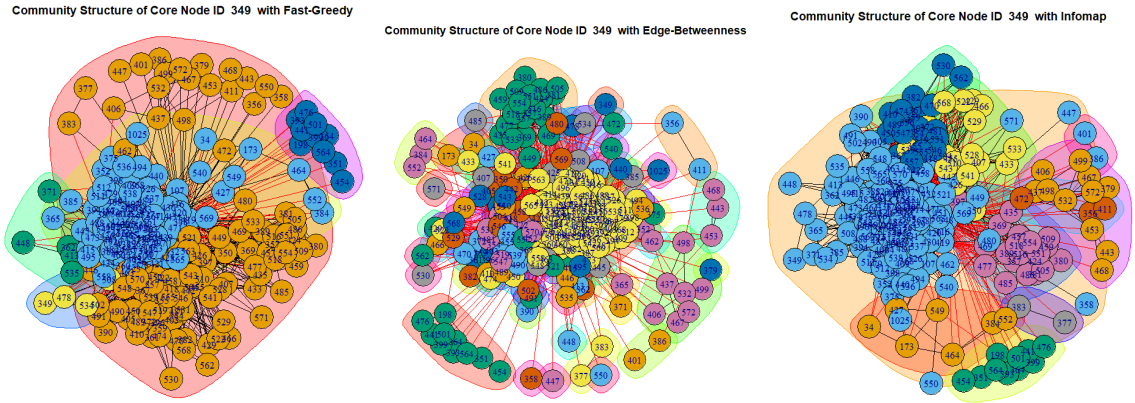
(d) Community structure for Node ID 108 with Fast-Greedy Algorithm.



(e) Community structure for Node ID 108 with Edge-Betweenness Algorithm.



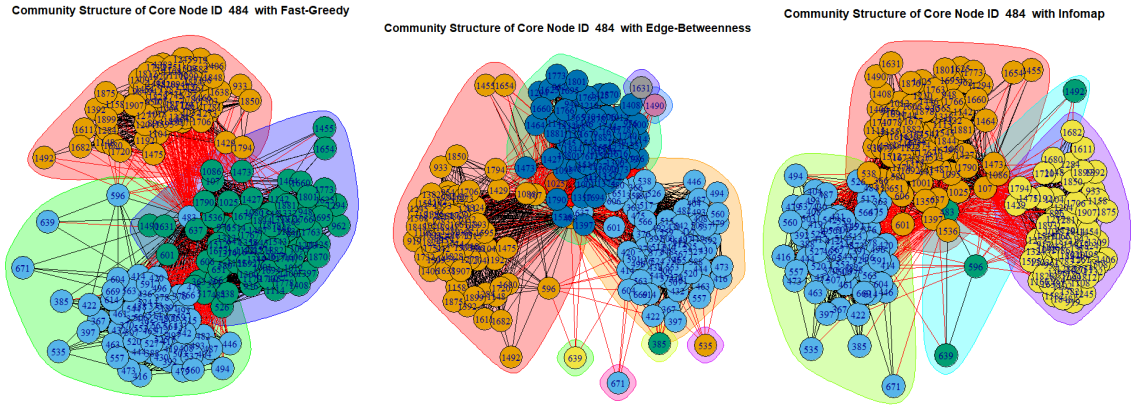
(f) Community structure for Node ID 108 with Infomap Algorithm.



(a) Community structure for Node ID 349 with Fast-Greedy Algorithm.

(b) Community structure for Node ID 349 with Edge-Betweenness Algorithm.

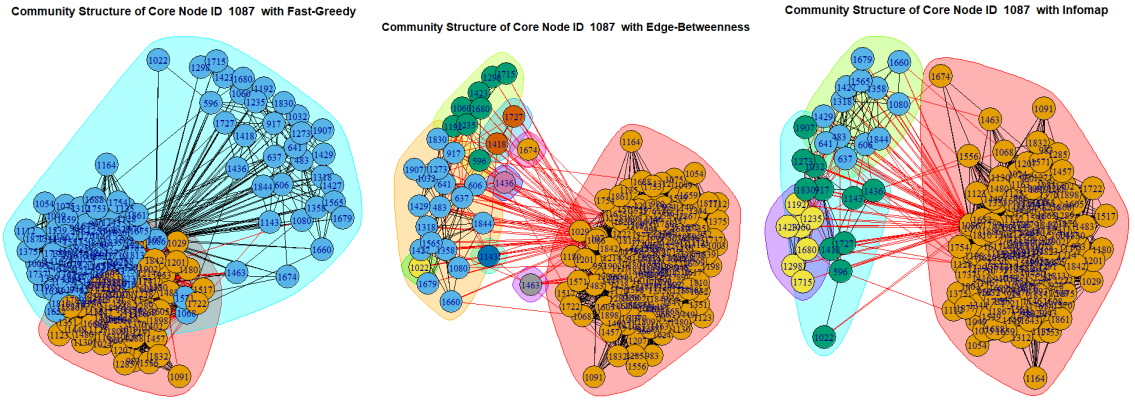
(c) Community structure for Node ID 349 with Infomap Algorithm.



(d) Community structure for Node ID 484 with Fast-Greedy Algorithm.

(e) Community structure for Node ID 484 with Edge-Betweenness Algorithm.

(f) Community structure for Node ID 484 with Infomap Algorithm.



(g) Community structure for Node ID 1087 with Fast-Greedy Algorithm.

(h) Community structure for Node ID 1087 with Edge-Betweenness Algorithm.

(i) Community structure for Node ID 1087 with Infomap Algorithm.

Figure 4: Community Structures for Core Nodes 1, 108, 349, 484, and 1087 using Fast-Greedy, Edge-Betweenness, and Infomap.

In general, the Fast-Greedy algorithm tends to produce higher modularities than the other

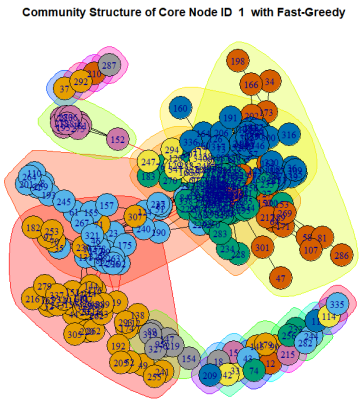
Table 1: Modularity for each core node and each algorithm

	Degree	**Fast-Greedy**	**Edge-Betweenness**	**Infomap**
Node ID 1	346	0.4160	0.3509	0.3883
Node ID 108	1045	0.4359	0.5068	0.5082
Node ID 349	229	0.2502	0.1335	0.2038
Node ID 484	231	0.5070	0.4891	0.5153
Node ID 1087	205	0.1455	0.0276	0.0269

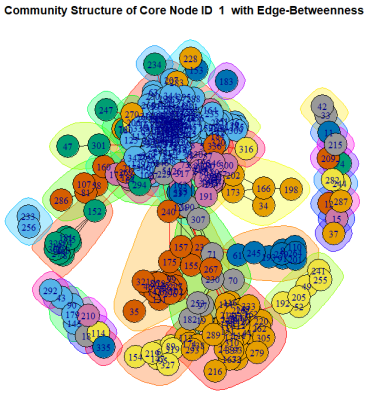
two except for Core Node 108. Infomap tends to outperform Edge-Betweenness or produce a very similar modularity except for Core Node 349, but both ended up with a very low modularity anyway. Since Fast-Greedy performed best for most cases, except for Core Node 108, it could be better at handling smaller networks as Core Node 108 has 1045 nodes while the rest of the graphs have about 200 to 300 nodes.

1.3.2 Community structure with the core node removed

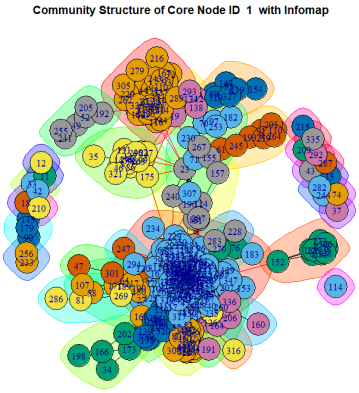
Q10. For each of the core node's personalized network (use same core nodes as Question 9), remove the core node from the personalized network and find the community structure of the modified personalized network. Use the same community detection algorithm as Question 9. Compare the modularity score of 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.



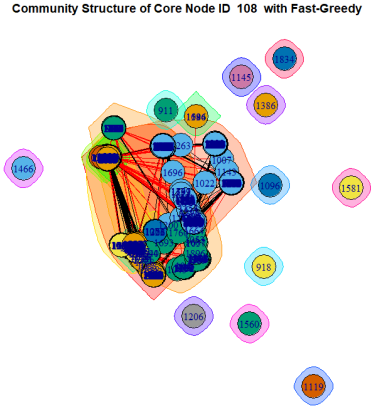
(a) Community structure without Node ID 1 with Fast-Greedy Algorithm.



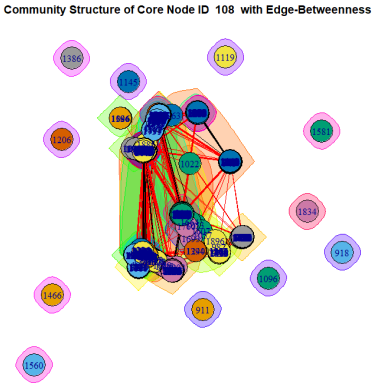
(b) Community structure without Node ID 1 with Edge-Betweenness Algorithm.



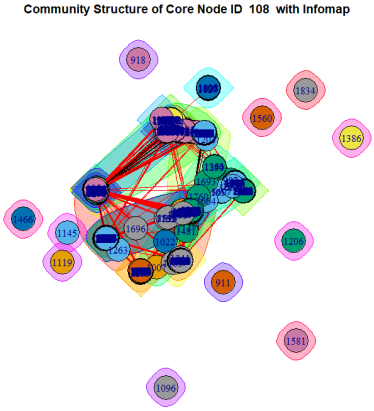
(c) Community structure without Node ID 1 with Infomap Algorithm.



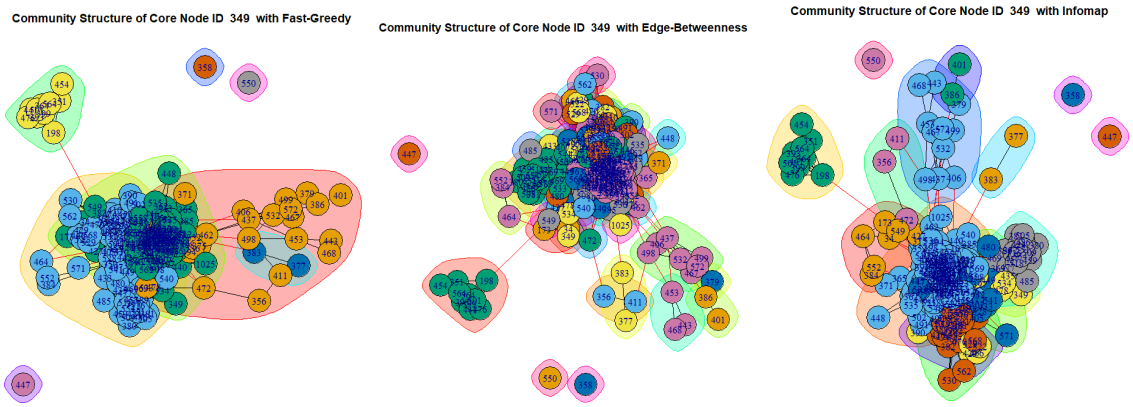
(d) Community structure without Node ID 108 with Fast-Greedy Algorithm.



(e) Community structure without Node ID 108 with Edge-Betweenness Algorithm.



