ECE 232E Project 2

Yifan Shu, Chengshun Zhang, Xuan Yang, Yifan Zhang

Introduction

In this project, we will study the various properties of social networks. In the first part of the project, we will study an undirected social network (Facebook). In the second part of the project, we will study a directed social network (Google +).

Part 1: Facebook Network

For this part of the project, we will be using the dataset given as follows: http://snap.stanford.edu/data/egonets-Facebook.html. The Facebook network can be created from the edgelist file (facebook combined.txt)

Question 1.

We read the data from facebook_combined.txt and called the read.graph() function in igraph to construct the facebook network, represented as an undirected graph. We then called is.connect() function and got the result that the graph is connected.

Question 2.

The graph has 4039 nodes and 88234 edges in total. The diameter is 8.

Question 3.

The degree distribution of the network is shown in Fig 1.1. The average degree is 43.69101.

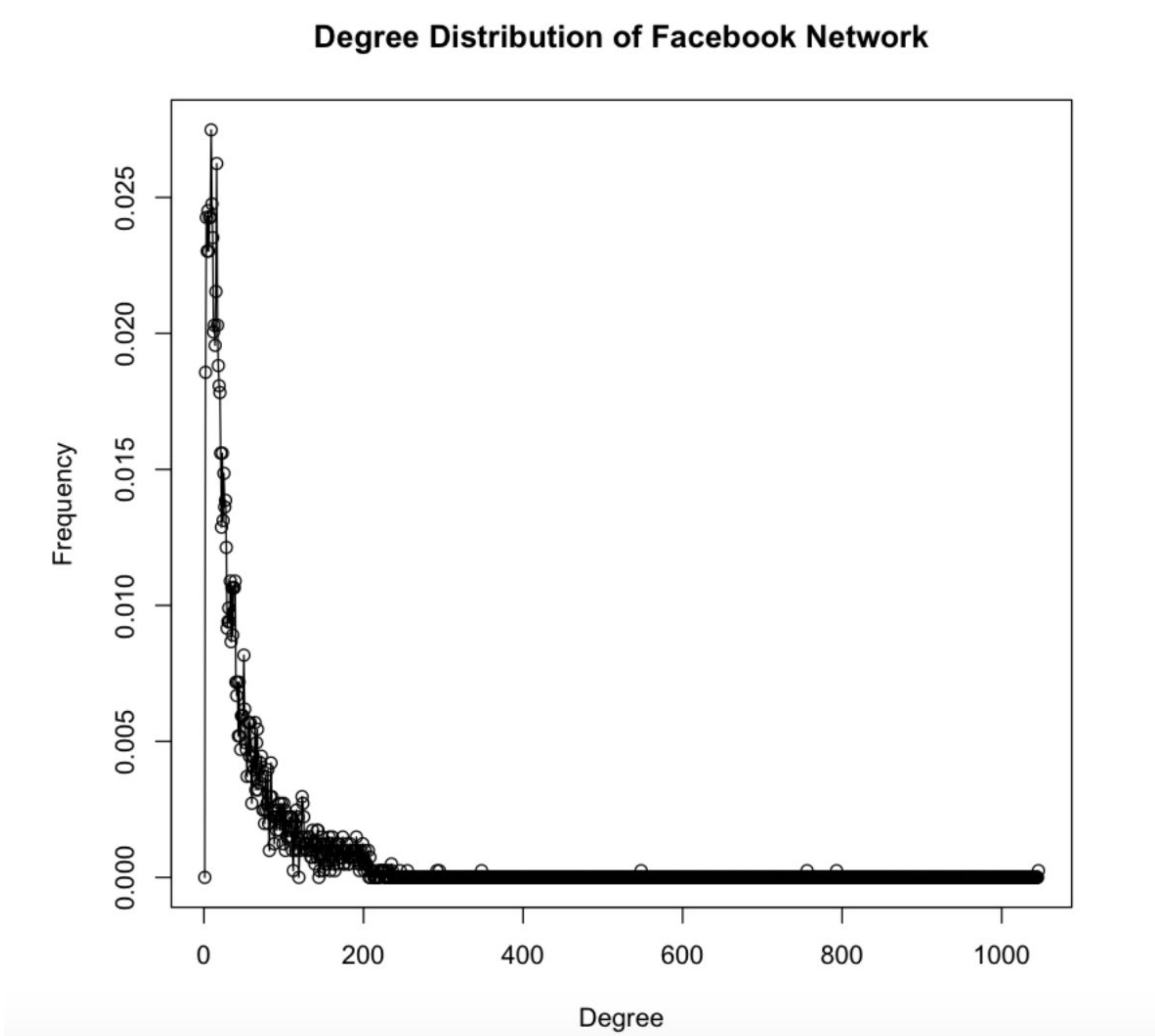

Figure 1.1 Degree Distribution of Facebook Network

Question 4.

We replotted the above graph in a log-log scale and the result is shown in Fig 1.2. We fitted the points using a line and the slope of this line is -1.247526.

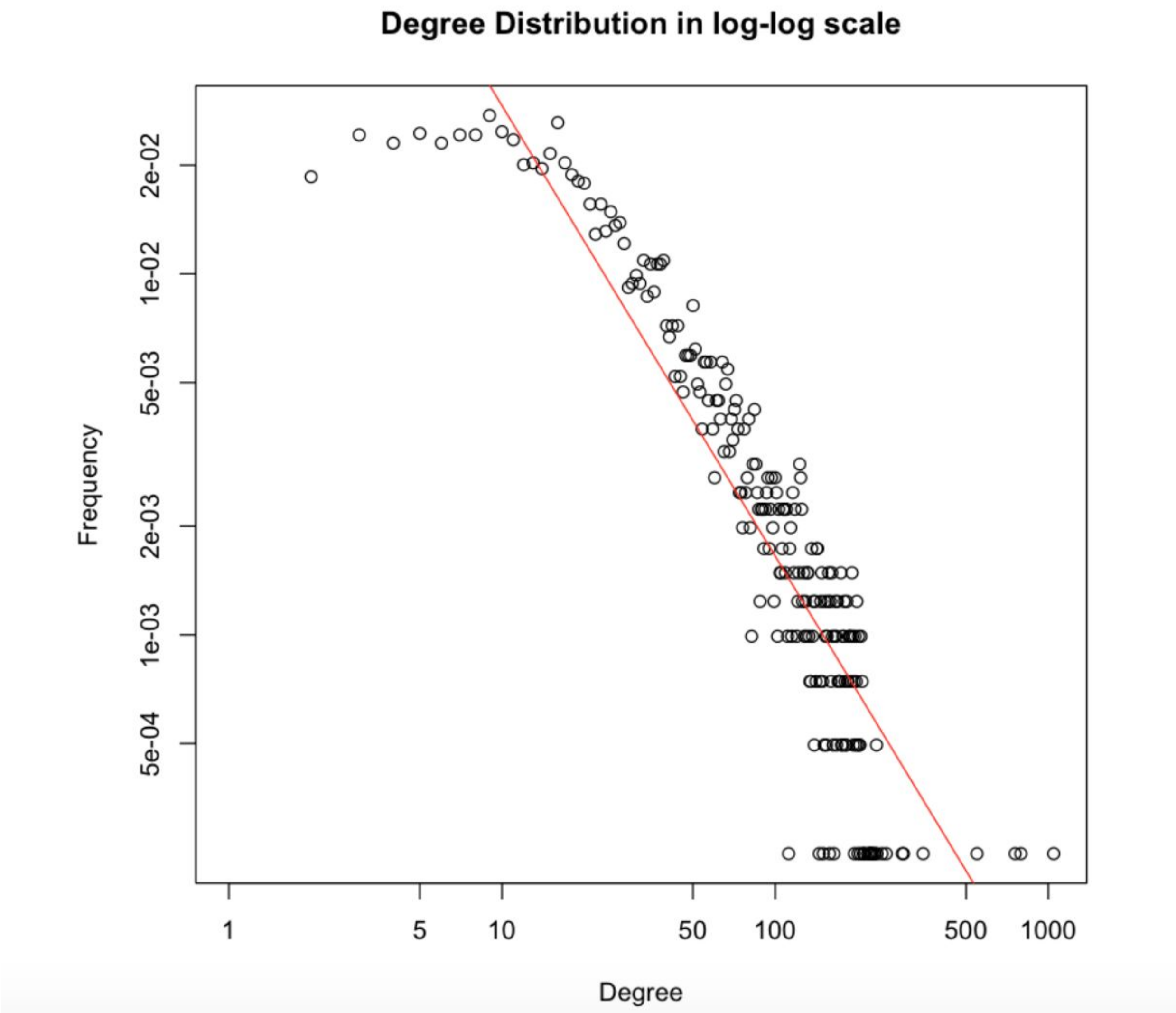

Figure 1.2 Degree Distribution in Log-log Scale

Question 5.

We used the induced.subgraph() function to create the personalized network of the user with ID 1. The subgraph has 348 nodes and 2866 edges.

Question 6.

The diameter of the personalized network in Question 5 is 2.

The diameter of a graph is defined as the longest distance among all the shortest paths between two nodes in the graph. Since the personalized network is centralized by one certain node by definition, i.e., this node is connected to every node in the subgraph, the largest distance between any two nodes is 2. And if there is only one node, the central node, in the graph, the diameter is 0. And if there are at least two nodes, the diameter is 1. So the trivial upper and lower bound of a personalized network is 2 and 0 respectively for one node network. For more than one node network, the upper and lower bound is 2 and 1 respectively.

Question 7.

With the assumption that there are more than three nodes in the network, the upper and lower bound of the network are 2 and 0 respectively. When the diameter of the network to be equal to the upper bound 2, it means that at least one pair of nodes in the network do not have an edge between them. Put it another way, at least two friends of the user with ID 1 are not friends with each other.

When the diameter of the network to be equal to the lower bound 1, it means that every pair of nodes in the network has an edge between them. In other words, every friend of the ID 1 user is also the friend of each other.

Question 8.

We define the core nodes as those that have more than 200 neighbors. With this threshold, there are 40 core nodes in the facebook network and the average degree of core nodes is 279.375.

Question 9.

We first constructed the personalized network of nodes with ID 1, 108, 349, 484 and 1087 and the number of their nodes and edges is shown below in Table 1.1.

We applied Fast-Greedy, Edge-Betweenness and Infomap community detection algorithm to nodes with ID 1, 108, 349, 484 and 1087. The modularity scores of these algorithms are shown in Table 1.2. We plotted the community structures for these nodes with nodes belonging to the same community having the same color and nodes in different community having different color. The result is shown in Fig. 1.3 to Fig. 1.17.

Core Node ID	# Nodes	# Edges
1	348	2866
108	1046	27795
349	230	3441
484	232	4525
1087	206	7409

Table 1.1 The Number of Nodes and Edges of Core Nodes' Personalized Networks

Core Node ID	Modularity Score for Fast-Greedy Algorithm	Modularity Score for Edge-Betweenness Algorithm	Modularity Score for Infomap Community Detection Algorithm
1	0.4131014	0.3533022	0.3891185
108	0.4359294	0.5067549	0.5082492
349	0.2517149	0.133528	0.203753
484	0.5070016	0.4890952	0.5152788
1087	0.1455315	0.02762377	0.02690662

Table 1.2 The Modularity Score of Different Algorithms for Different Core Nodes' Personalized Networks