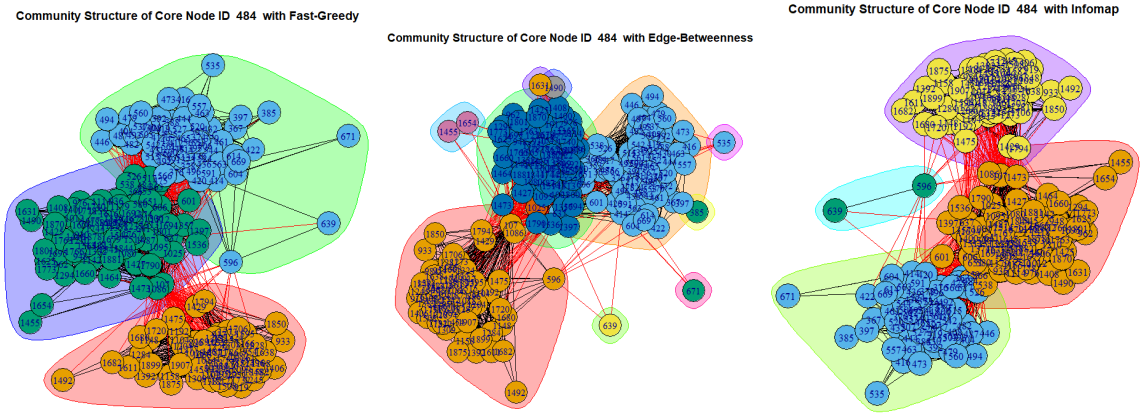
(f) Community structure without Node ID 108 with Infomap Algorithm.



(a) Community structure without Node ID 349 with Fast-Greedy Algorithm.

(b) Community structure without Node ID 349 with Edge-Betweenness Algorithm.

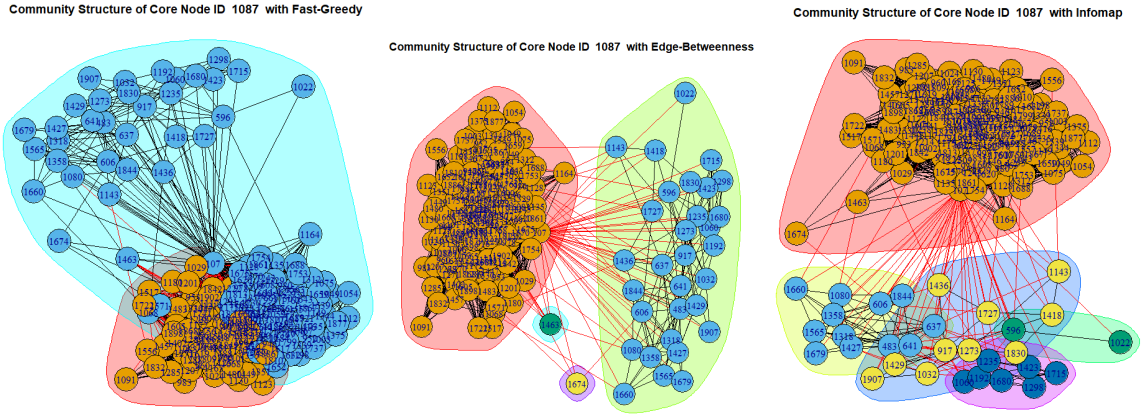
(c) Community structure without Node ID 349 with Infomap Algorithm.



(d) Community structure without Node ID 484 with Fast-Greedy Algorithm.

(e) Community structure without Node ID 484 with Edge-Betweenness Algorithm.

(f) Community structure without Node ID 484 with Infomap Algorithm.



(g) Community structure without Node ID 1087 with Fast-Greedy Algorithm.

(h) Community structure without Node ID 1087 with Edge-Betweenness Algorithm.

(i) Community structure without Node ID 1087 with Infomap Algorithm.

Figure 6: Community Structures without Core Nodes 1, 108, 349, 484, and 1087 using Fast-Greedy, Edge-Betweenness, and Infomap.

Table 2: Modularity for each core node's network without the core node and each algorithm

	Degree	**Fast-Greedy**	**Edge-Betweenness**	**Infomap**
Node ID 1	346	0.4160	0.3509	0.3883
W/O Node ID 1	346	0.4413	0.4144	0.4170
Node ID 108	1045	0.4359	0.5068	0.5082
W/O Node ID 108	1045	0.4581	0.5213	0.5205
Node ID 349	229	0.2502	0.1335	0.2038
W/O Node ID 349	229	0.2457	0.1506	0.2448
Node ID 484	231	0.5070	0.4891	0.5153
W/O Node ID 484	231	0.5342	0.5154	0.5434
Node ID 1087	205	0.1455	0.0276	0.0269
W/O Node ID 1087	205	0.1482	0.0325	0.02737

Table 2 shows that modularity increased across all algorithms when the core node was removed from its own network. This makes sense as the core node served as a connection between communities. Removing it removes a connecting path, making all of the communities less connected to each other, thus increasing the modularity.

1.3.3 Characteristic of nodes in the personalized network

Q11. Write an expression relating the Embeddedness between the core node and a non-core node to the degree of the non-core node in the personalized network of the core node.

The embeddedness of a node is equal to the number of mutually shared nodes it has with the core node. When looking at the personalized networks we've been working with, the only present nodes are the neighbors of the core node. This means that all nodes connect to the core node and will have an embeddedness somewhere between 0 and $N-1$ where N is the degree of the core node. The actual embeddedness between a node and the core node in the personalized network can then be summarized as $D-1$ where D is the degree of the node and $D \leq N$. For example, if a node has a degree of 3, that means it connects to 3 nodes in the personalized network. 1 of the nodes must be the core node. The other 2 nodes must also be connected to the core node, making them mutually shared nodes. This means that the embeddedness would be 2 ($D-1=3-1=2$).

Q12. For each of the core node's personalized network (use the same core nodes as Question 9), plot the distribution histogram of embeddedness and dispersion. In this question, you will have 10 plots.

Based on the provided paper from Backstrom & Kleinberg, we will define the "distance" in dispersion as 1 if the mutual nodes are not directly connected in a subgraph created by removing the core node and the neighboring node we are testing and 0 in all other cases.

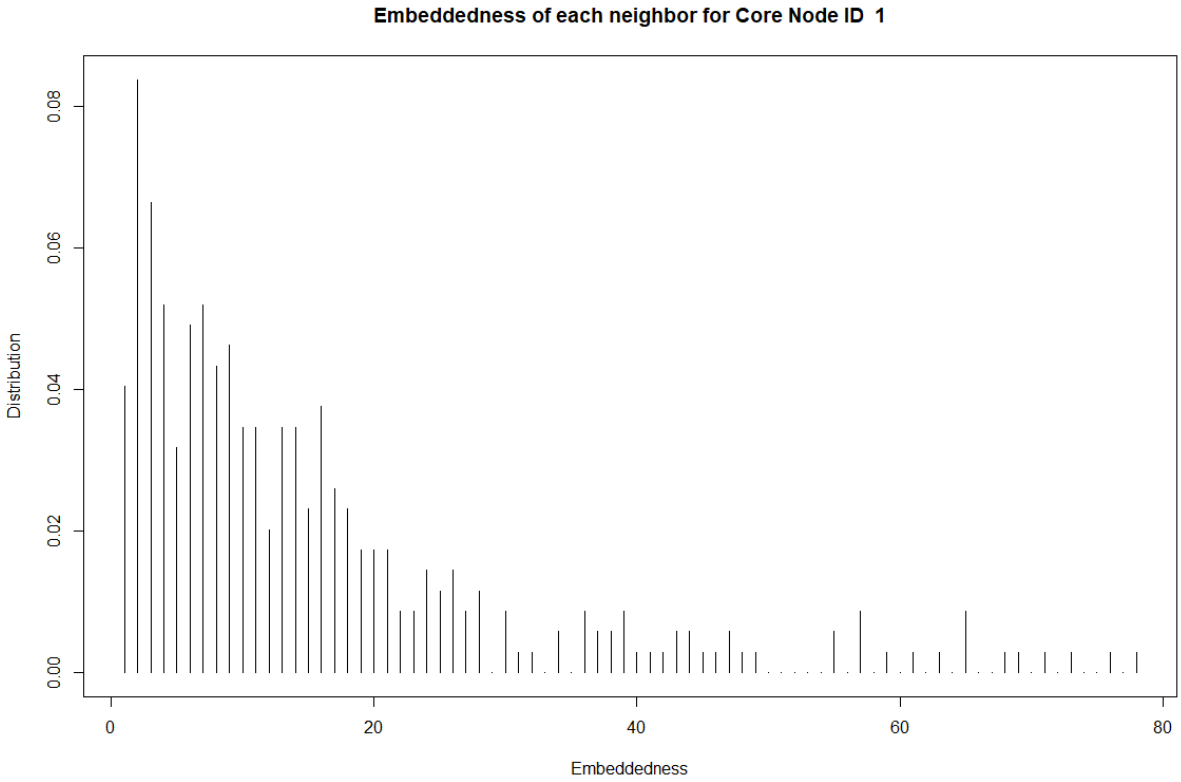


Figure 7: Embeddedness distribution for Core Node 1.

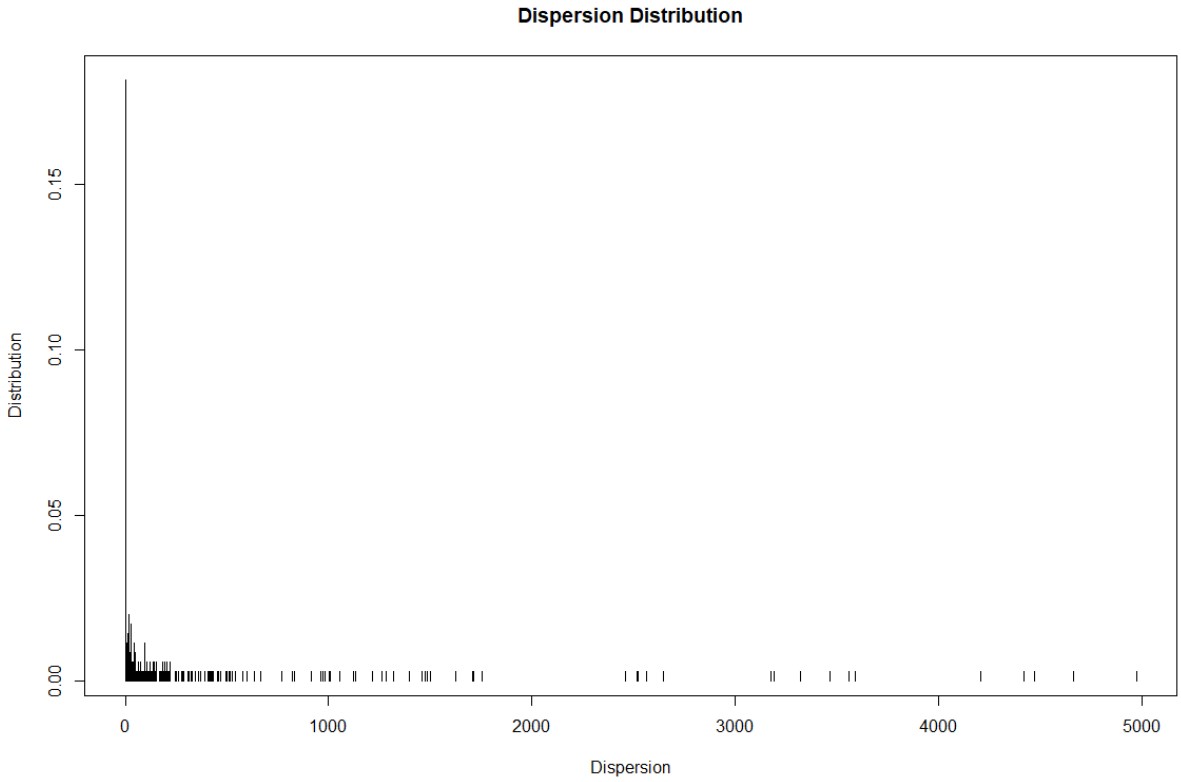


Figure 8: Dispersion distribution for Core Node 1.