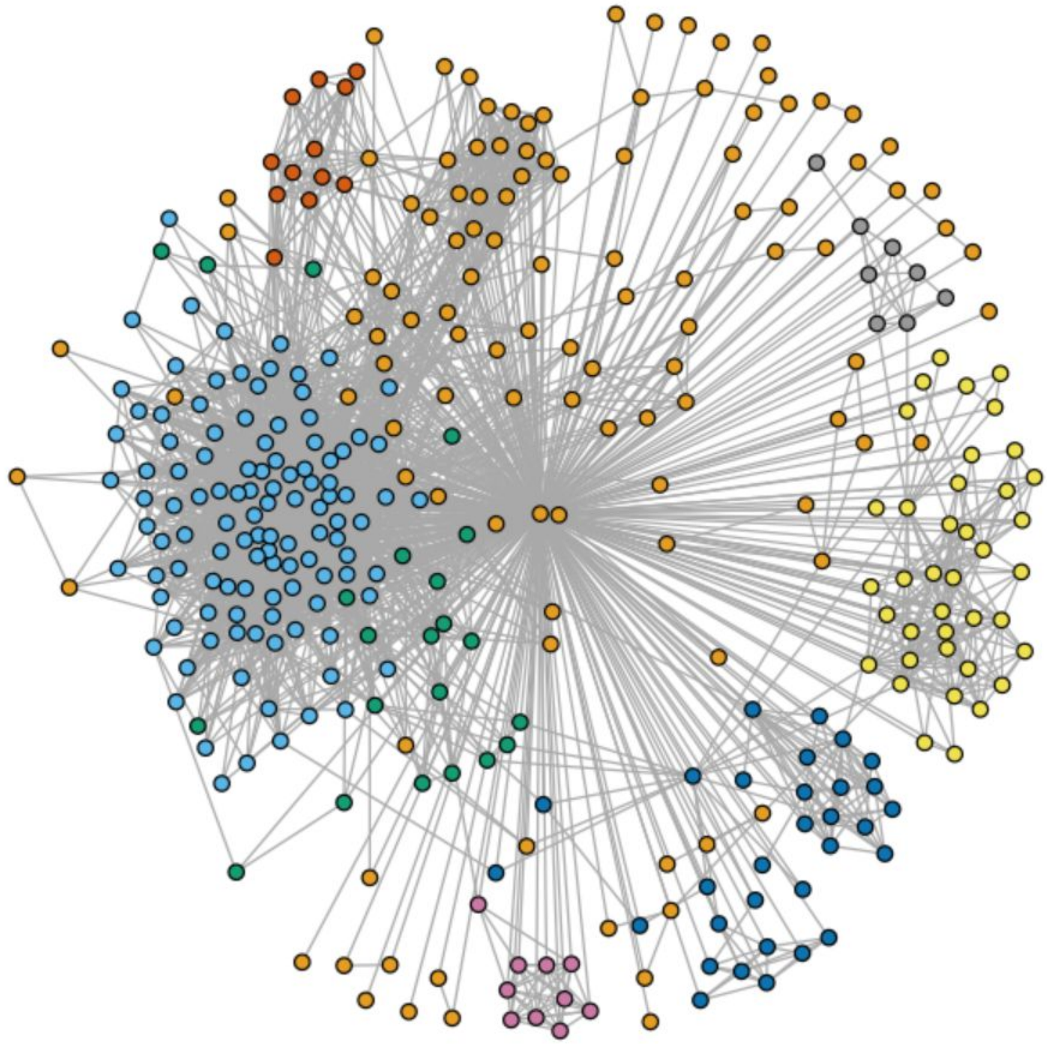

Figure 1.3 Community Structure using Fast-Greedy Algorithm for Node 1

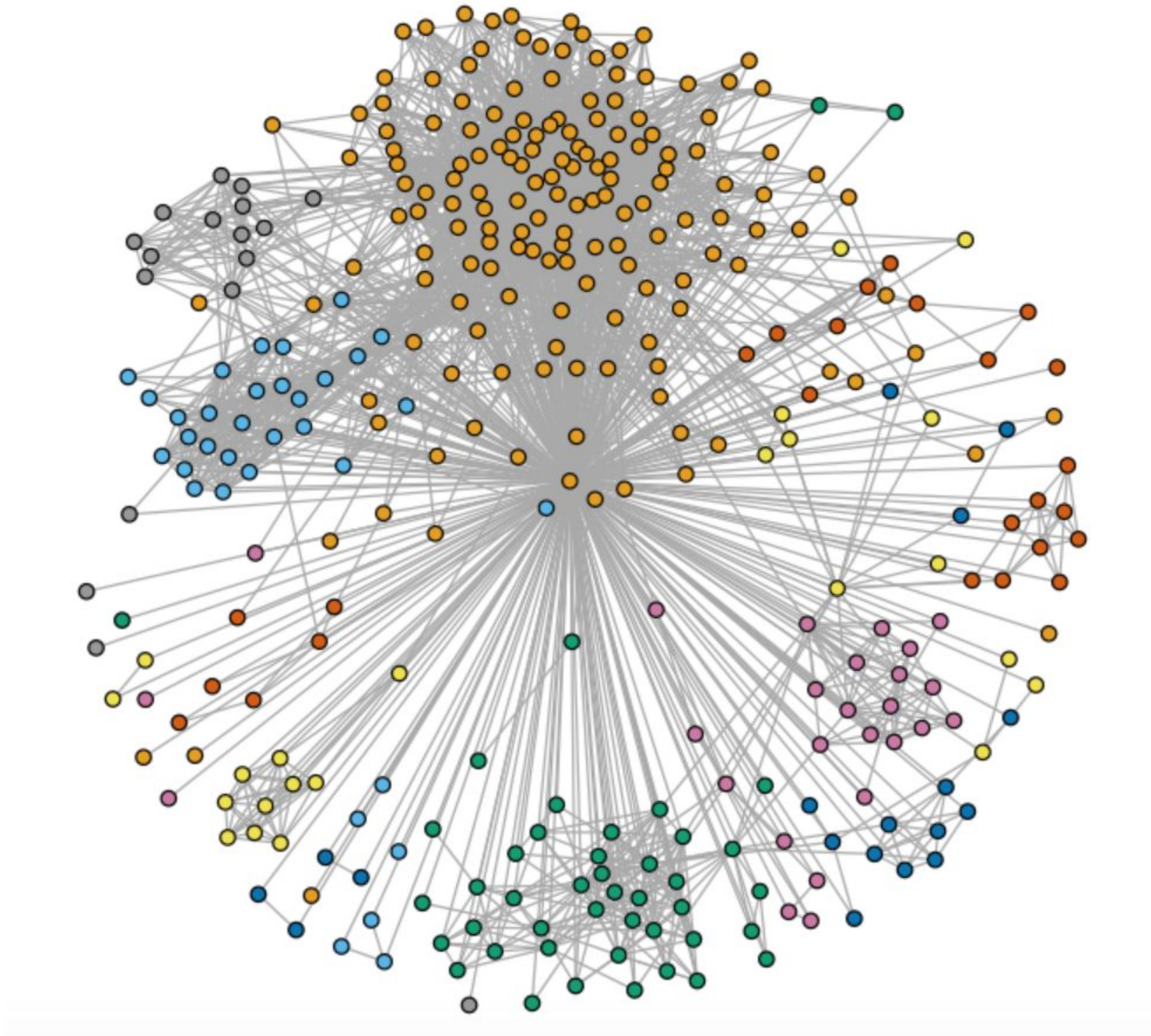

Figure 1.4 Community Structure using Edge-Between Algorithm for Node 1

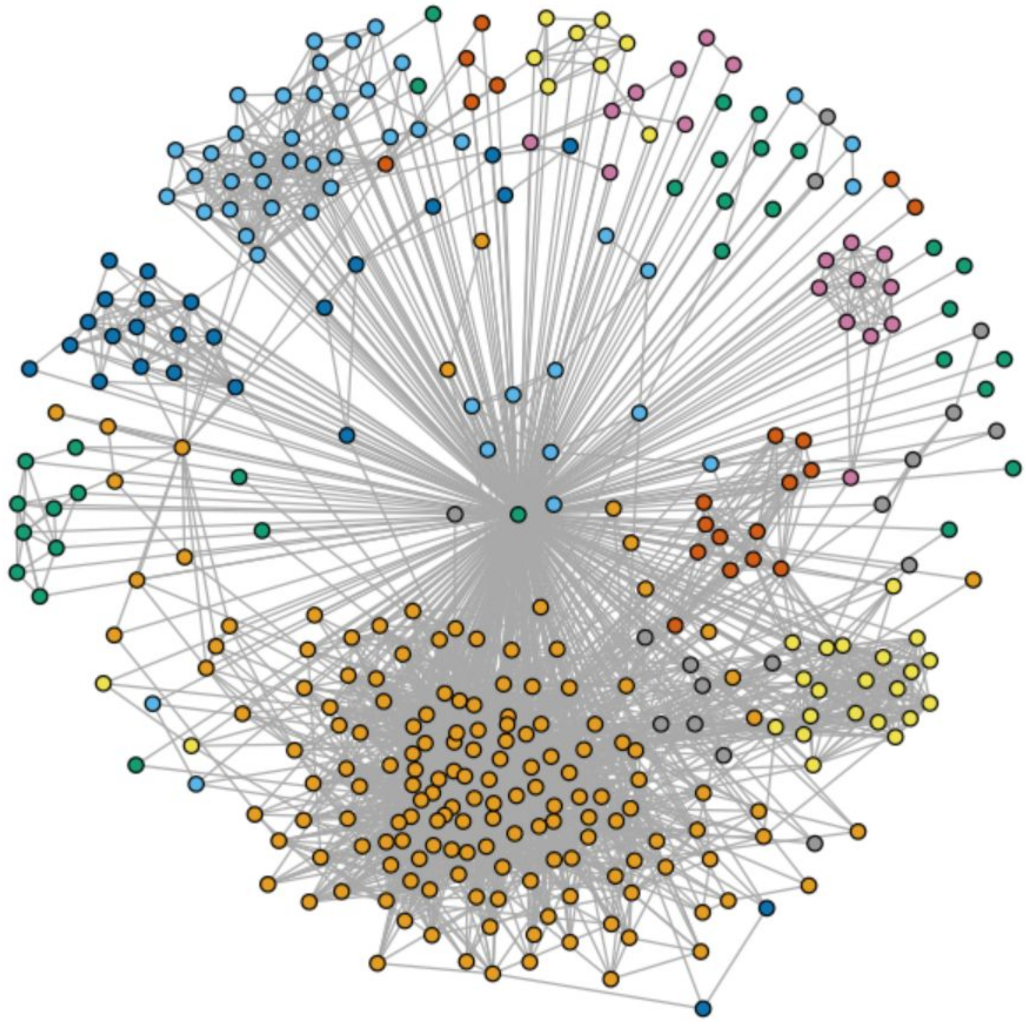

Figure 1.5 Community Structure using Infomap Algorithm for Node 1

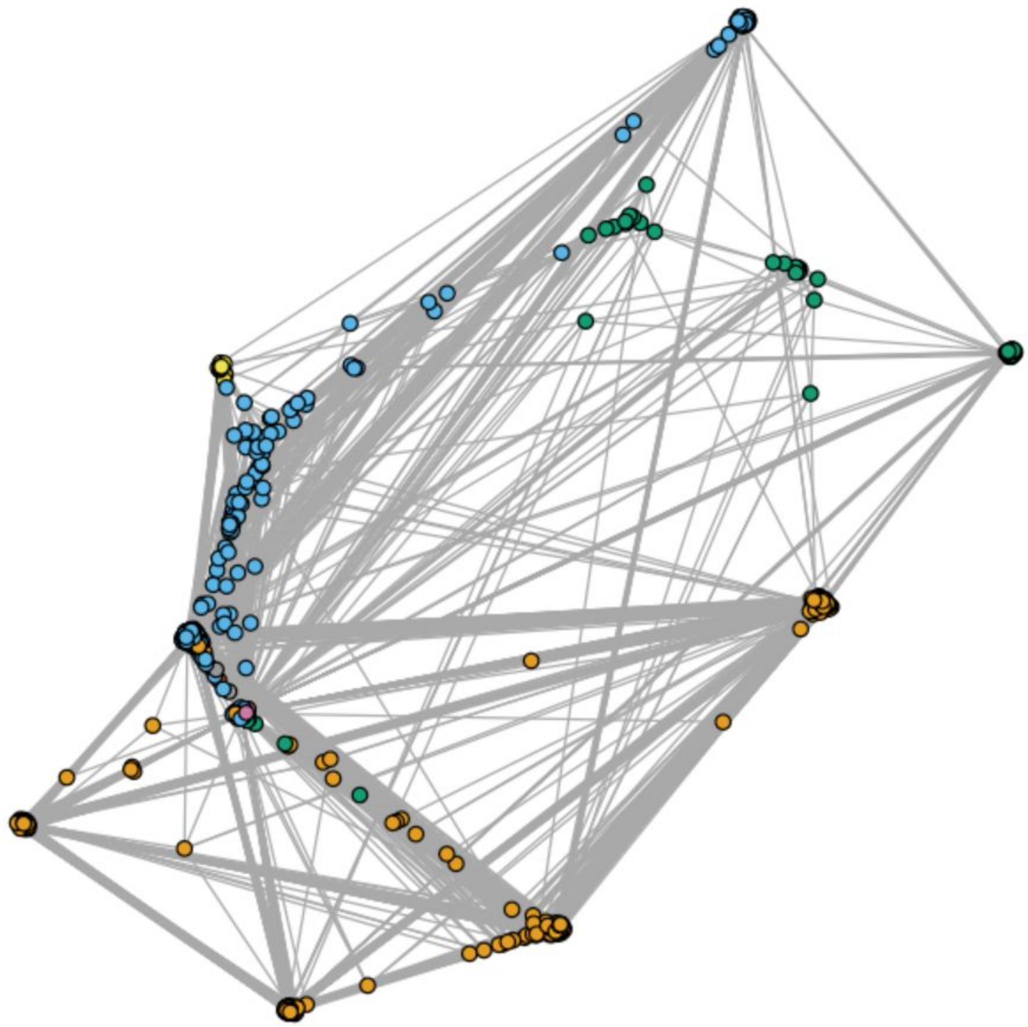

Figure 1.6 Community Structure using Fast-Greedy Algorithm for Node 108

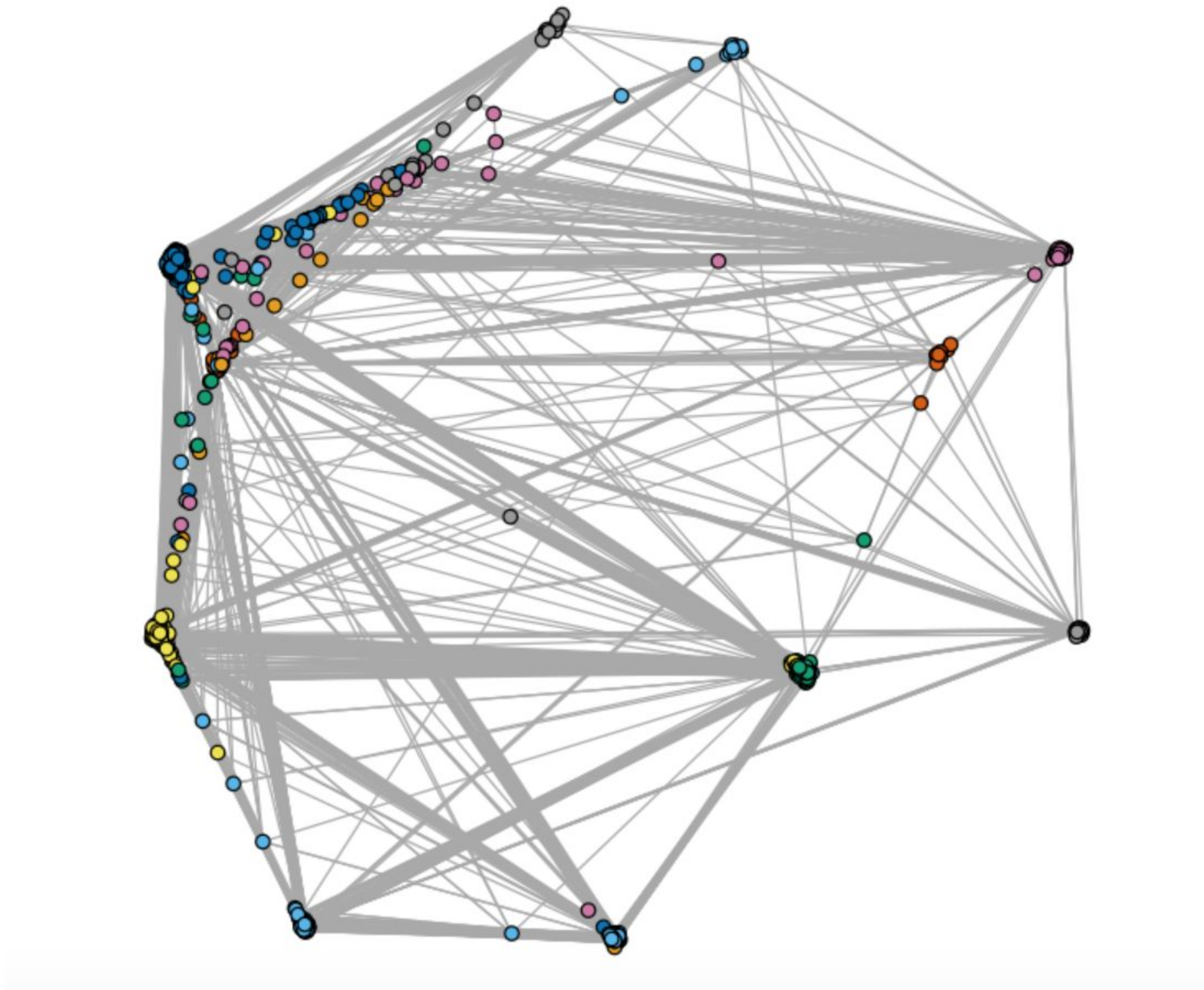

Figure 1.7 Community Structure using Edge-Between Algorithm for Node 108

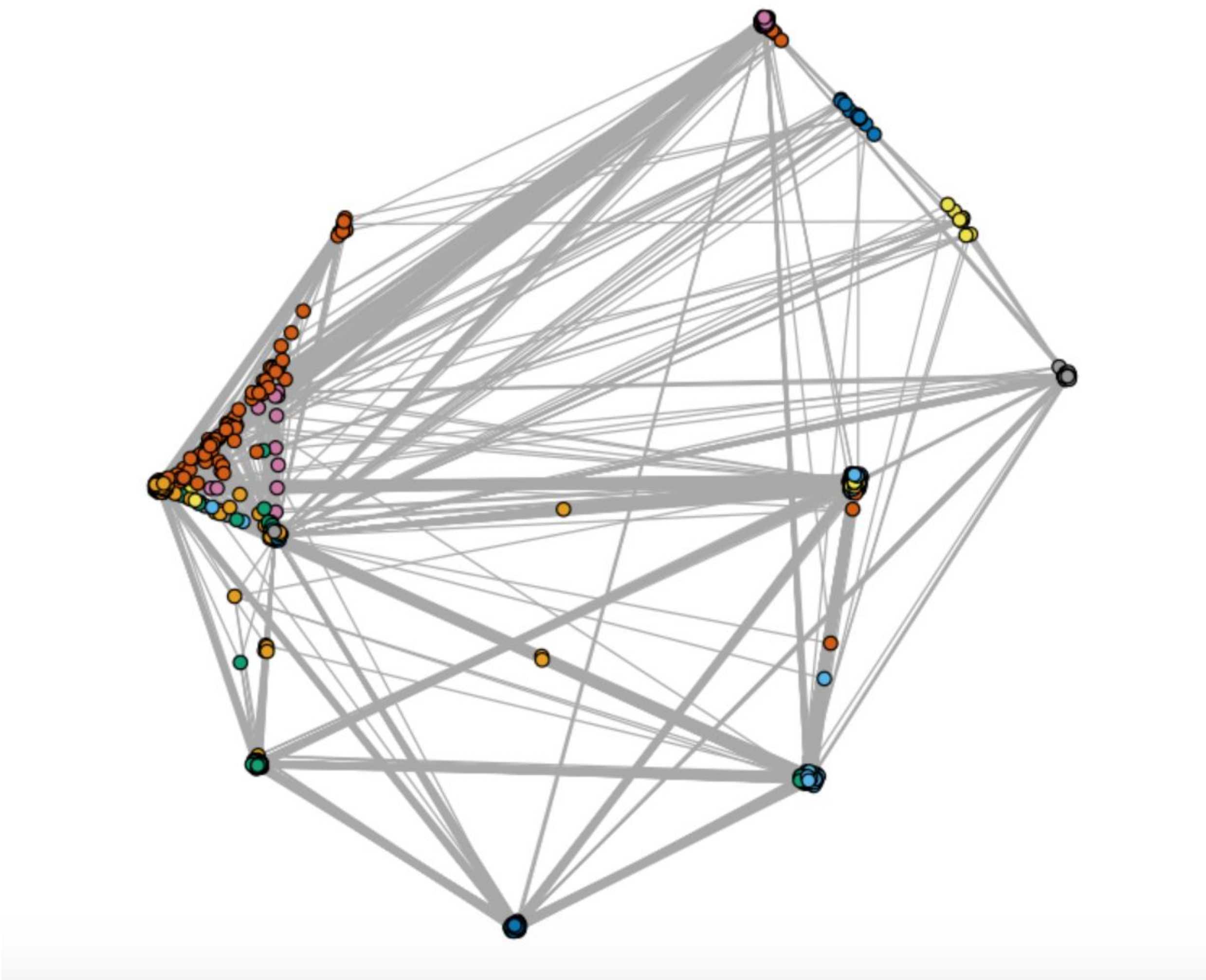

Figure 1.8 Community Structure using Infomap Algorithm for Node 108

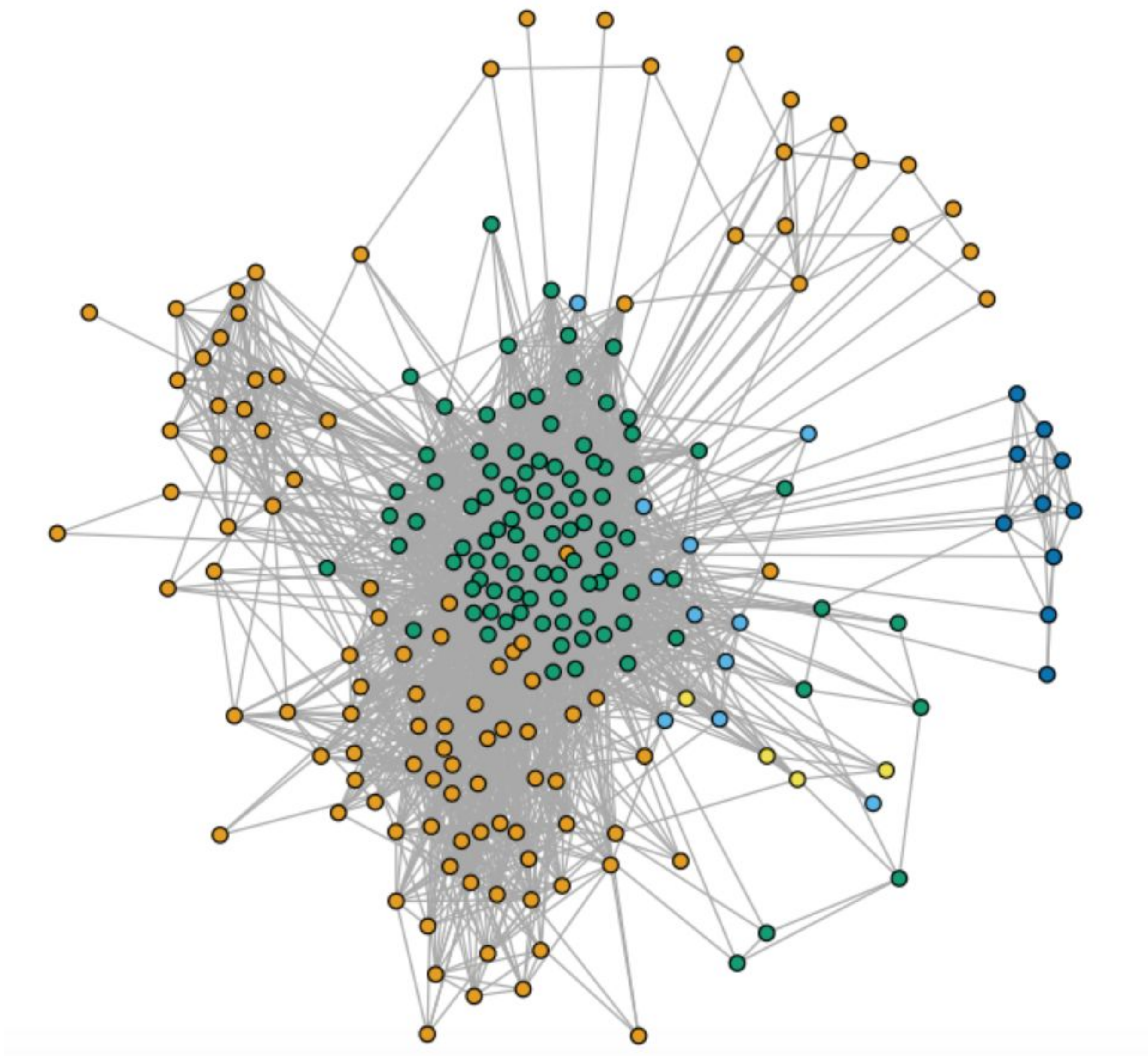

Figure 1.9 Community Structure using Fast-Greedy Algorithm for Node 349