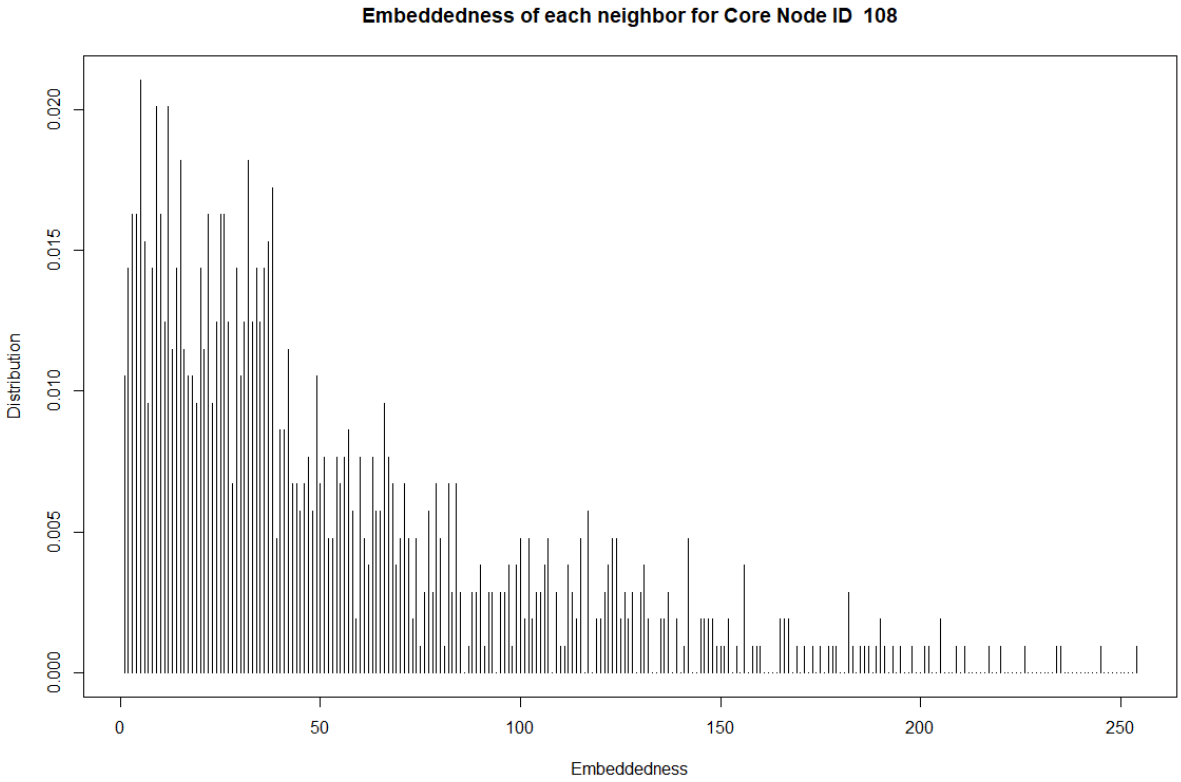


Figure 9: Embeddedness distribution for Core Node 108.

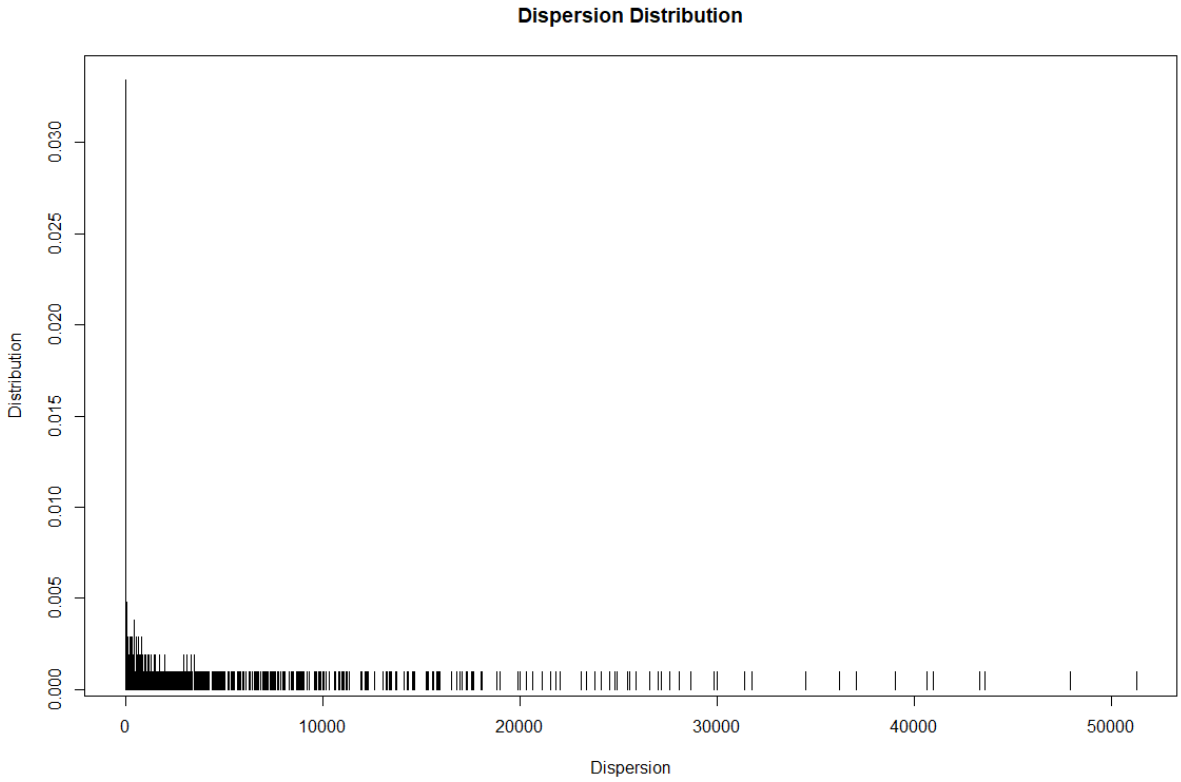


Figure 10: Dispersion distribution for Core Node 108.

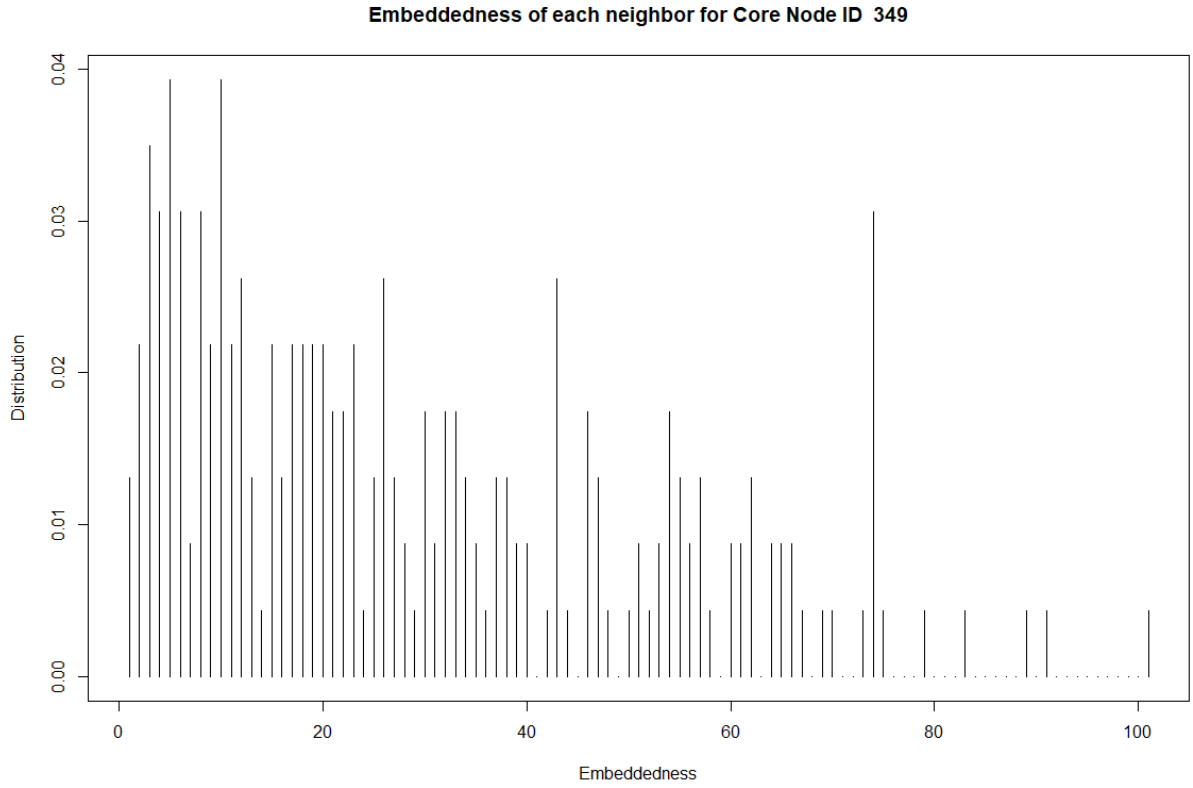


Figure 11: Embeddedness distribution for Core Node 349.

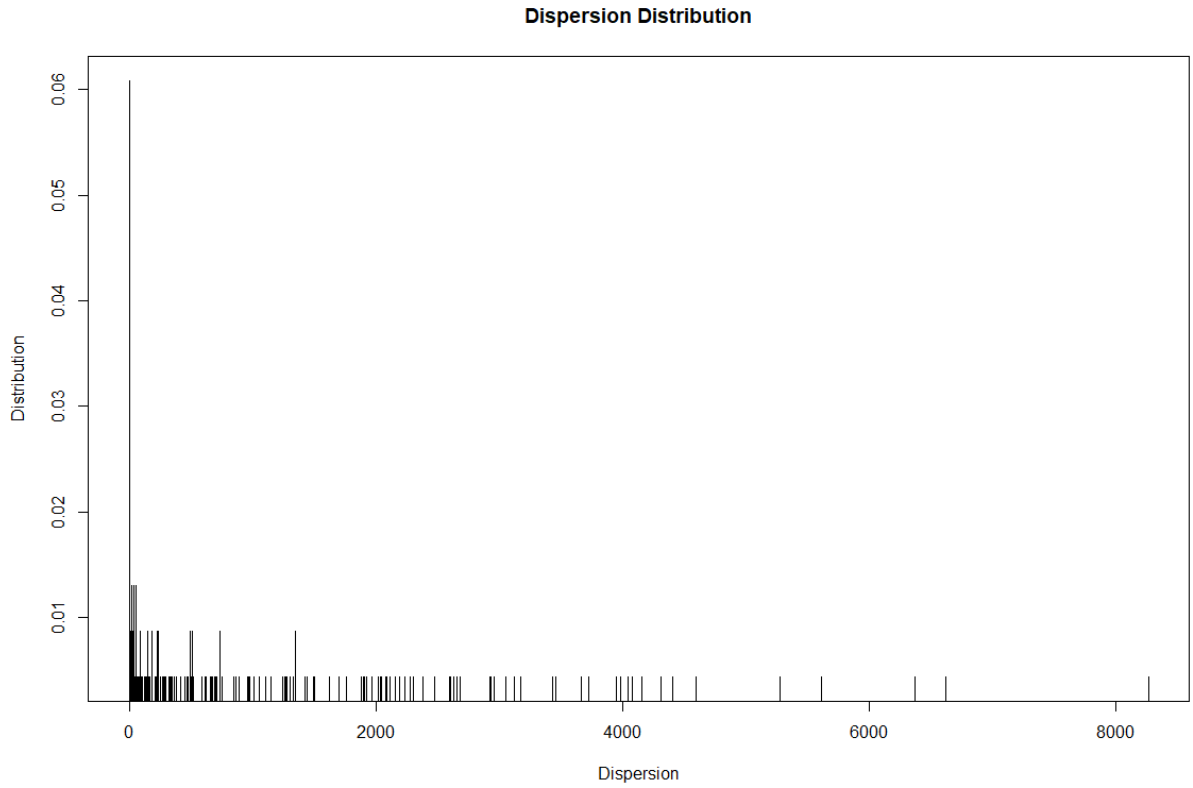


Figure 12: Dispersion distribution for Core Node 349.

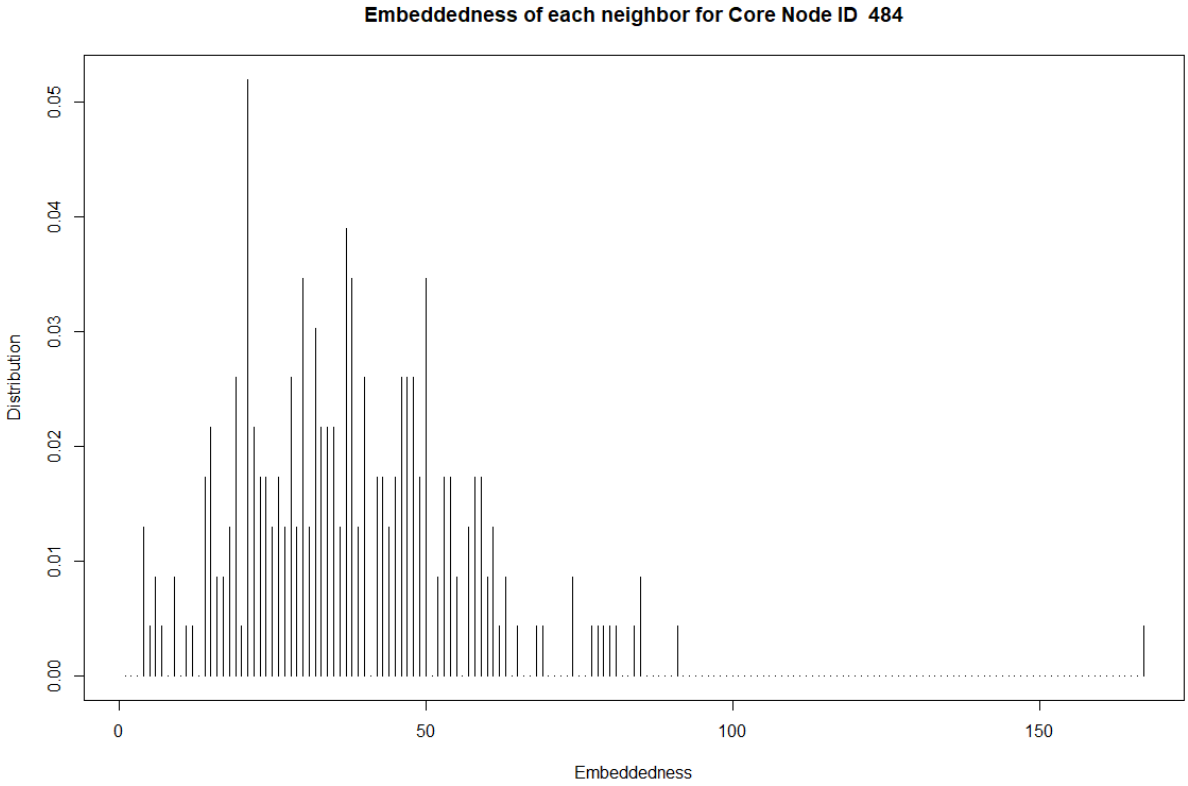


Figure 13: Embeddedness distribution for Core Node 484.

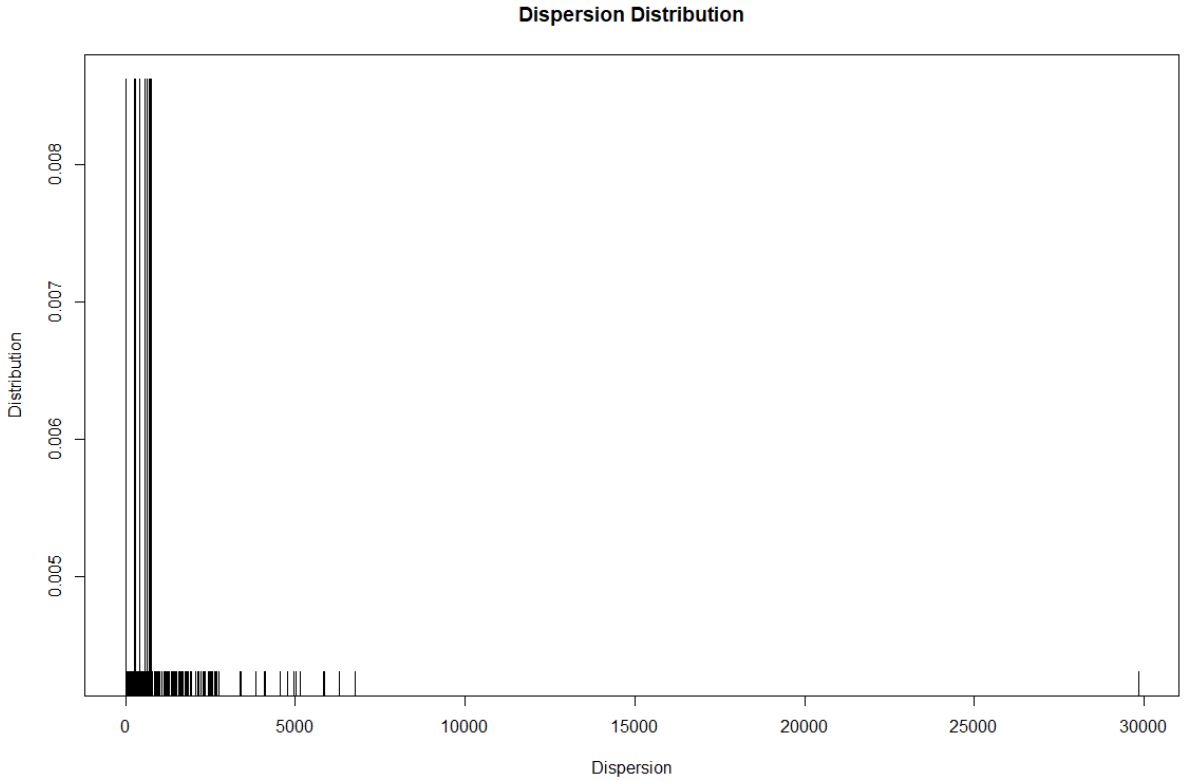


Figure 14: Dispersion distribution for Core Node 484.

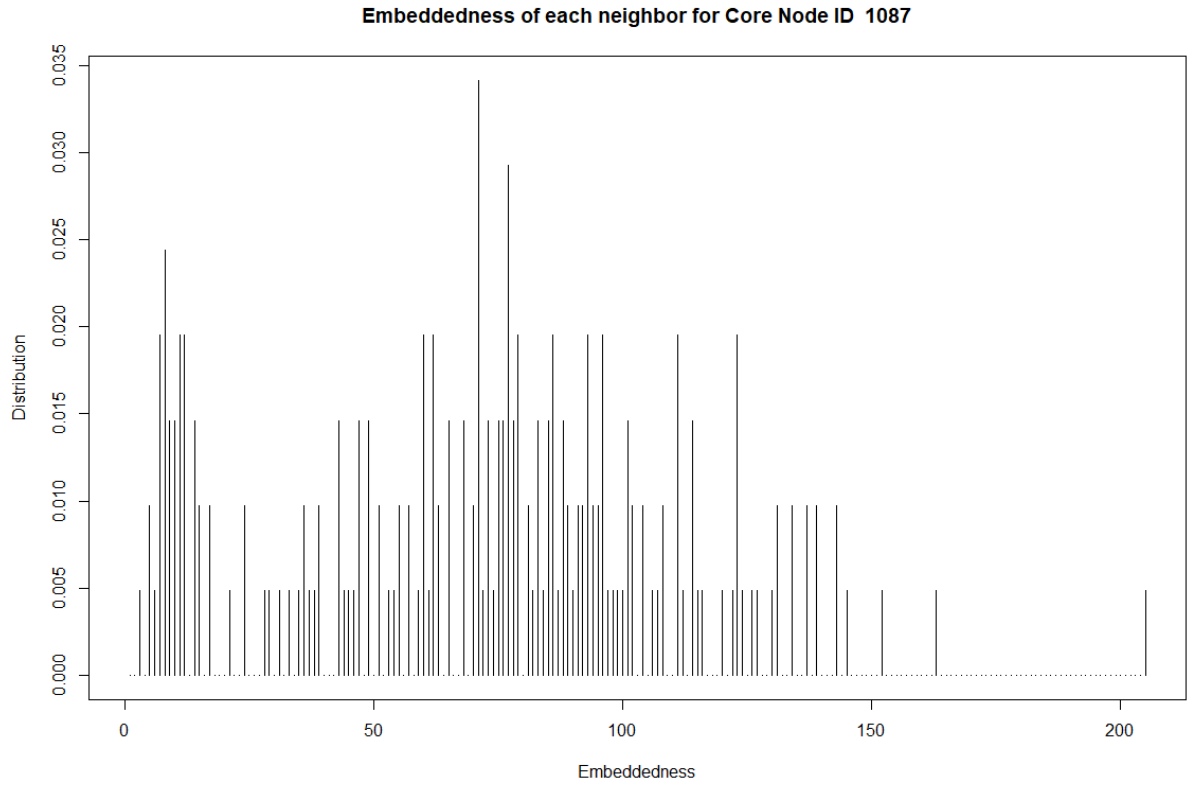


Figure 15: Embeddedness distribution for Core Node 1087.

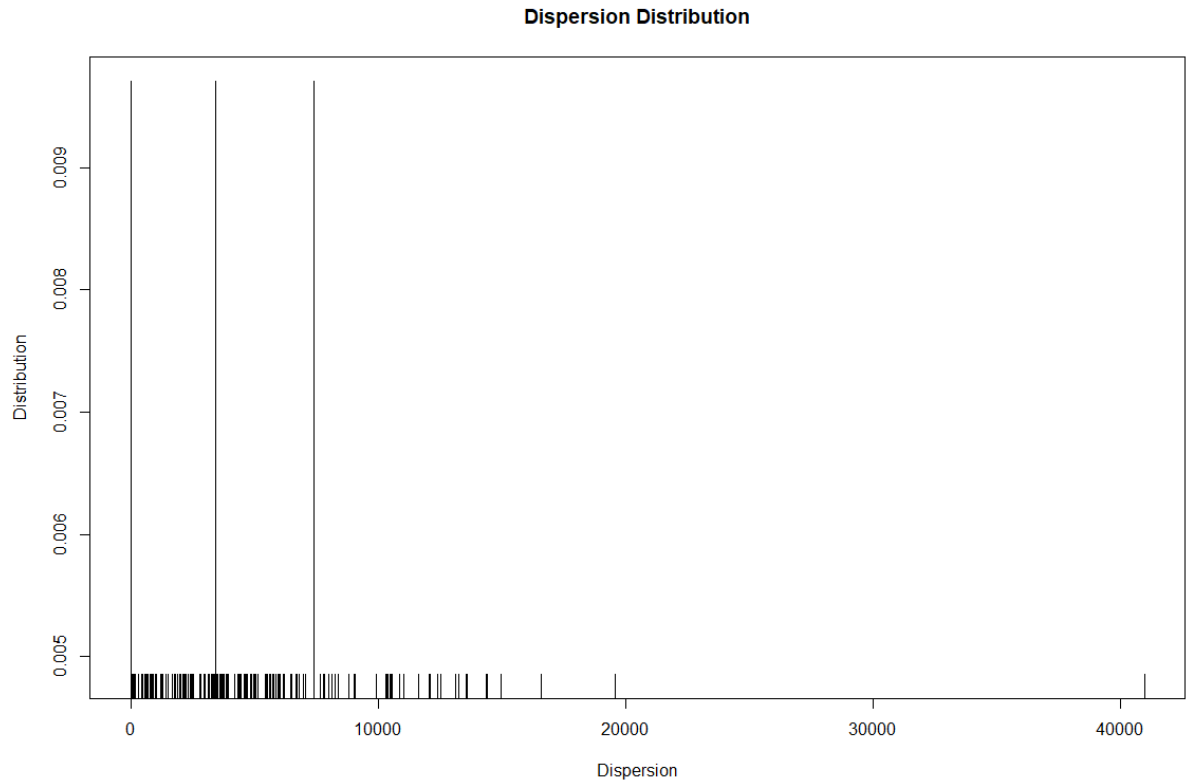


Figure 16: Dispersion distribution for Core Node 1087.

We can see that most of these networks have nodes with a dispersion of 0, indicating that they are well connected enough that, for most nodes, there exists at least a secondary path between any two nodes in the personalized network that does not include the core node or the node we are measuring the dispersion for. Generally, the dispersion probability for a value higher than 0 is fairly low for all of the core nodes but with noteable spikes for core node 1087.