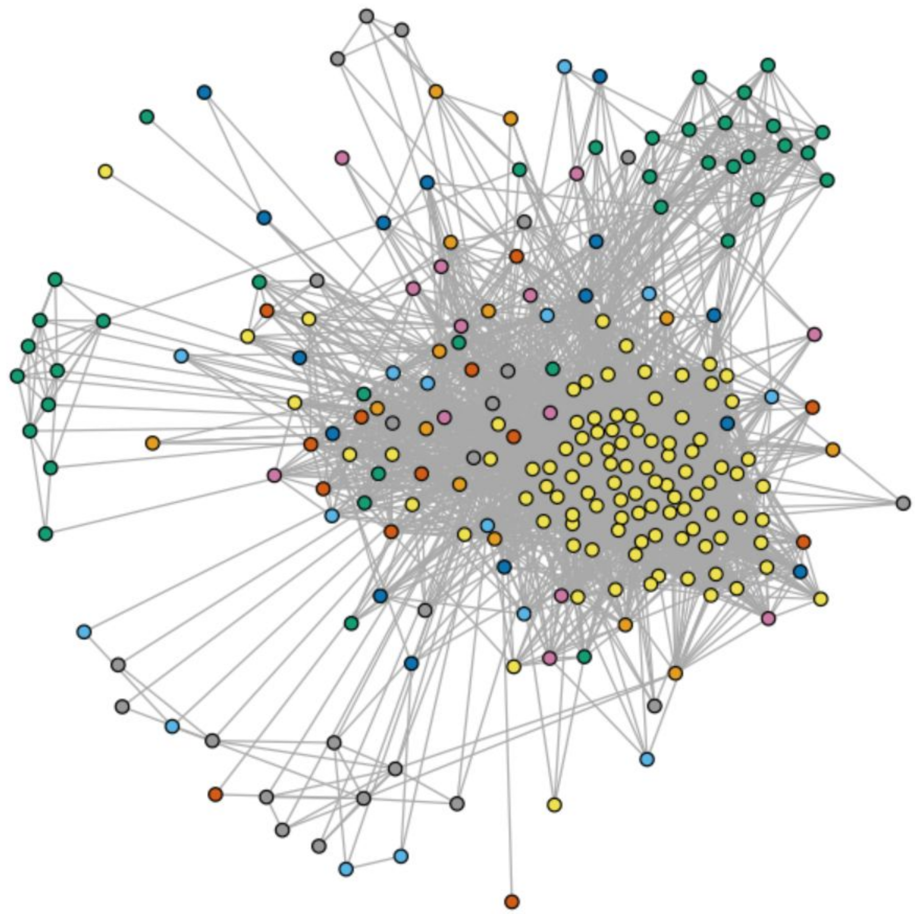

Figure 1.10 Community Structure using Edge-Between Algorithm for Node 349

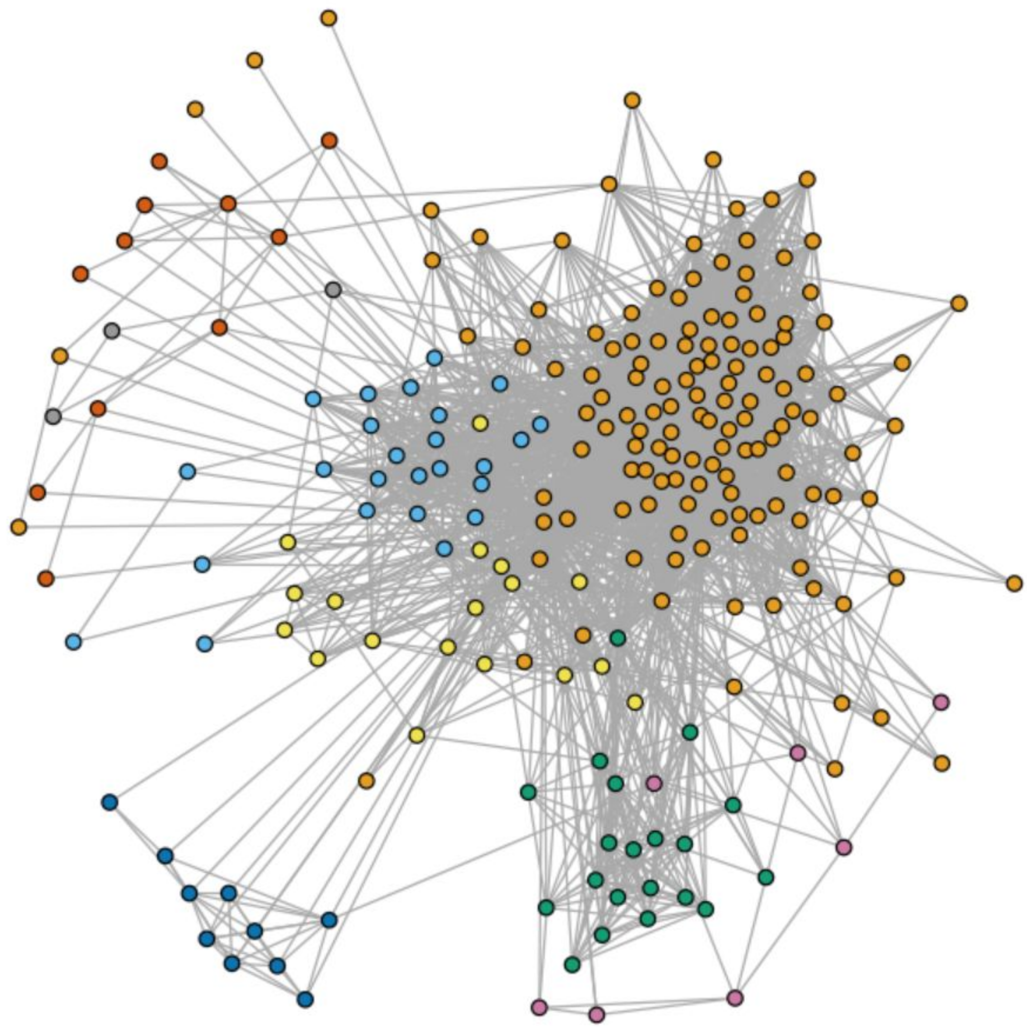

Figure 1.11 Community Structure using Infomap Algorithm for Node 349

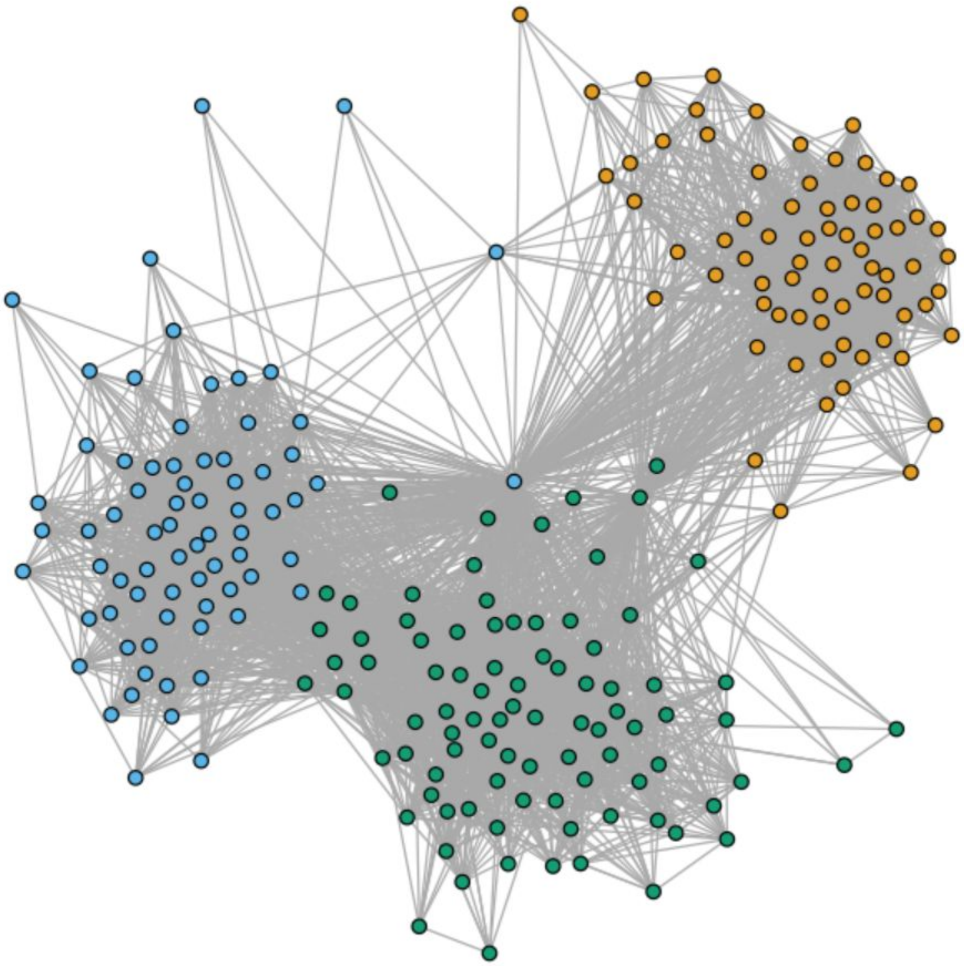

Figure 1.12 Community Structure using Fast-Greedy Algorithm for Node 484

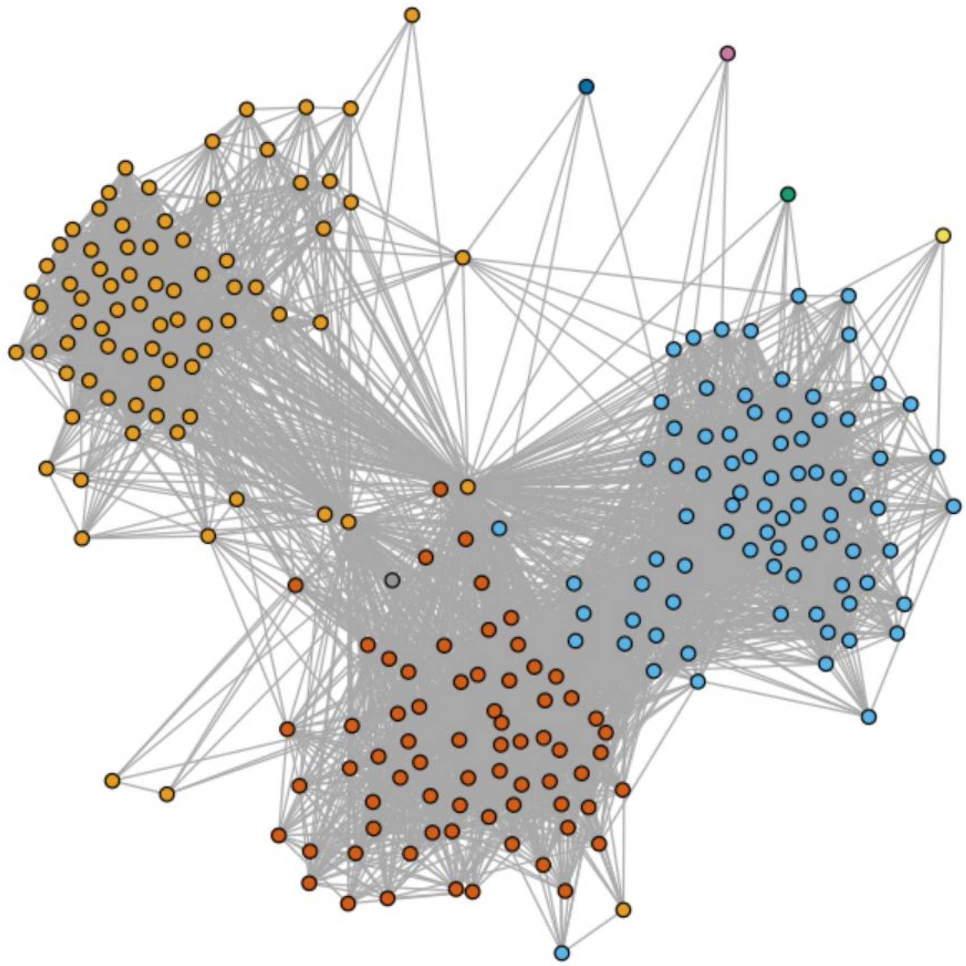

Figure 1.13 Community Structure using Edge-Between Algorithm for Node 484

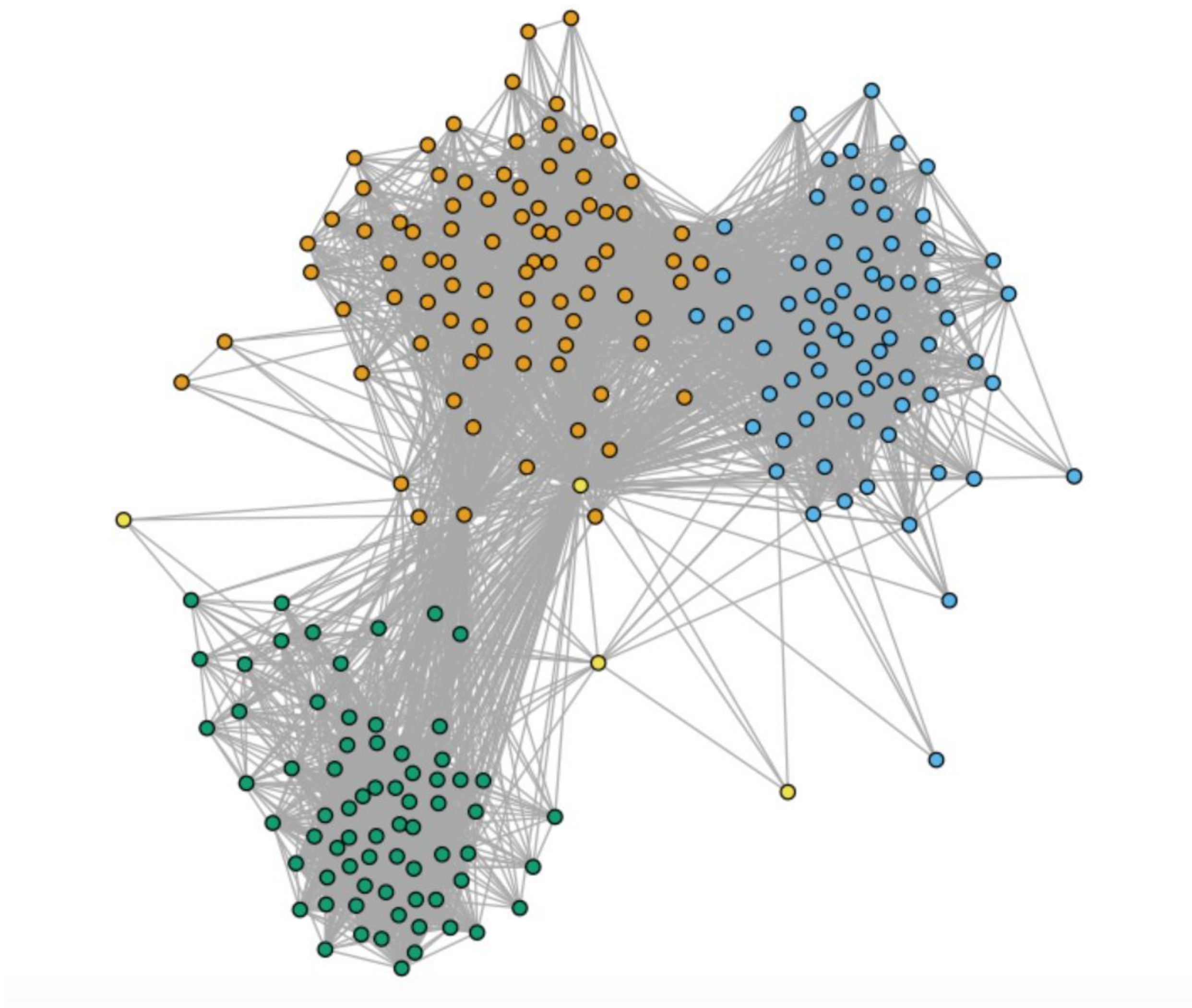

Figure 1.14 Community Structure using Infomap Algorithm for Node 484

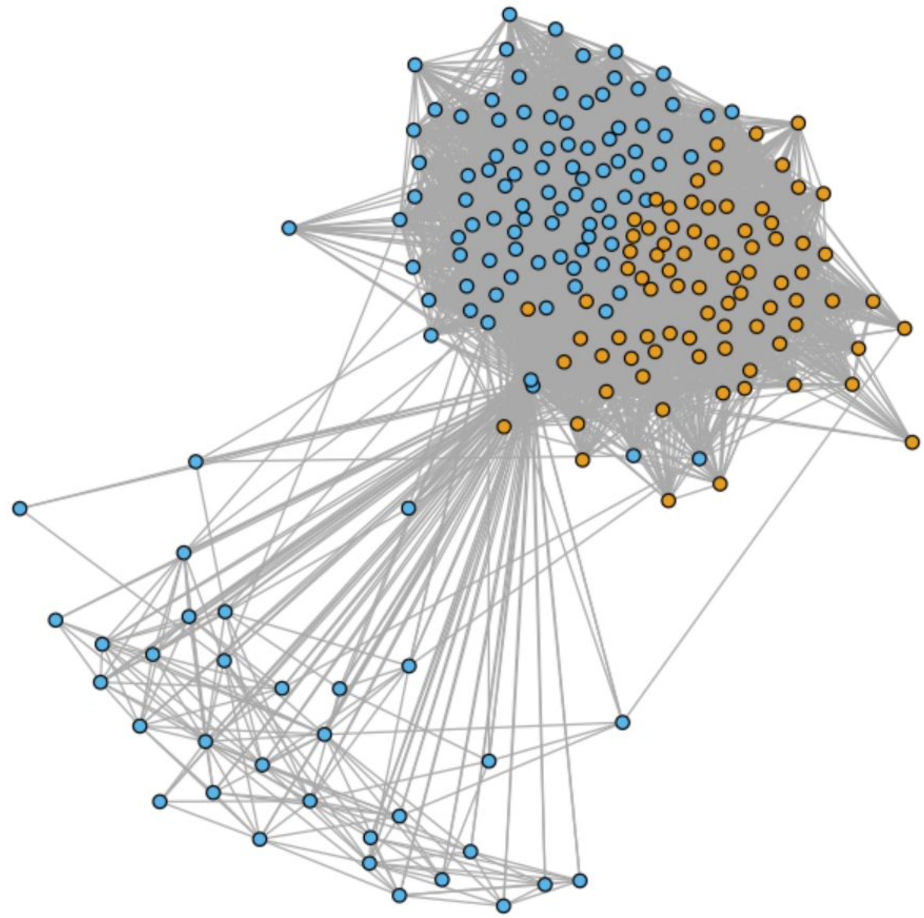

Figure 1.15 Community Structure using Fast-Greedy Algorithm for Node 1087

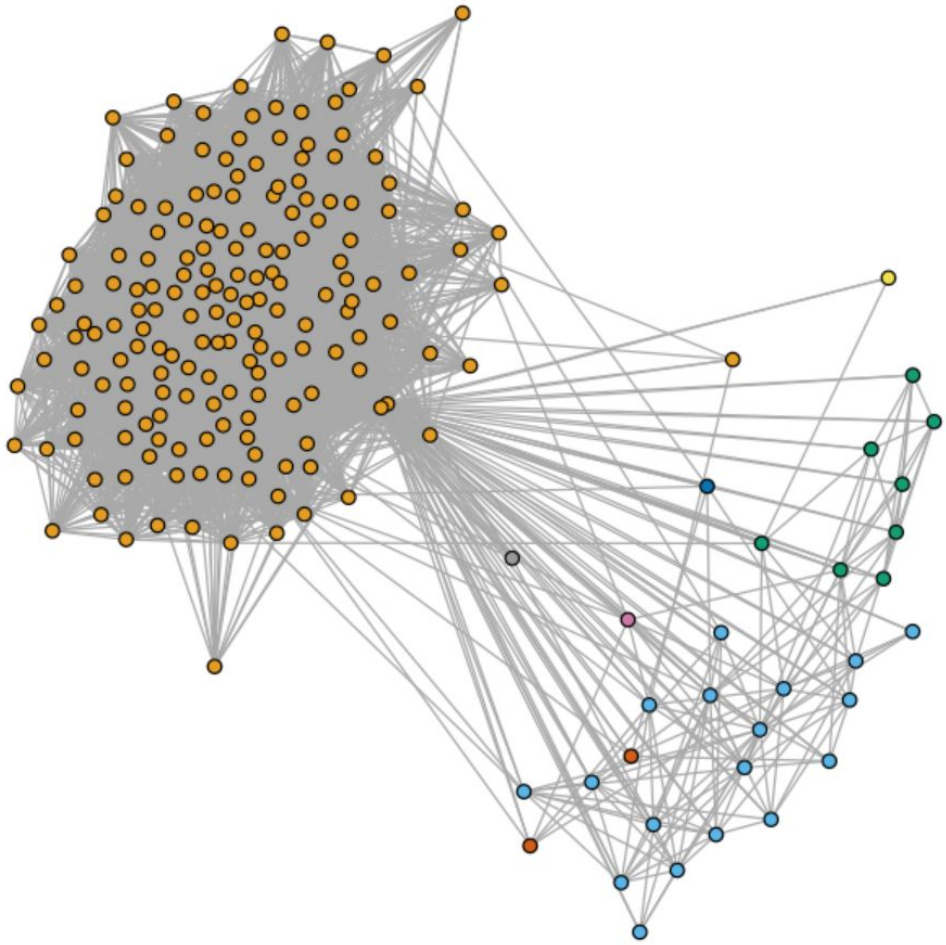

Figure 1.16 Community Structure using Edge-Between Algorithm for Node 1087