Q13. For each of the core node's personalized network, plot the community structure of the personalized network using colors and highlight the node with maximum dispersion. Also, highlight the edges incident to this node. To detect the community structure, use Fast-Greedy algorithm. In this question, you will have 5 plots.

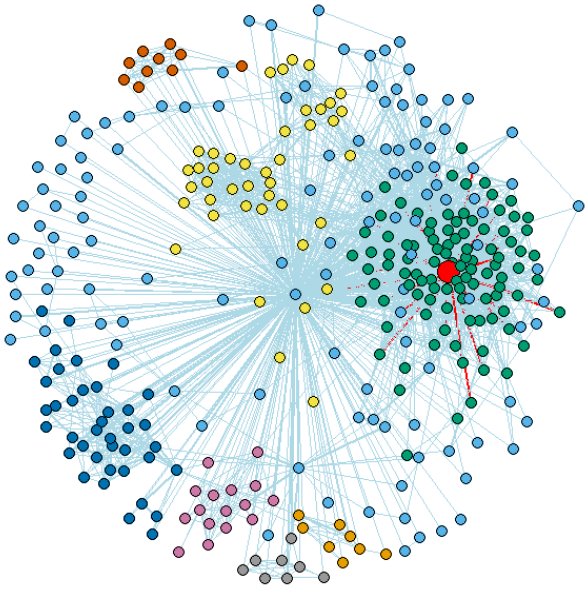


Figure 17: Highlighted Personalized Network for Core Node 1

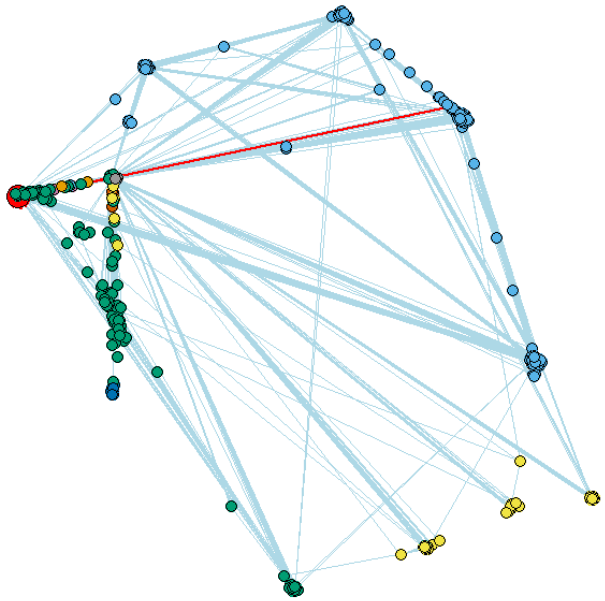


Figure 18: Highlighted Personalized Network for Core Node 108

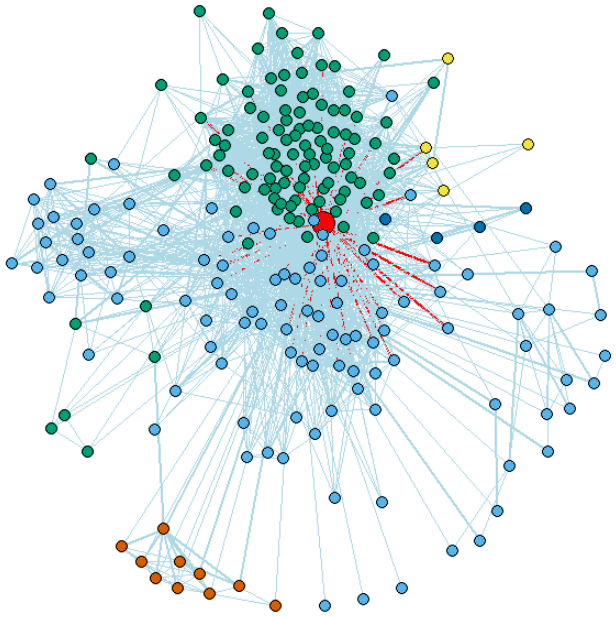


Figure 19: Highlighted Personalized Network for Core Node 349

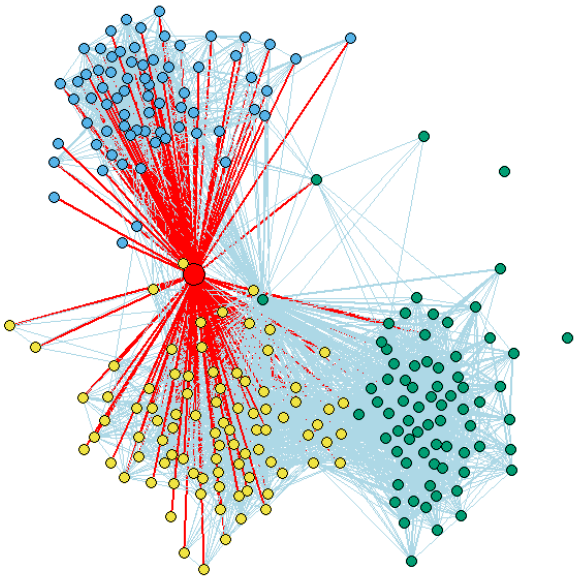


Figure 20: Highlighted Personalized Network for Core Node 484

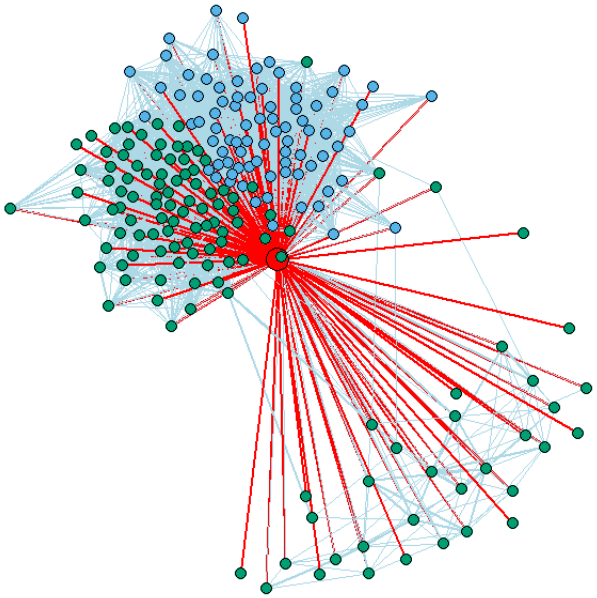


Figure 21: Highlighted Personalized Network for Core Node 1087

For each of the personalized networks in Figures 17-21, the maximum dispersion node and its incident edges are highlighted in red.

Q14. Repeat Question 13, but now highlight the node with maximum embeddedness and the node with maximum dispersion embeddedness (excluding the nodes having zero embeddedness if there are any). Also, highlight the edges incident to these nodes. Report the id of those nodes.

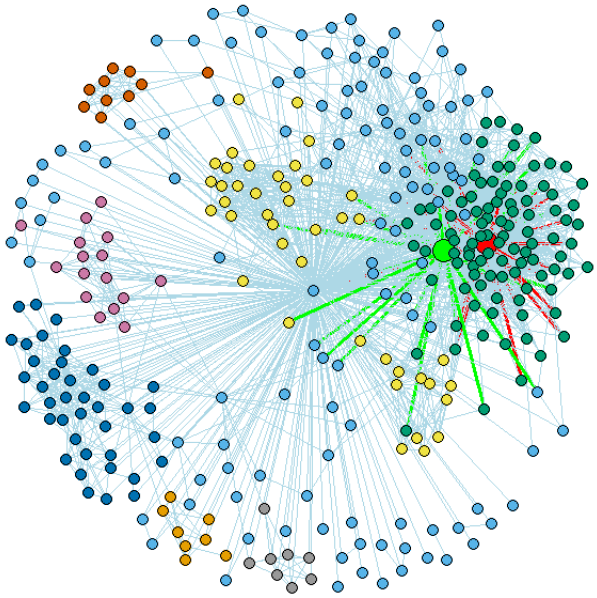


Figure 22: Highlighted Personalized Network for Core Node 1

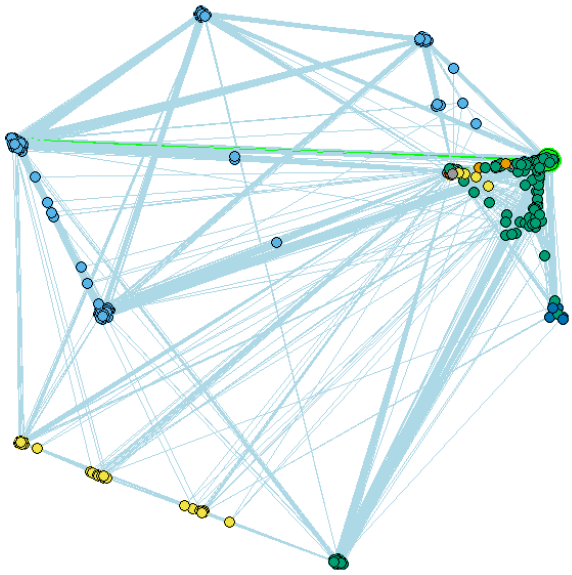


Figure 23: Highlighted Personalized Network for Core Node 108

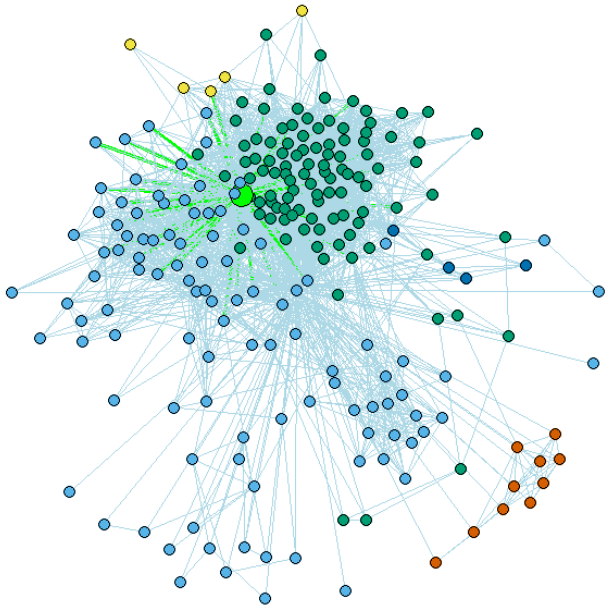


Figure 24: Highlighted Personalized Network for Core Node 349

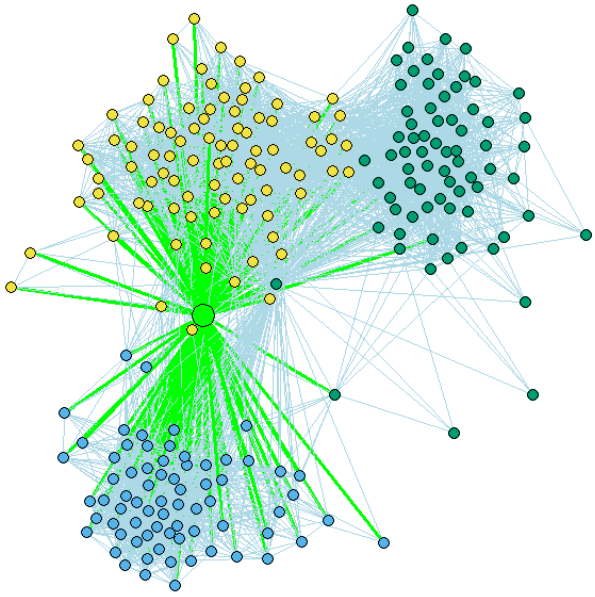


Figure 25: Highlighted Personalized Network for Core Node 484

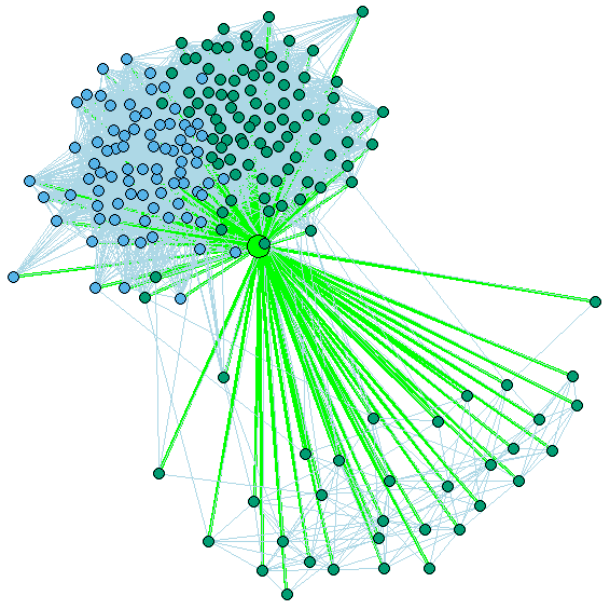


Figure 26: Highlighted Personalized Network for Core Node 1087

In Figures 22-26, the maximum embeddedness node and its edges are highlighted in red and the maximum $\frac{dispersion}{embeddedness}$ node and edges are highlighted in green. As a note, when the two nodes are the same, only the green appears as it covers the red completely.