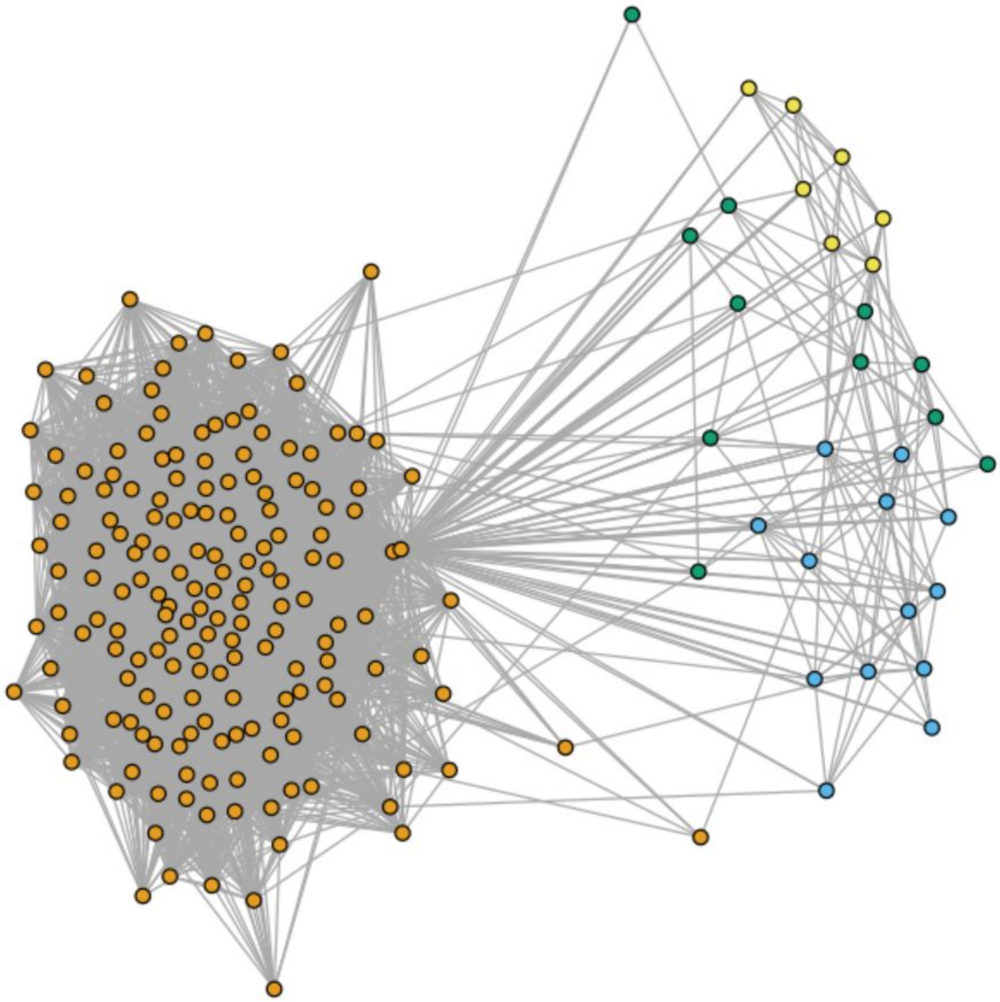

Figure 1.17 Community Structure using Infomap Algorithm for Node 1087

Question 10.

For this part, we first removed the core node from the personalized network and then we used the Fast-Greedy algorithm, Edge-Betweenness algorithm, and Infomap community detection algorithm to find the community structure of the modified personalized network. We tested these algorithms on different core nodes' (Node ID 1, Node ID 108, Node ID 349, Node ID 484, and Node ID 1087) personalized network and the modularity scores can be shown in table 1.3.

Core Node ID	Modularity Score for Fast-Greedy Algorithm	Modularity Score for Edge-Betweenness Algorithm	Modularity Score for Infomap Community Detection Algorithm

1	0.4418533	0.4161461	0.4180077
108	0.4581271	0.5213216	0.517954