Table 3: Node Ids for Maximum Embeddedness and $\frac{dispersion}{embeddedness}$

Core Node ID	**Max Embeddedness Node ID**	**Max $\frac{dispersion}{embeddedness}$ Node ID**
1	50	22
108	1713	1713
349	315	315
484	100	100
1087	100	100

We can see from Table 3 that all of the personalized networks except for Core Node 1 have the same node for the maximum embeddedness and $\frac{dispersion}{embeddedness}$.

Q15. Use the plots from Question 13 and 14 to explain the characteristics of a node revealed by each of this measure.

The dispersion of a node indicates how many nodes it shares with the core node that are not connected to each other. In the case of a social network like Facebook, this would mean that a user with a high dispersion has many mutual friends with the core user, but those mutual friends are not friends with each other. A low dispersion would indicate that the user and core user's mutual friends almost all know each other.

The embeddedness is an indication of how many neighbors a given node shares with the core node. If a person is in the same friend group as the core user, then they likely have a high embeddedness as they would share many mutual friends.

For $\frac{dispersion}{embeddedness}$, this is more a representation of how well connected a user and core user's mutual friend group is. For example, if the two users have exactly the same friends and all of those friends are friends with each other, this ratio would be 0. However, if they have the same friends but these friends do not know each other, then the ratio will be very high.

1.4 Friend recommendation in personalized networks

1.4.1 Neighborhood based measure

S_i is the neighbor set of node i in the network S_j is the neighbor set of node j in the network

$$CommonNeighbors(i,j) = |S_i \cap S_j| \tag{1}$$

$$Jaccard(i,j)\frac{|S_i \cap S_j|}{S_i \cup S_j|} \tag{2}$$

$$AdamicAdar(i,j) = \sum_{k \epsilon S_i \cap S_j} \frac{1}{log(|S_k|)} \tag{3}$$

1.4.2 Friend recommendation using neighborhood based measures

1. For each node in the network that is not a neighbor of i, compute the jaccard measure between the node i and the node not in the neighborhood of i.

The numerator in the Jaccard measure is asking for the number of mutual friends two users have which we can provide using the embeddedness. The denominator is asking for the total size of all of the friends of both users combined, being sure to not count mutual friends twice. This can be found by adding up the number of neighbors each of the two nodes have, then subtracting the embeddedness. This can represented as a modified version of the Jaccard measure:

$$Jaccard(i,j) = \frac{E_{i,j}}{|S_i| + |S_j| - E_{i,j}} \tag{4}$$

where $E_{i,j}$ is the embeddedness of node j in relation to node i.

2. Then pick t nodes that have the highest Jaccard measure with node i and recommend these nodes as friends to node i

In this case, we will arbitrarily selecte $t = 10$ to recommend 10 friends to node i.

1.4.3 Creating the list of users

Q16. What is $|N_r|$, i.e. the length of the list N_r?

There are 11 nodes with a degree of 24 in the core node 415's personalized network.

1.4.4 Average accuracy of freidn recommendation algorithm

Q17. Compute the average accuracy of the friend recommendation algorithm that uses: Common Neighbors measure, Jaccard measure, and Adamic Adar measure. Based on the average accuracy values, which friend recommendation algorithm is the best?

Table 4 indicates that all three algorithms perform fairly well. Jaccard tends to be the worst performing algorithm while Common Neighbors and Adamic Adar are fairly close. However, Common Neighbors averages slightly better.

Table 4: Average Accuracy for Each Algorithm

	Average Accuracy
Common Neighbors	0.8419
Jaccard	07939
Adamic Adar	0.8359

2 Google+ Network

Q18. How many personal networks are there with more than 2 circles? The .circles files contain the circles in the network organized as each circle starting with its name followed by it's nodes. Each circle takes up one line in the file. This means we have two possible methods of finding the number of circles in each file. Either search for the names of the circles which are the only strings with letters or count the number of lines in each file. Counting the number of lines will likely be faster as then we do not have to iterate through every single "word" in each file. In this case, there are 57 networks with more than 2 circles.

Q19. QUESTION 19: For the 3 personal networks (node ID given below), plot the in-degree and out degree distribution of these personal networks. Do the personal networks have a similar in and out degree distribution? In this question, you should have 6 plots. 109327480479767108490, 115625564993990145546, 101373961279443806744

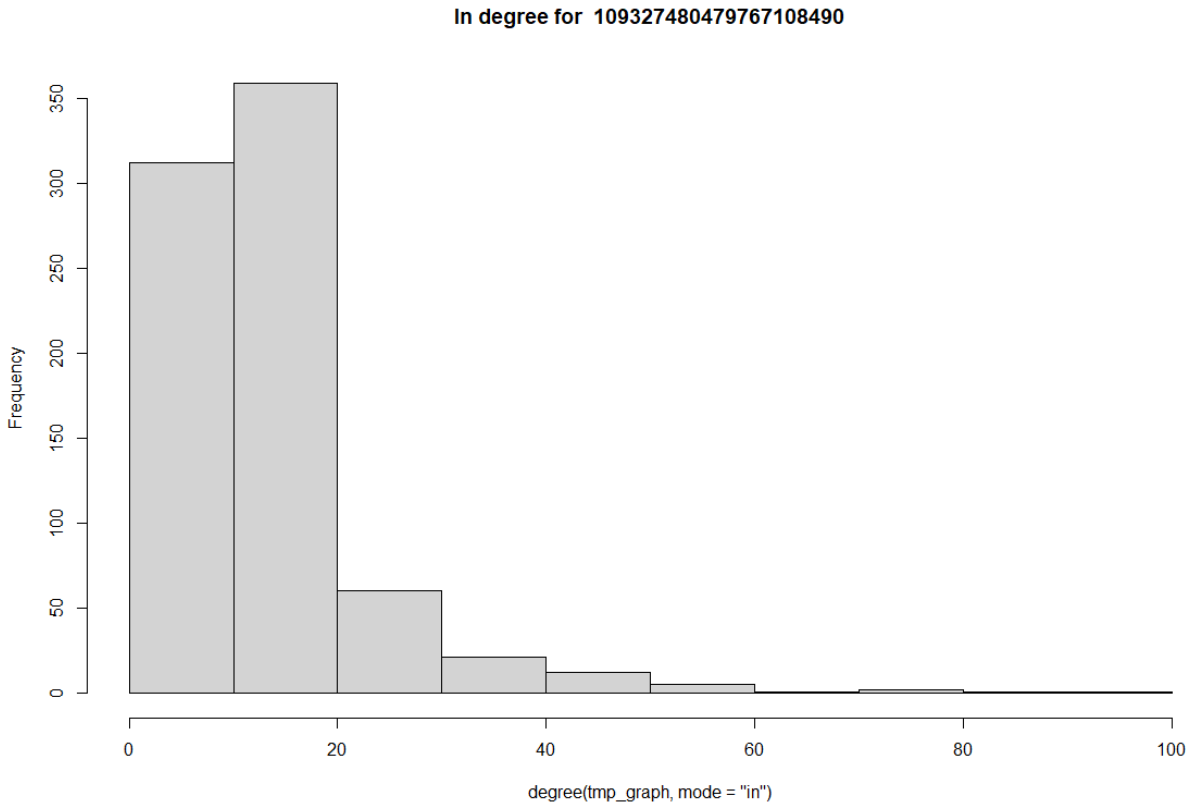


Figure 27: In degree distribution for node 10932748047976710849.

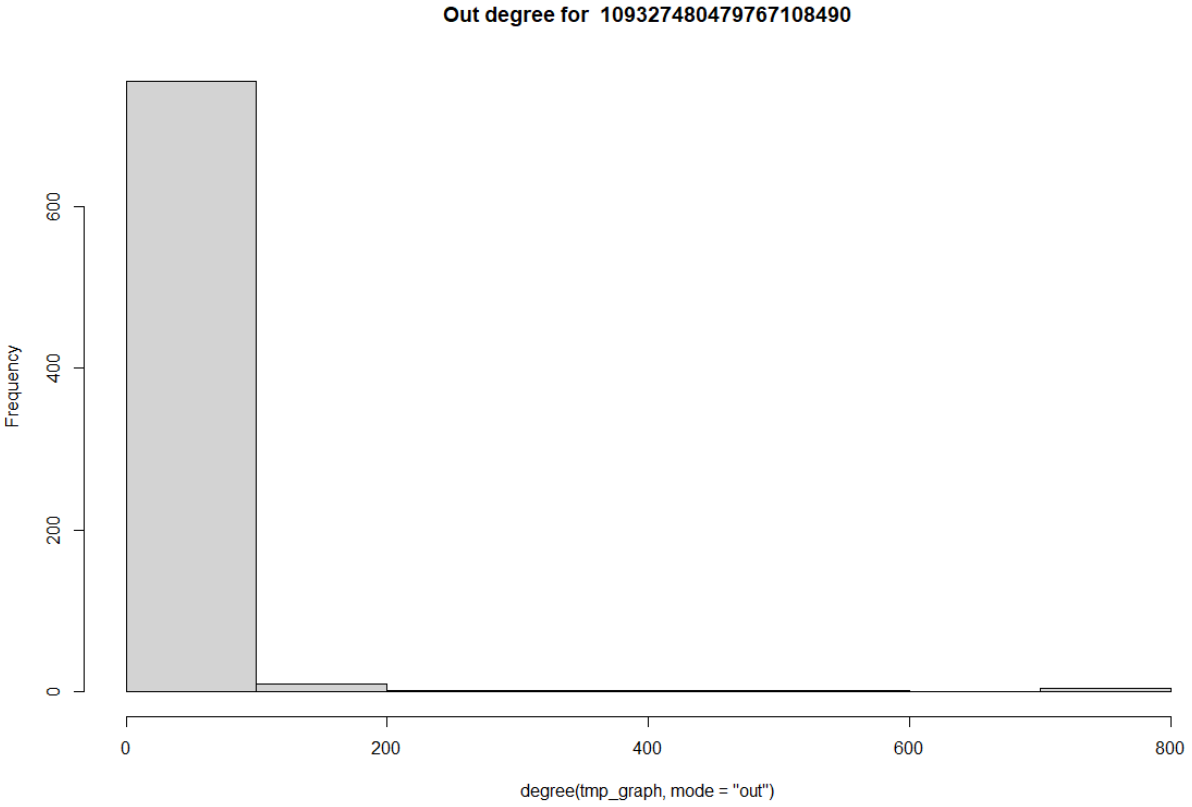


Figure 28: Out degree distribution for node 10932748047976710849.

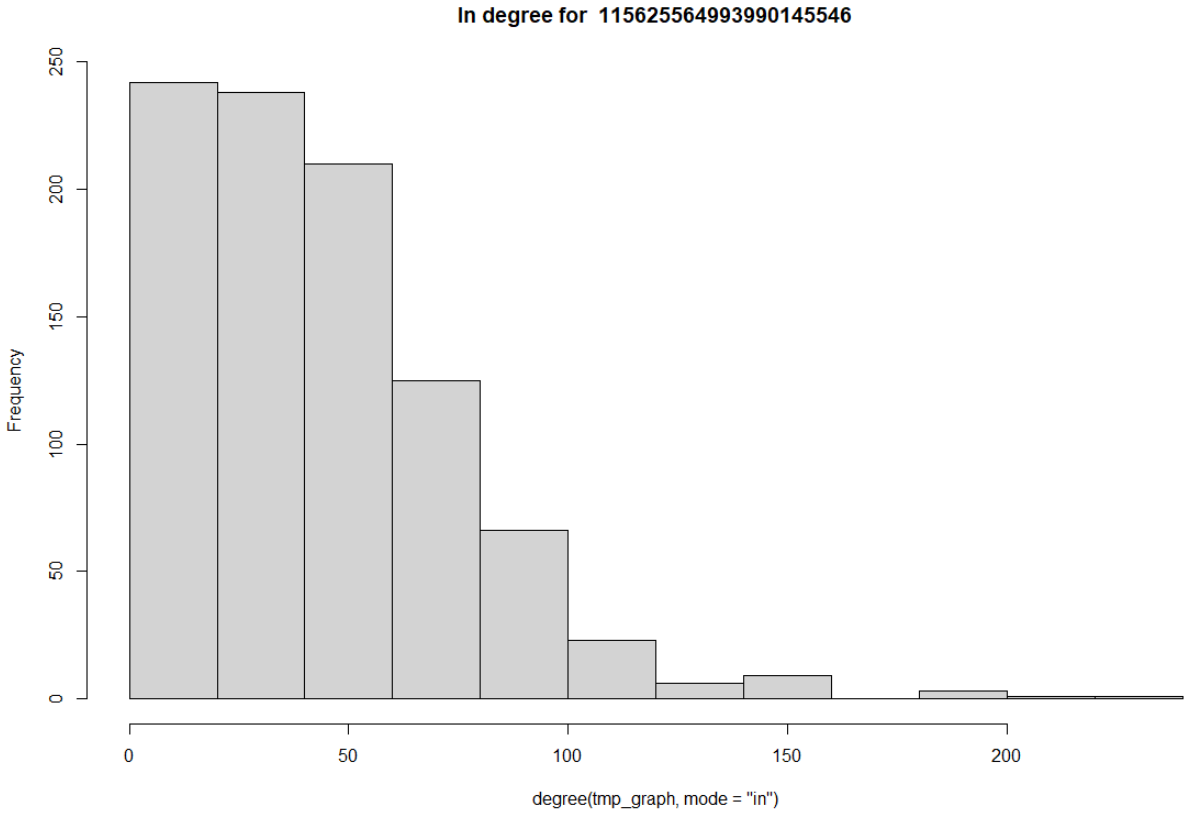


Figure 29: In degree distribution for node 11562556499399014554 6.

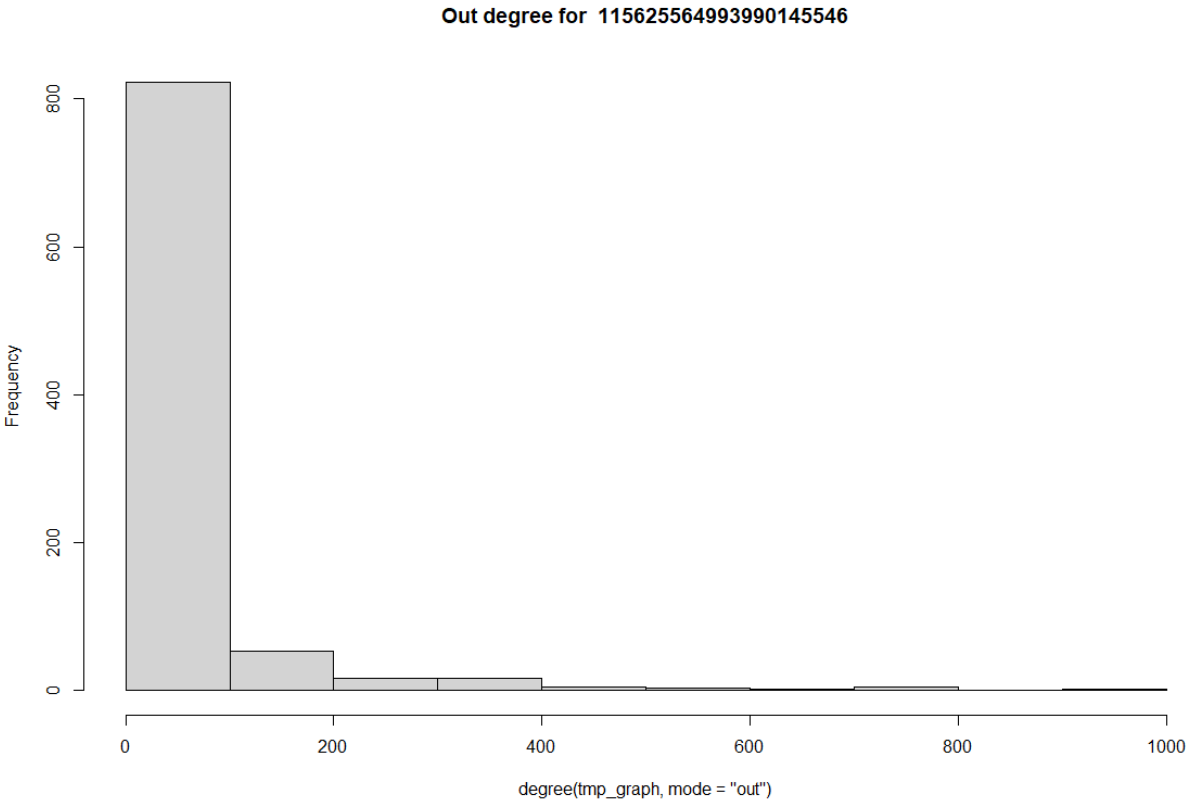


Figure 30: Out degree distribution for node 11562556499399014554 6.

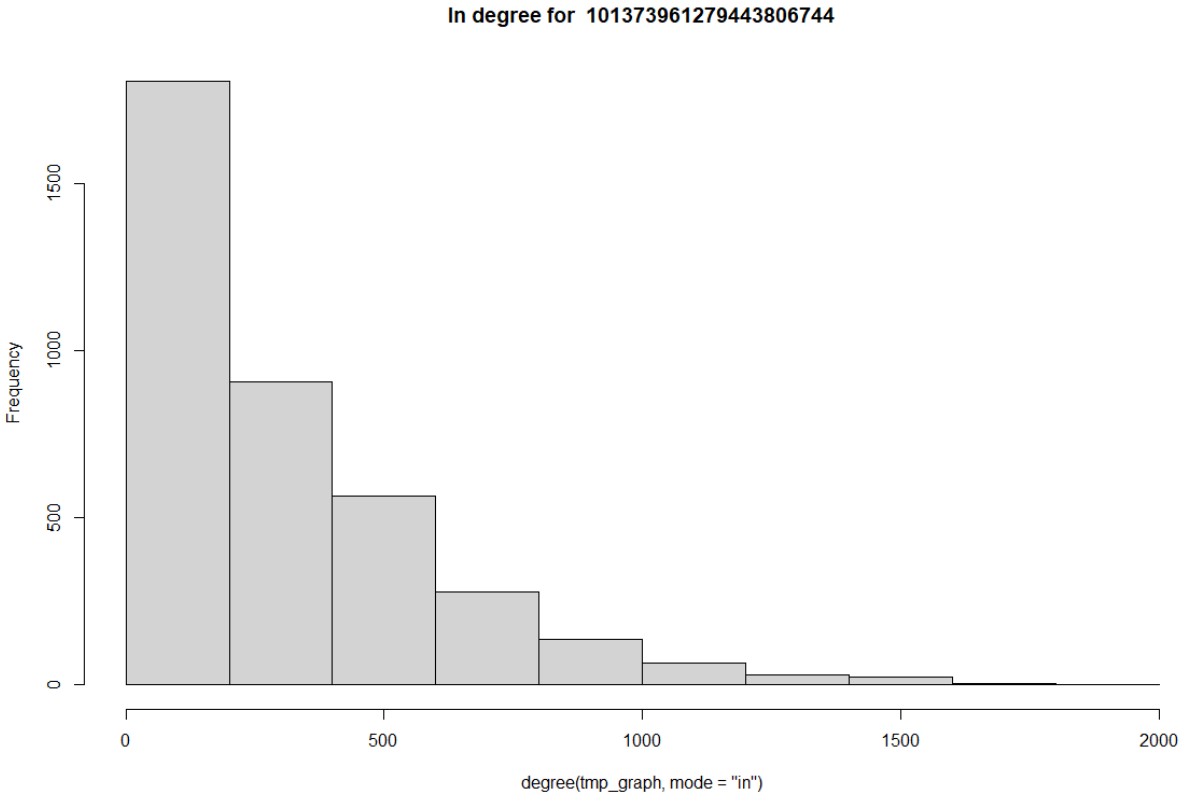


Figure 31: In degree distribution for node 10137396127944380674 4.

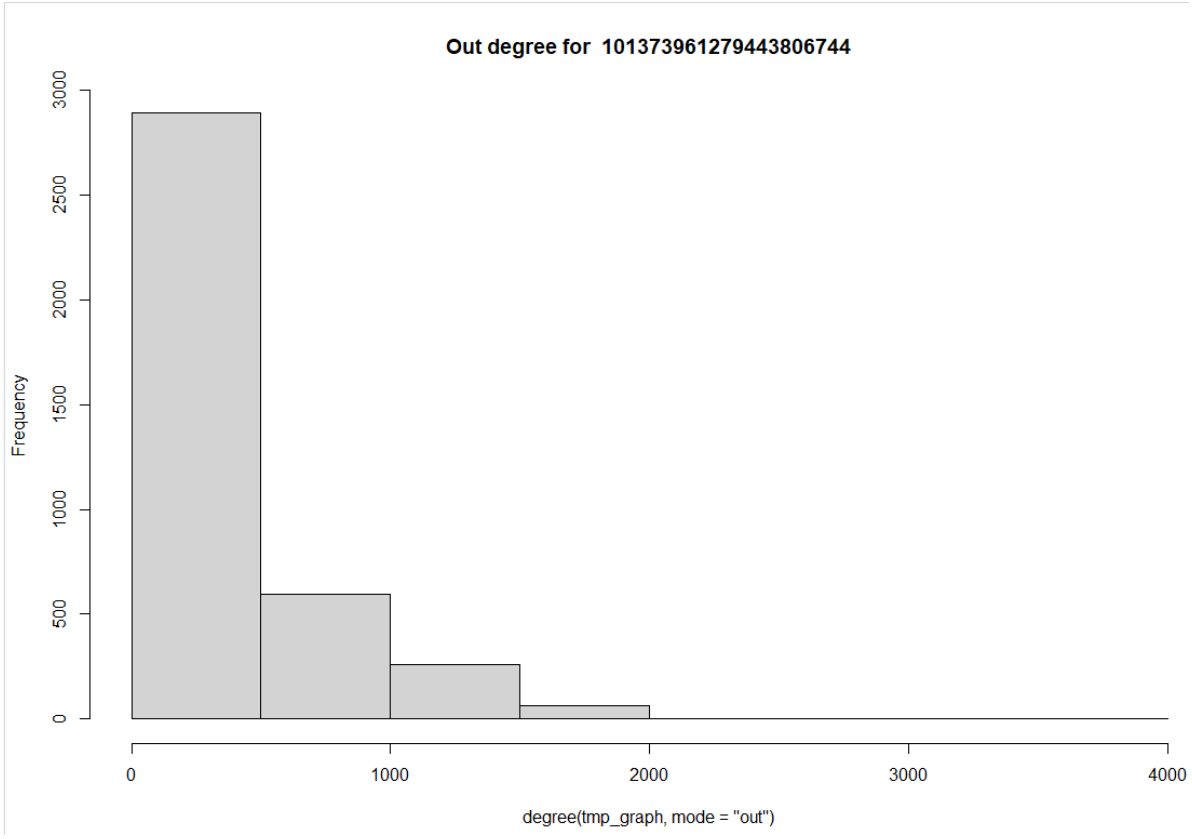


Figure 32: Out degree distribution for node 10137396127944380674 4.