349	0.2456918	0.1505663	0.2448156
484	0.5342142	0.5154413	0.5434437
1087	0.1481956	0.0324953	0.02737159

Table 1.3 The Modularity Score of Different Algorithms with Core Nodes Removed

Besides, we also plotted the community structure of community structure of the modified personalized network using colors.

The community structures (using different algorithm) of the modified personalized network using node id 1 as core node can can be shown as Fig. 1.18, Fig. 1.19, and Fig. 1.20.

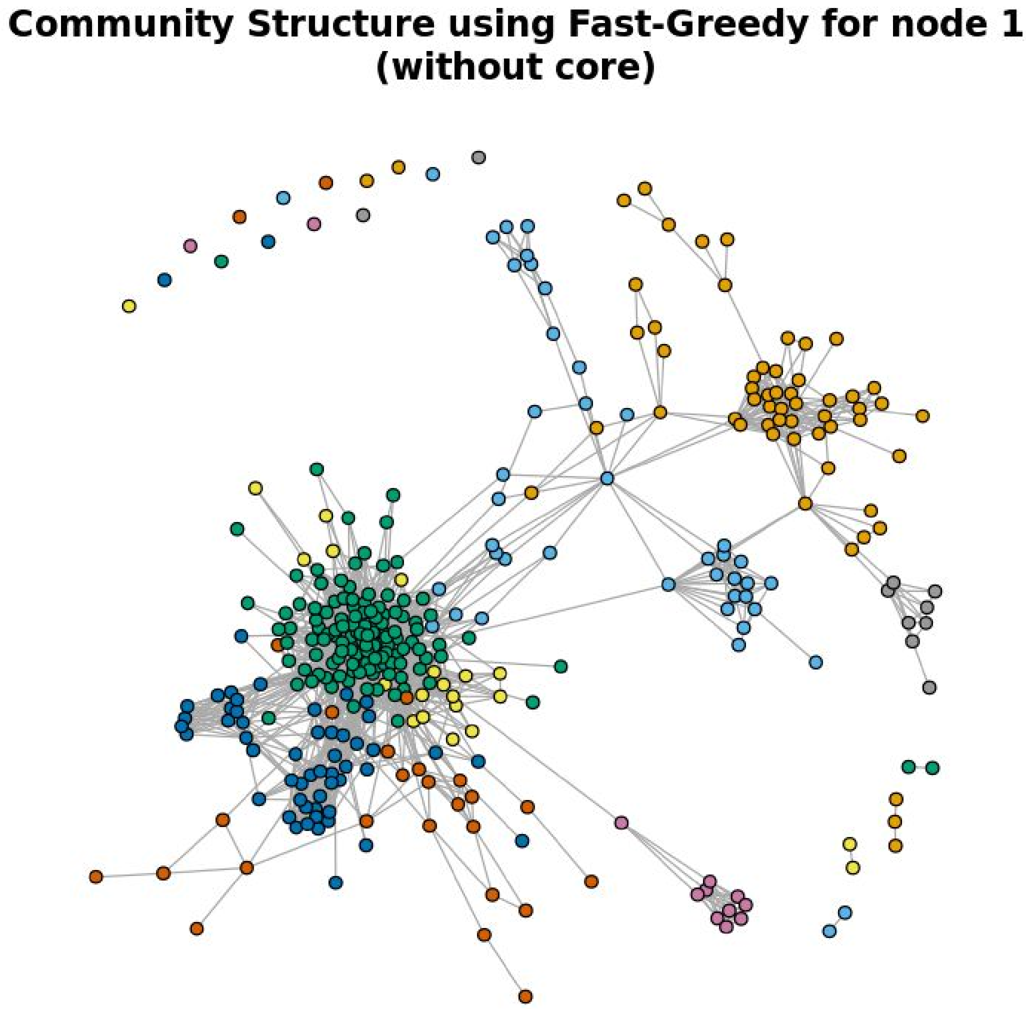

Figure 1.18 Community Structure using Fast-Greedy Algorithm for the Modified Personalized Network Using Node 1 as Core Node

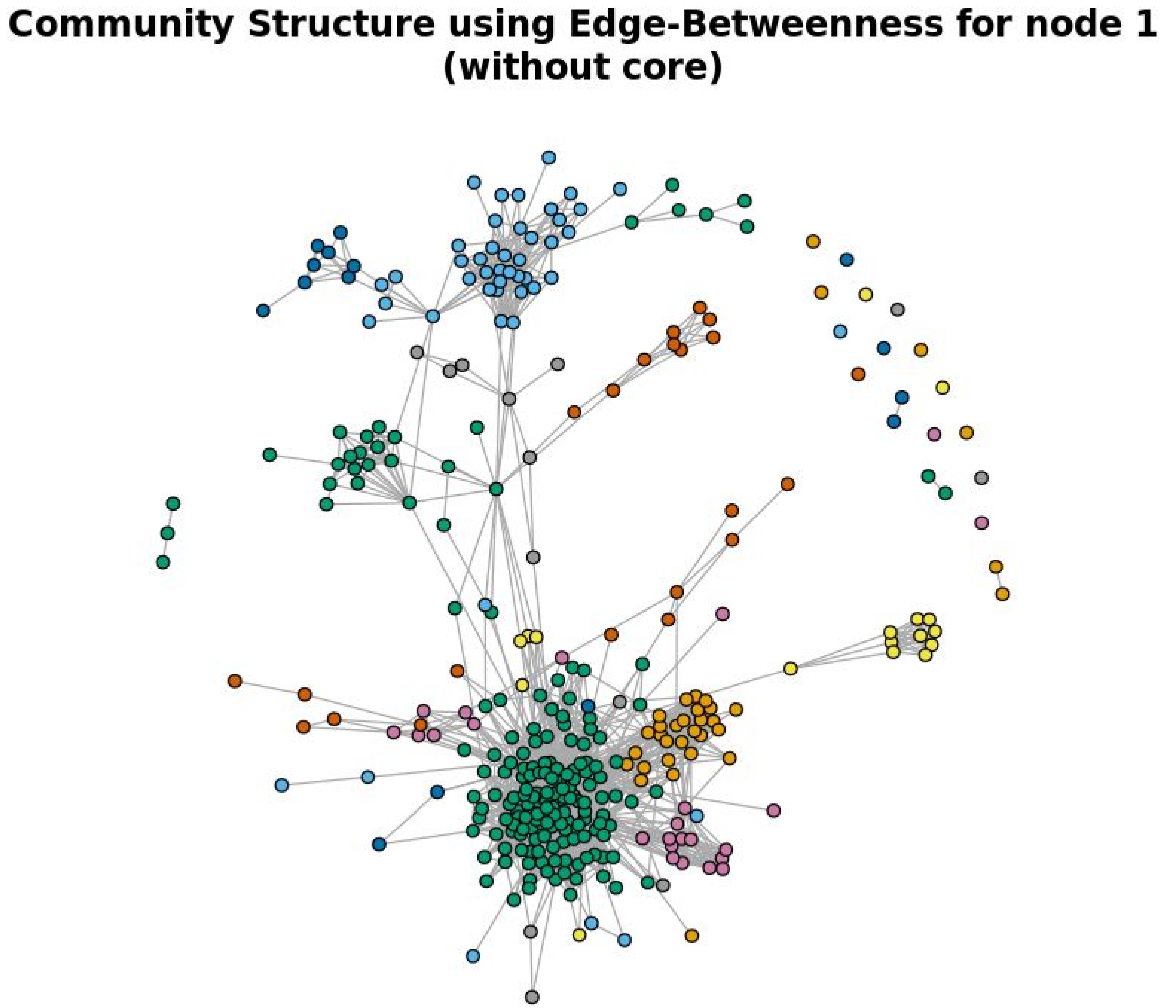

Figure 1.19 Community Structure using Edge-Betweenness Algorithm for the Modified Personalized Network Using Node 1 as Core Node