The out degree distribution is very similar for all of the nodes in that any given node is most likely to have a very low degree with only a very small chance of having a high degree.

The in degree distribution varies for each of the networks. The first network, for node 109327480479767108490, shows an in degree distribution that roughly follows a binomial distribution. The second network, for node 115625564993990145546, is a more spread out graph that does not quite follow a binomial distribution. There is a distinct spike in the begining that does not follow the typical "bump" shape of a binomial graph. Lastly, the third network, for node 101373961279443806744, has an in degree distribution that looks more logmarithic where there is a very high chance of any given node having a low degree that rapidly drops off as the degree increases.

2.1 Community structure of personal networks

Q20. For the 3 personal networks picked in Question 19, extract the community structure of each personal network using Walktrap community detection algorithm. Report the modularity scores and plot the communities using colors. Are the modularity scores similar? In this question, you should have 3 plots.

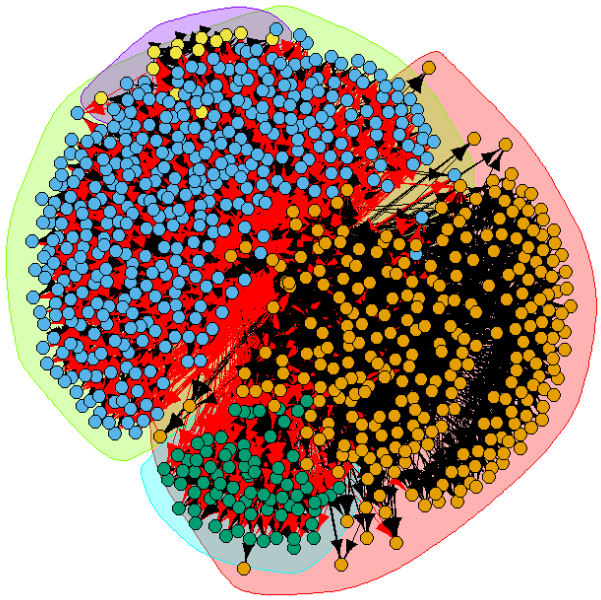


Figure 33: Community structure for node 109327480479767108490.

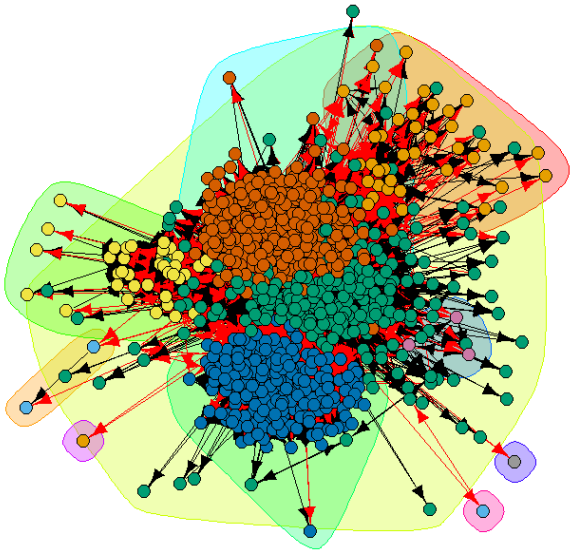


Figure 34: Community structure for node 115625564993990145546.

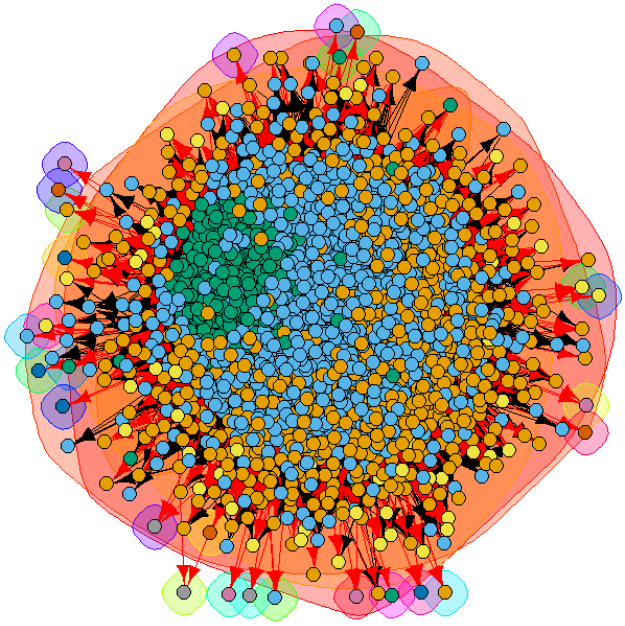


Figure 35: Community structure for for node 101373961279443806744.

Table 5: Modularity for each personalized network

	Modularity
Node ID 109327480479767108490	0.2527654
Node ID 115625564993990145546	0.3194726
Node ID 101373961279443806744	0.1910903

Table 5 shows that the second network has the highest modularity score, followed by the first and then the third network. The networks with higher modularity scores are better connected within communities and/or less well connected between communities while the networks with lower modularity scores are less well connected within communities and/or better connected between communities.

Q21. Based on the expression for h and c, explain the meaning of homogeneity and completeness in words.

Homogeneity indicates if a cluster contains only data points that belong to a single class. Completeness indicates if all the data points of a single class are in the same cluster. In the case of circles and communities, a homogeneity indicates if a given loop is mostly made of nodes from the same community while completeness indicates if most of the nodes in a given community are part of a loop. So if there is a large community that completely contains a single loop that only uses a small number of the nodes in the community, then it would have a high homogeneity score and a low completeness score.

Q22. Compute the h and$c$ values for the community structures of the 3 personal network (same nodes as Question 19). Interpret the values and provide a detailed explanation. Are there negative values? Why?

Table 6: Homogeneity and Completeness for each personalized network

	Homogeneity	**Completeness**
Node ID 109327480479767108490	0.524932571237115	0.549724634098476
Node ID 115625564993990145546	0.106027739162641	0.382183101461206
Node ID 101373961279443806744	0.000374489512941167	0.000859060403838097

Table 6 shows the resulting homogeneity and completeness for each personal network. The first network has a homogeneity and completeness scores both around 0.5. This indicates that the network has some circles either completely or partially contained within its community and a roughly equal amount of circles that are spread over multiple networks. The completeness score indicates that within each community, roughly half of the nodes are in a circle while the rest are not. The next two personalized networks have progressively lower homogeneity and completeness scores indicating that more and more circles are spread over multiple communities and a smaller ratio of nodes within a community are part of a loop.

There are no negative values for either score as they should always return a value between 0 and 1 since they are based on probabilities that add up to 1.

3 Cora dataset

Q23. Idea 1

Use Graph Convolutional Networks [1]. What hyperparameters do you choose to get the optimal performance? How many layers did you choose?

The paper by Kipf and Welling tested semi-supervised node classification on a two-layer GCN using the data, X, and the adjacency matrix, A, of the underlying graph strucutre so we tested

with two layers. The paper also discusses using softmax, and the code associated with the paper indicated other algorithsm and optimizations to use.

The hyperparameters that were tuned were the number of channels, dropout probability, learning rate, and number of epochs. The performance tended to increase before plateauing at an accuracy of around 80% at 20 channels and at 0.5 dropout probability. The learning rate followed a similar pattern except there was a noticeable dropoff once the value reached $1e^{-3}$. Increasing the epoch time primarily affected the length of time it took to train and fit the model with marginal differences past 100. A value of a few hundred would likely suffice for most hardware.

Table 7: GCN Classification Report

	Precision	**Recall**	**F1-Score**
Reinforce_Learning	0.81	0.84	0.82
Theory	0.87	0.92	0.90
Case_Based	0.84	0.82	0.83
Genetic_Algorithms	0.87	0.81	0.83
Probalistic_Methods	0.73	0.75	0.74
Neural_Networks	0.71	0.70	0.71
Rule_Learning	0.68	0.72	0.70
Accuracy			0.81
Macro Avg	0.79	0.79	0.79
Weighted Avg	0.81	0.81	0.81

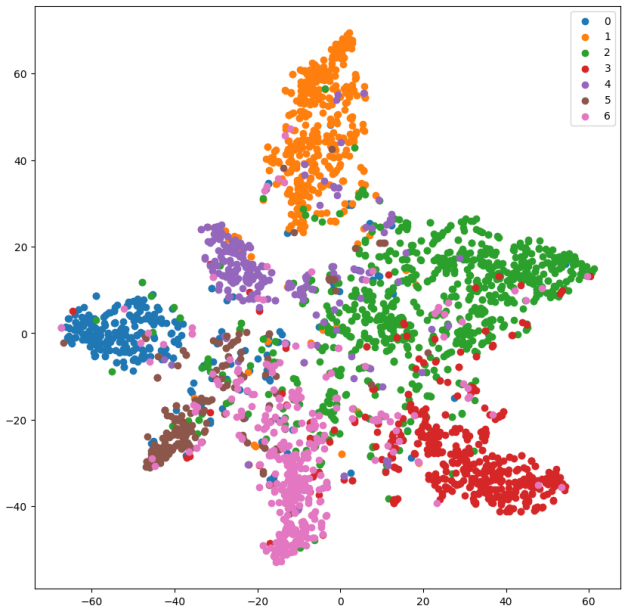


Figure 36: Representation of the hidden layers of the GCN.

Q24. Idea 2

Extract structure-based node features using Node2Vec. Briefly describe how Node2Vec finds node features. Choose your desired classifier (one of SVM, Neural Network, or

Random Forest) and classify the documents using only Node2Vec (graph structure) features. Now classify the documents using only the 1433-dimensional text features. Which one outperforms? Why do you think this is the case? Combine the Node2Vec and text features and train your classifier on the combined features. What is the best classification accuracy you get (in terms of the percentage of test documents correctly classified)?

Table 8: Accuracy for each set of features using Random Forest

	Node2Vec	**Features Only**	**Combination**
Accuracy	0.7348	0.5412	0.7279

Table 8 shows the result of each set of features. Node2Vec provided the highest result while just features provided the lowest accuracy. The two sets of features were combined using averaging which resulted in an accuracy slightly worse than just Node2Vec. Node2Vec provided better results than just the text features because the encoding allows the algorithm to store more information in the vector which allows for better, more accurate predictions.

Q25. Idea 3

Perform a random walk with the following customized properties: (a) teleportation takes the random walker to one of the seed documents of that class (with a uniform probability of 1/20 per seed document). Vary the teleportation probability in {0, 0.1, 0.2}. (b) the probability of transitioning to neighbors is proportional to the cosine similarity between the text features of the current node and the next neighboring node. Repeat part b for every teleportation probability in part a. Run the PageRank only on the GCC. for each seed node, do 1000 random walks. Maintain a class-wise visited frequency count for every unlabeled node. The predicted class for that unlabeled node is the class which lead to maximum visits to that node. Report accuracy and f1 scores.

Table 9: PageRank Teleportation Results

	Accuracy	**Macro F1**	**Weighted F1**
Teleport = 0, No TFIDF	0.38	0.36	0.40
Teleport = 0.1, No TFIDF	0.17	0.14	0.18
Teleport = 0.2, No TFIDF	0.15	0.12	0.16
Teleport = 0, w/ TFIDF	0.39	0.36	0.39
Teleport = 0.1, w/ TFIDF	0.20	0.19	0.20
Teleport = 0.2, w/ TFIDF	0.17	0.17	0.18

Table 9 shows the accuracy and F1 scores for the various runs. The results indicate that the

Table 10: PageRank Teleportation Unvisited Nodes

	# of Unvisited Nodes	**Predicted Node**
Teleport = 0, No TFIDF	0	Neural_Networks
Teleport = 0.1, No TFIDF	6	Neural_Networks
Teleport = 0.2, No TFIDF	58	Neural_Networks
Teleport = 0, w/ TFIDF	0	Neural_Networks
Teleport = 0.1, w/ TFIDF	17	Neural_Networks
Teleport = 0.2, w/ TFIDF	65	Nerual_Networks

algorithm was best at predicting when there was no teleportation with TFIDF. Table 10 shows the number of unvisited nodes and which node was predicted. Nerual_Networks was selected everytime either because the data contains the most of that type of paper, or because there is an error in the coding of the algorithm that causes it to favor transitioning to those types of nodes.