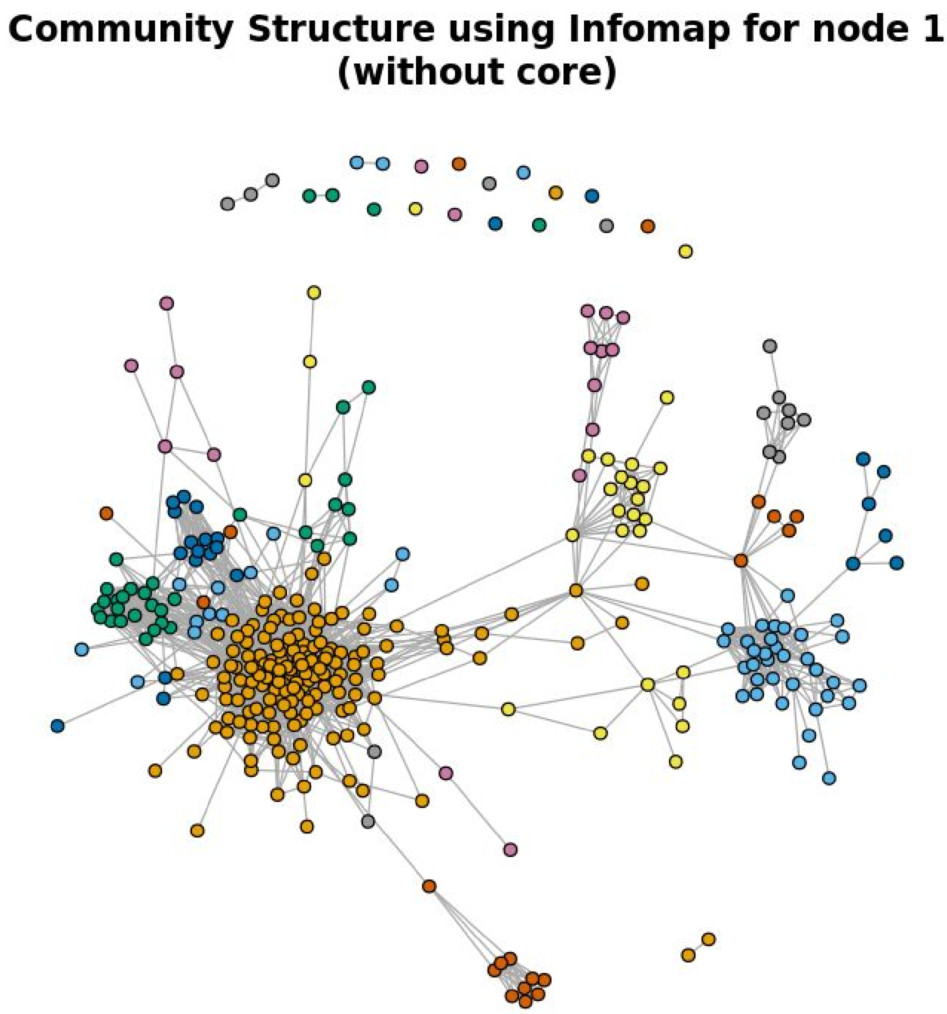

Figure 1.20 Community Structure using Infomap Community Detection Algorithm for the Modified Personalized Network Using Node 1 as Core Node

The community structures (using different algorithm) of the modified personalized network using node id 108 as core node can can be shown as Fig. 1.21, Fig. 1.22, and Fig. 1.22.

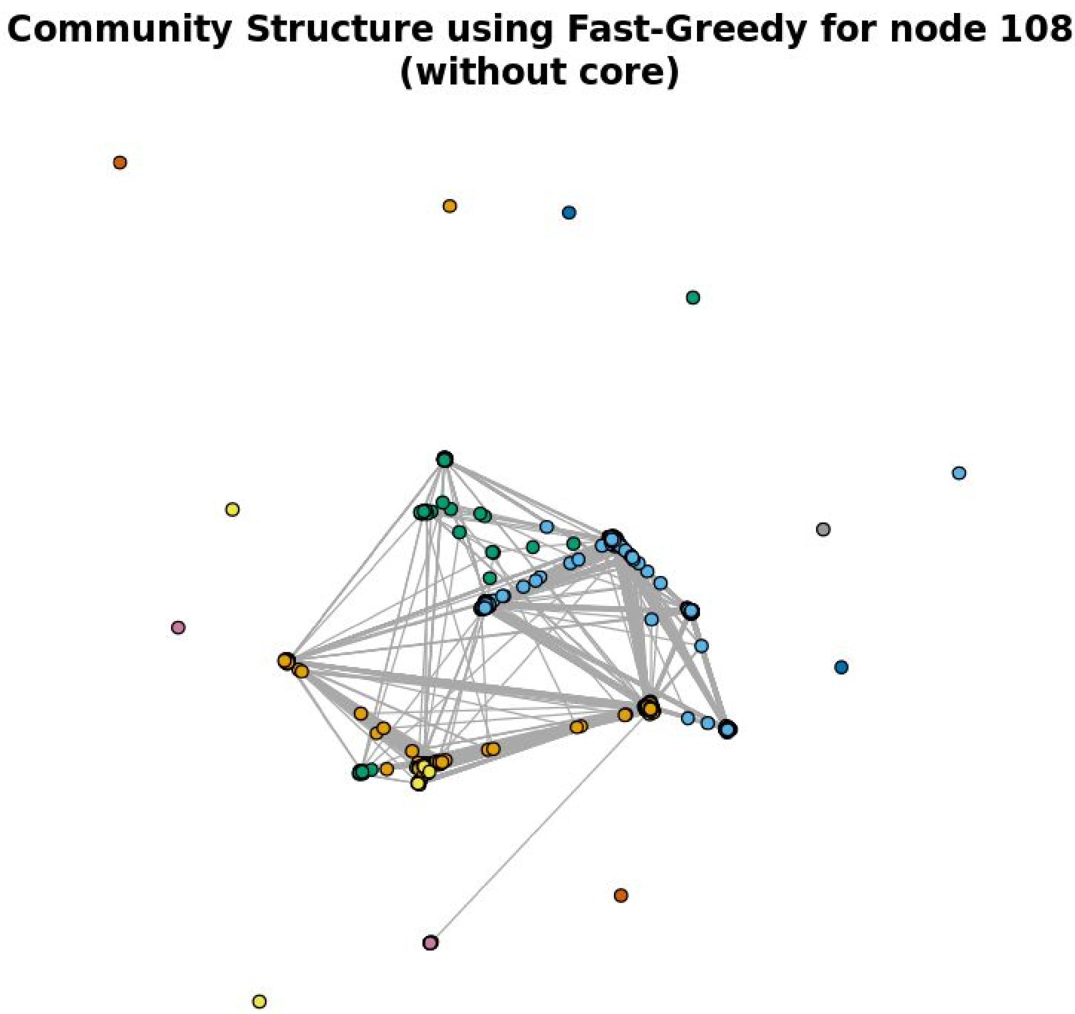

Figure 1.21 Community Structure using Fast-Greedy Algorithm for the Modified Personalized Network Using Node 108 as Core Node

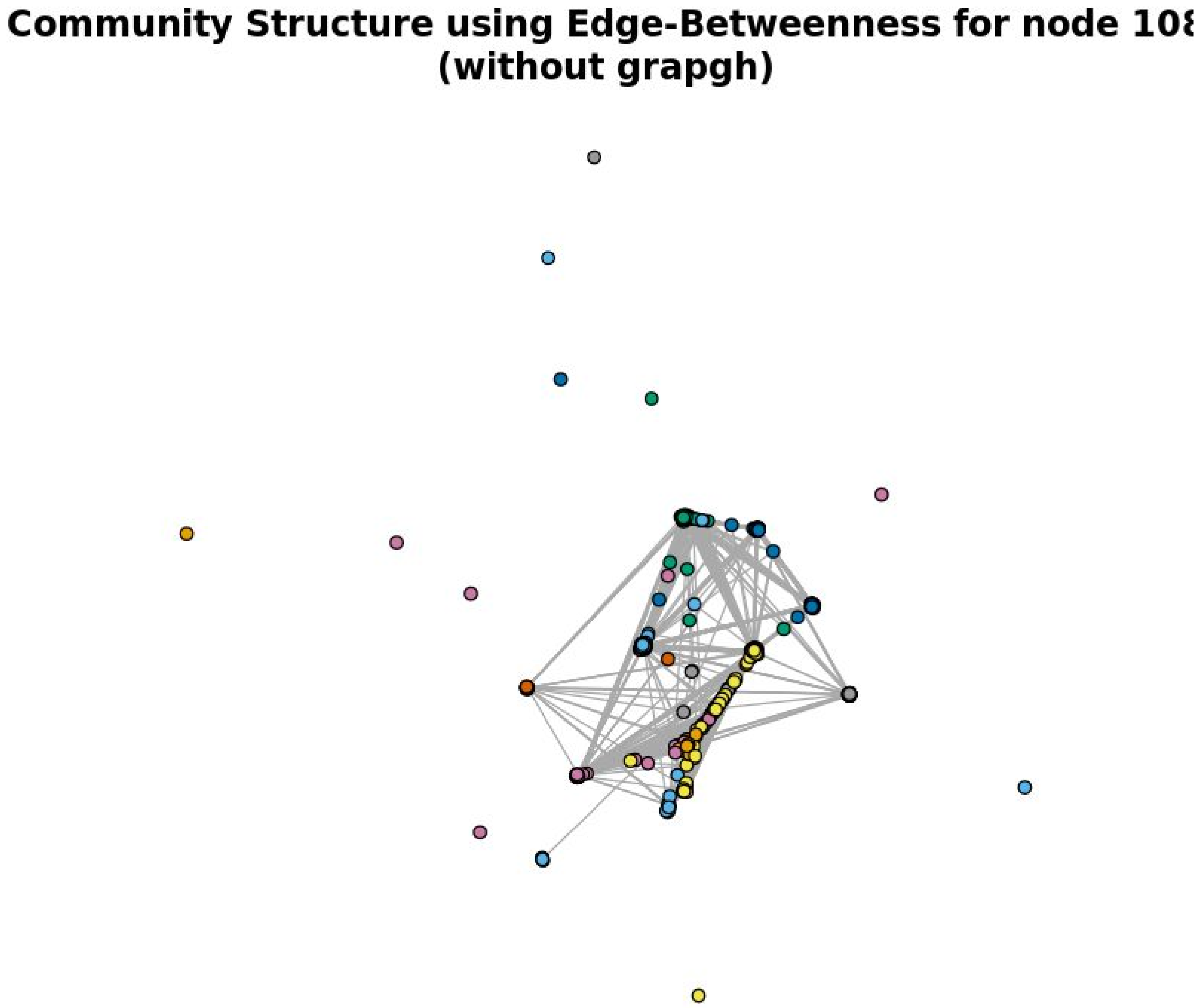

Figure 1.22 Community Structure using Edge-Betweenness Algorithm for the Modified Personalized Network Using Node 108 as Core Node

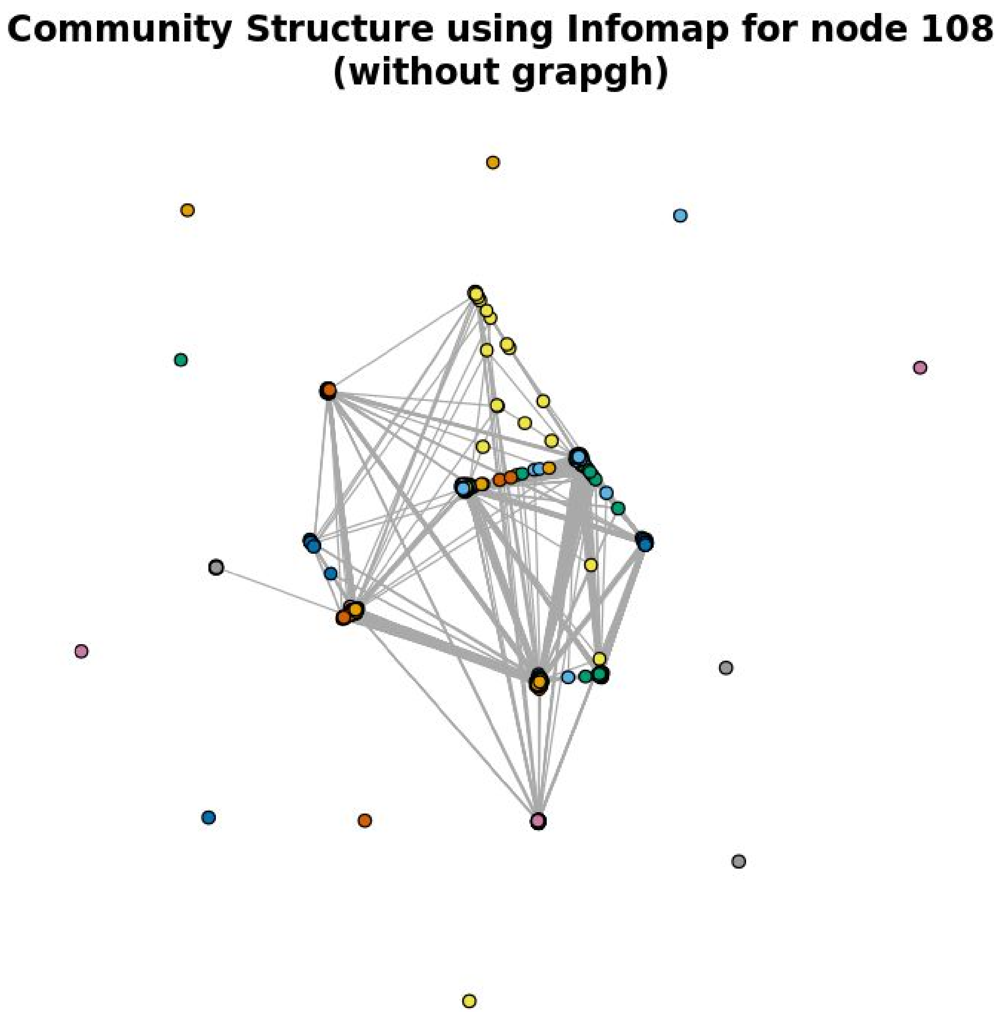

Figure 1.23 Community Structure using Infomap Community Detection Algorithm for the Modified Personalized Network Using Node 108 as Core Node

The community structures (using different algorithm) of the modified personalized network using node id 349 as core node can can be shown as Fig. 1.24, Fig. 1.25, and Fig. 1.26.

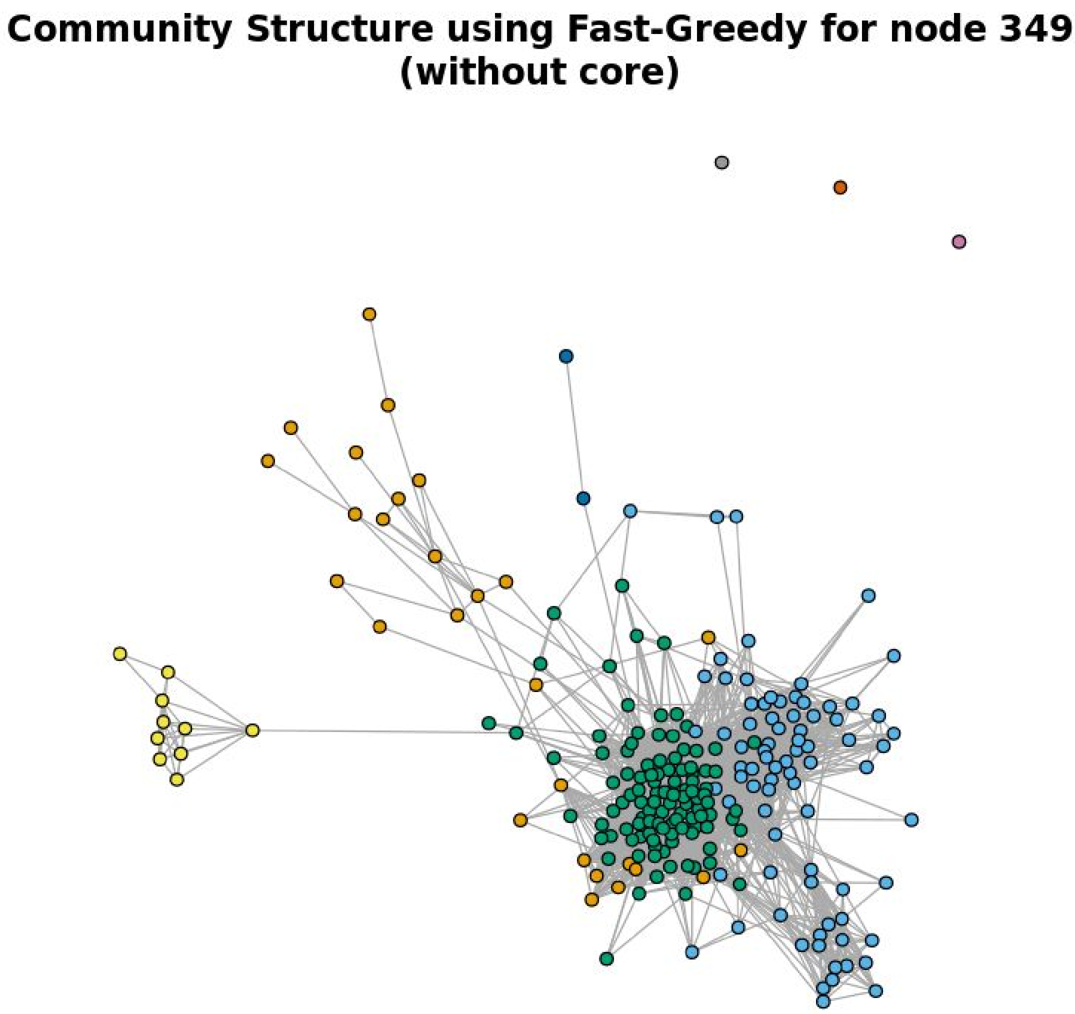

Figure 1.24 Community Structure using Fast-Greedy Algorithm for the Modified Personalized Network Using Node 349 as Core Node

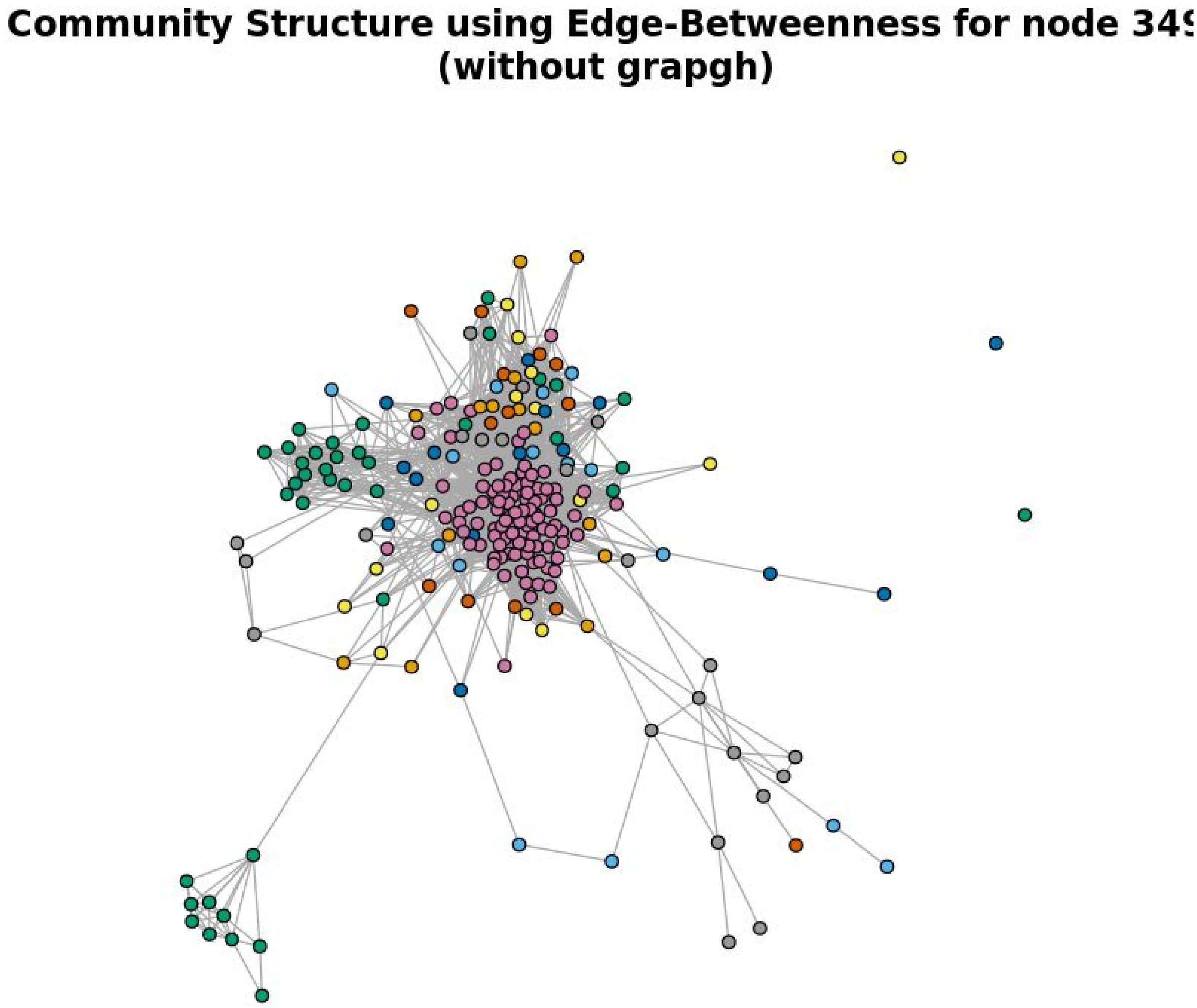

Figure 1.25 Community Structure using Edge-Betweenness Algorithm for the Modified Personalized Network Using Node 349 as Core Node

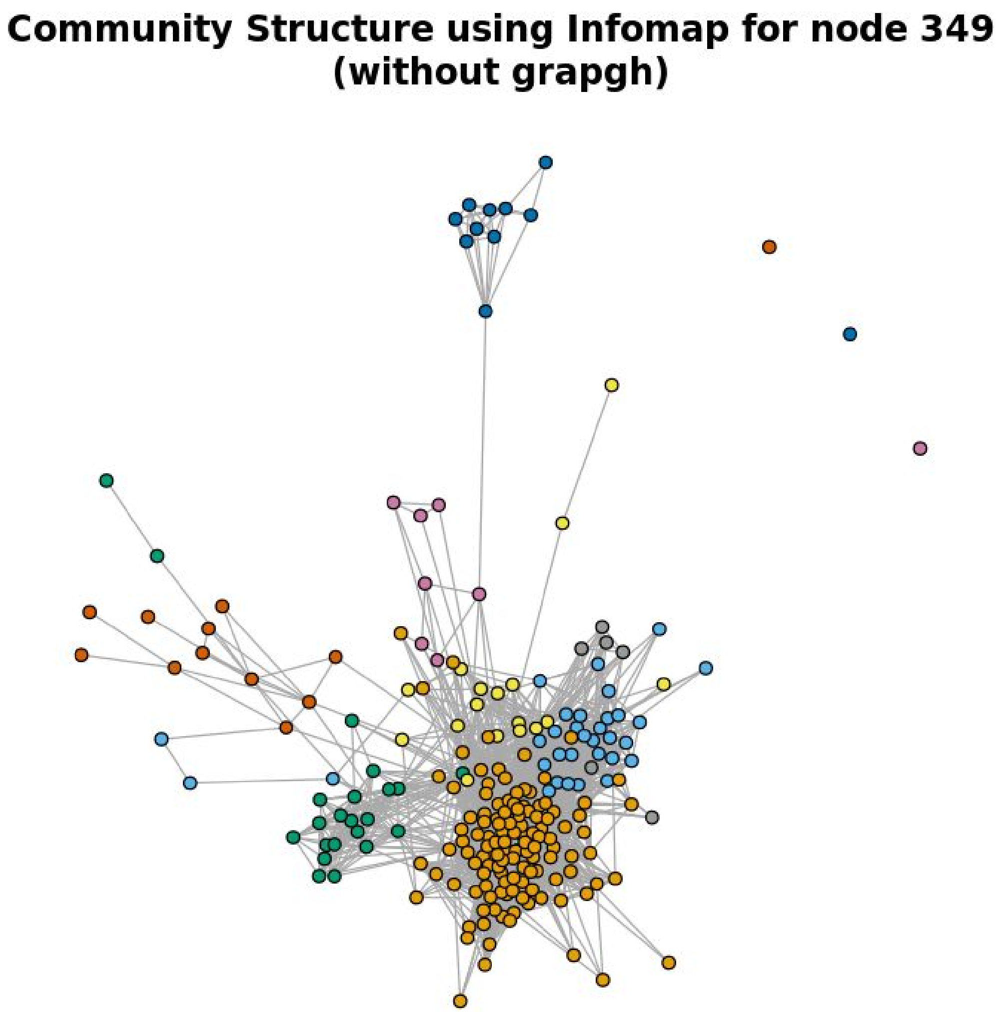

Figure 1.26 Community Structure using Infomap Community Detection Algorithm for the Modified Personalized Network Using Node 349 as Core Node

The community structures (using different algorithm) of the modified personalized network using node id 484 as core node can can be shown as Fig. 1.27, Fig. 1.28, and Fig. 1.29.

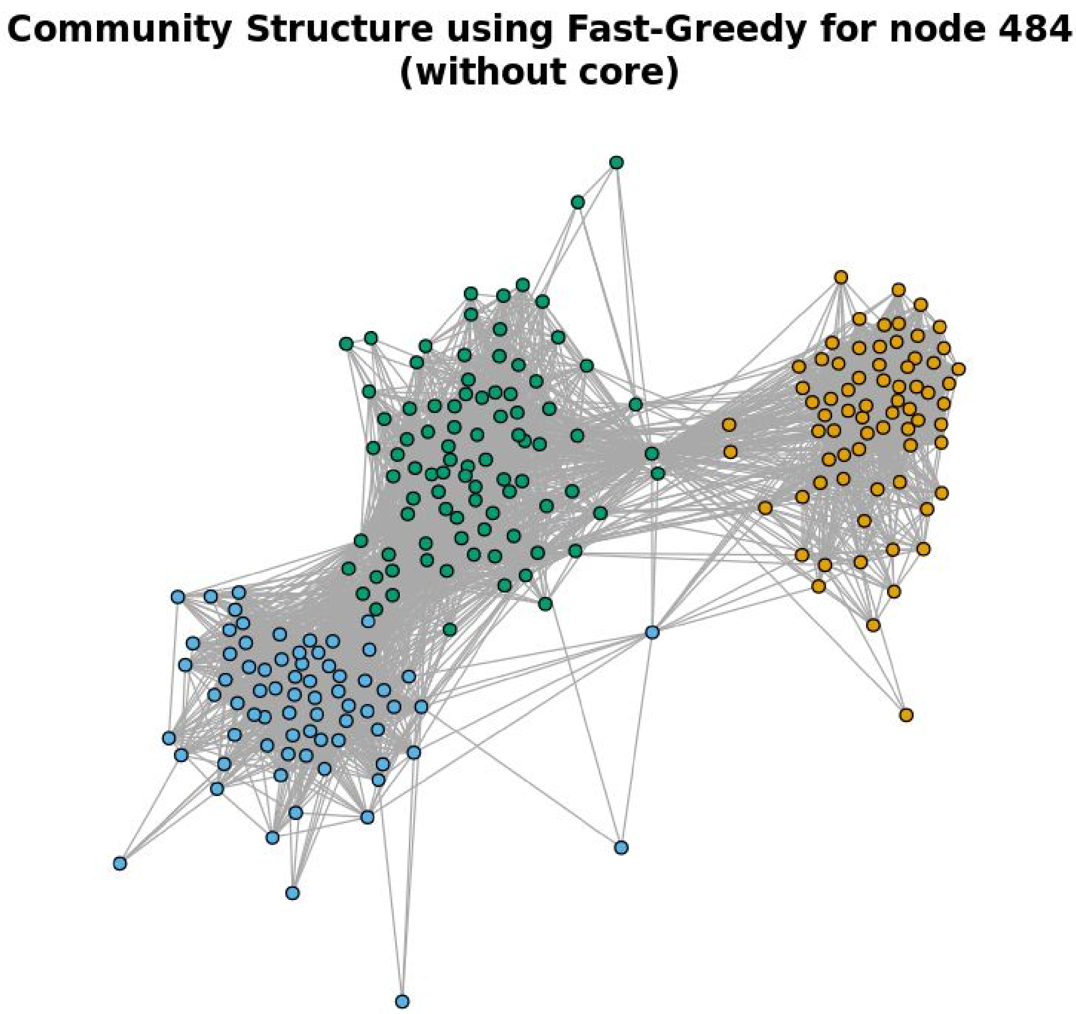

Figure 1.27 Community Structure using Fast-Greedy Algorithm for the Modified Personalized Network Using Node 484 as Core Node

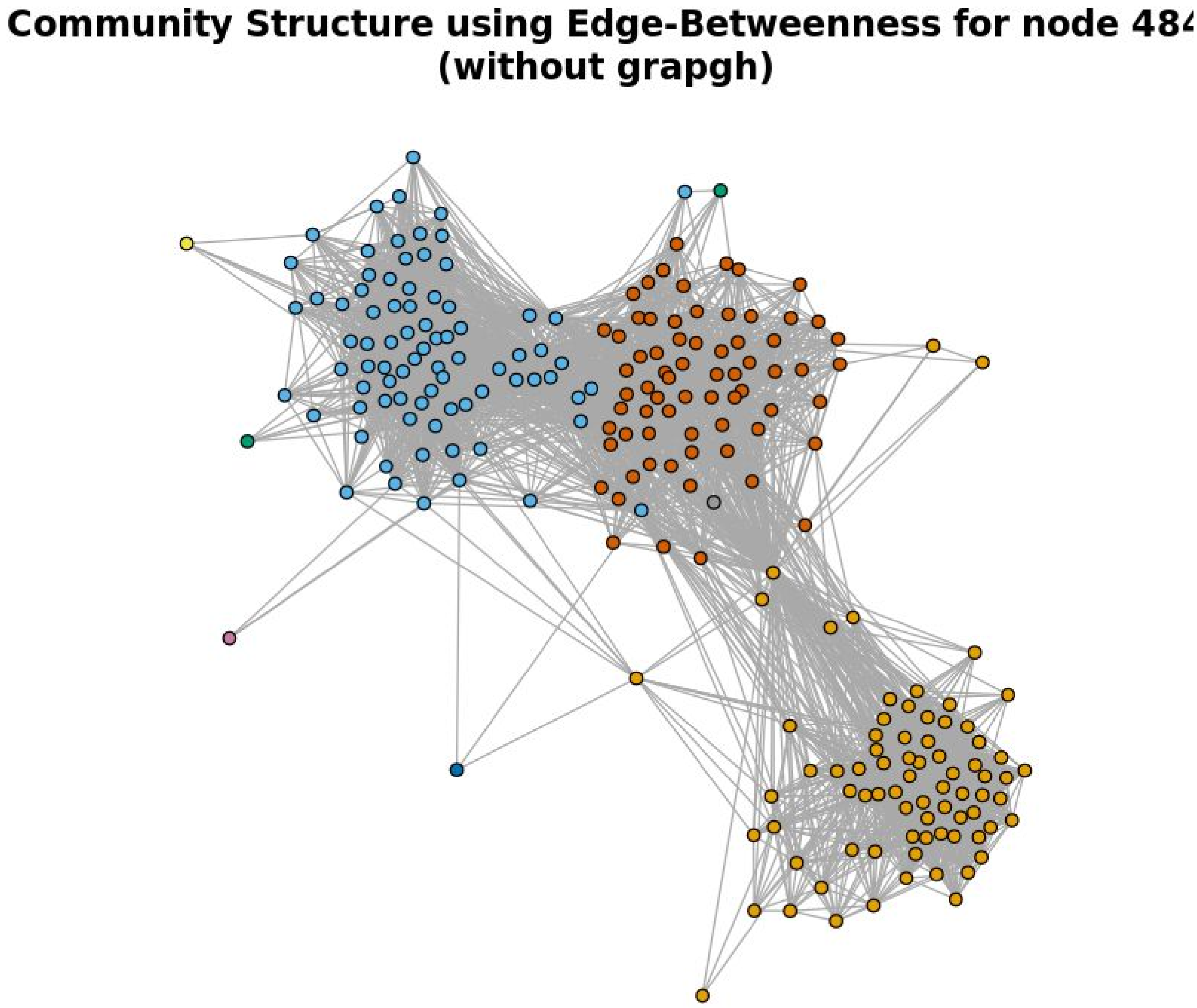

Figure 1.28 Community Structure using Edge-Betweenness Algorithm for the Modified Personalized Network Using Node 484 as Core Node

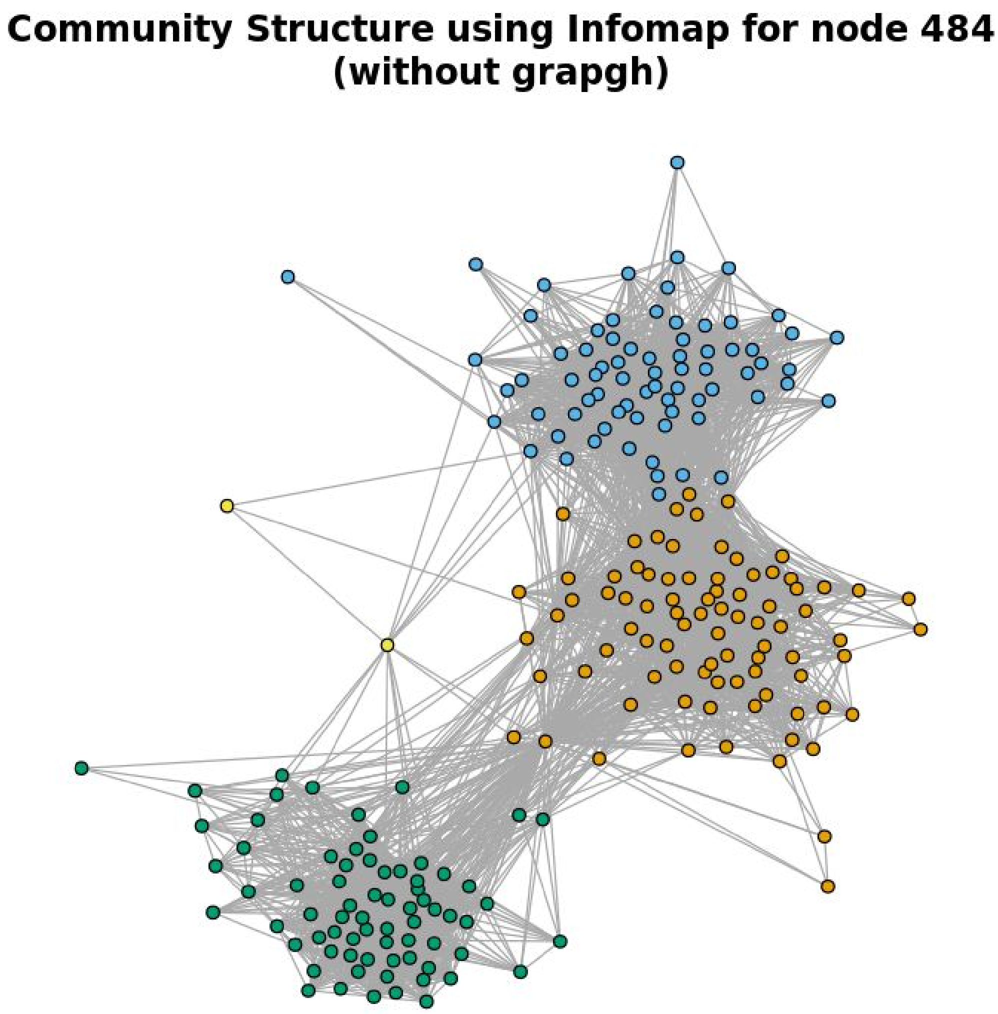

Figure 1.29 Community Structure using Infomap Community Detection Algorithm for the Modified Personalized Network Using Node 484 as Core Node

The community structures (using different algorithm) of the modified personalized network using node id 1087 as core node can can be shown as Fig. 1.30, Fig. 1.31, and Fig. 1.32.

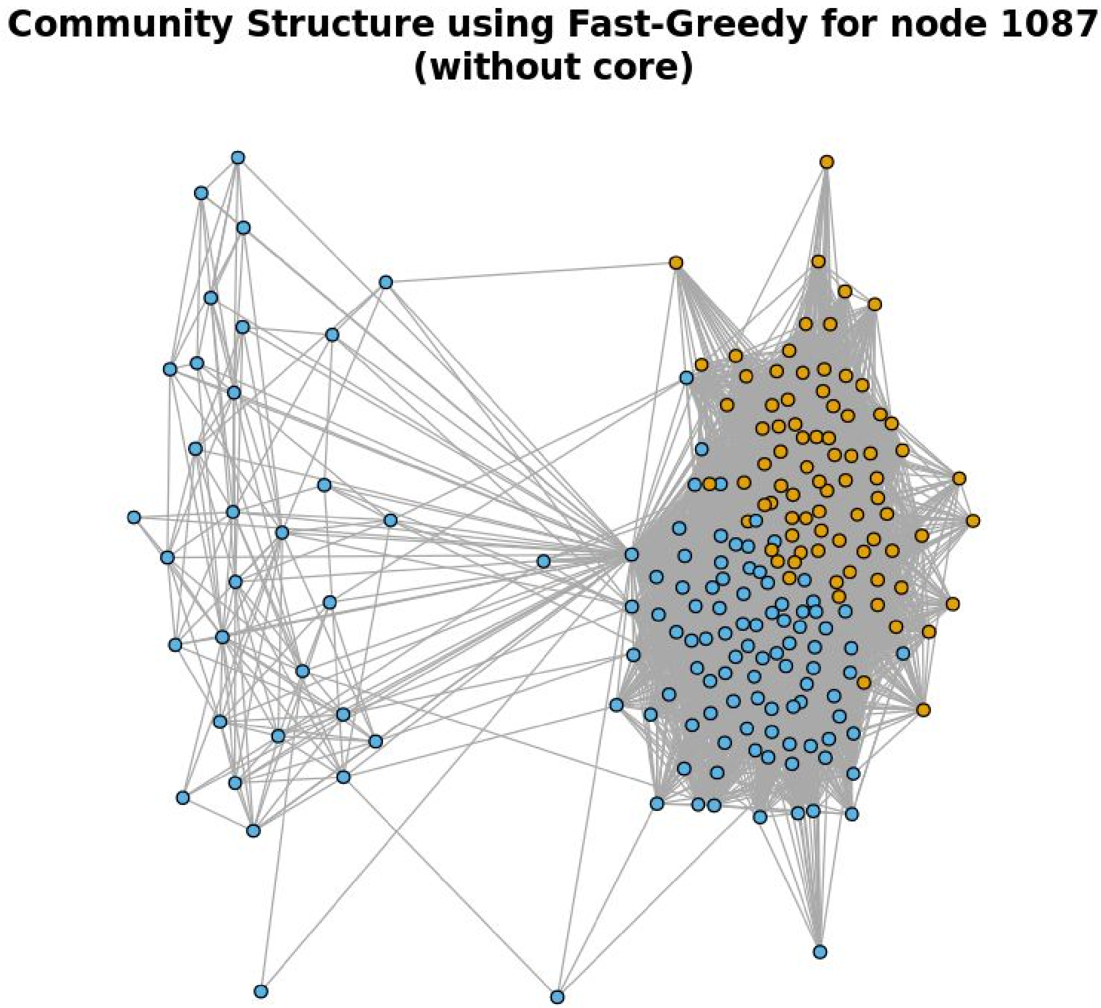

Figure 1.30 Community Structure using Fast-Greedy Algorithm for the Modified Personalized Network Using Node 1087 as Core Node

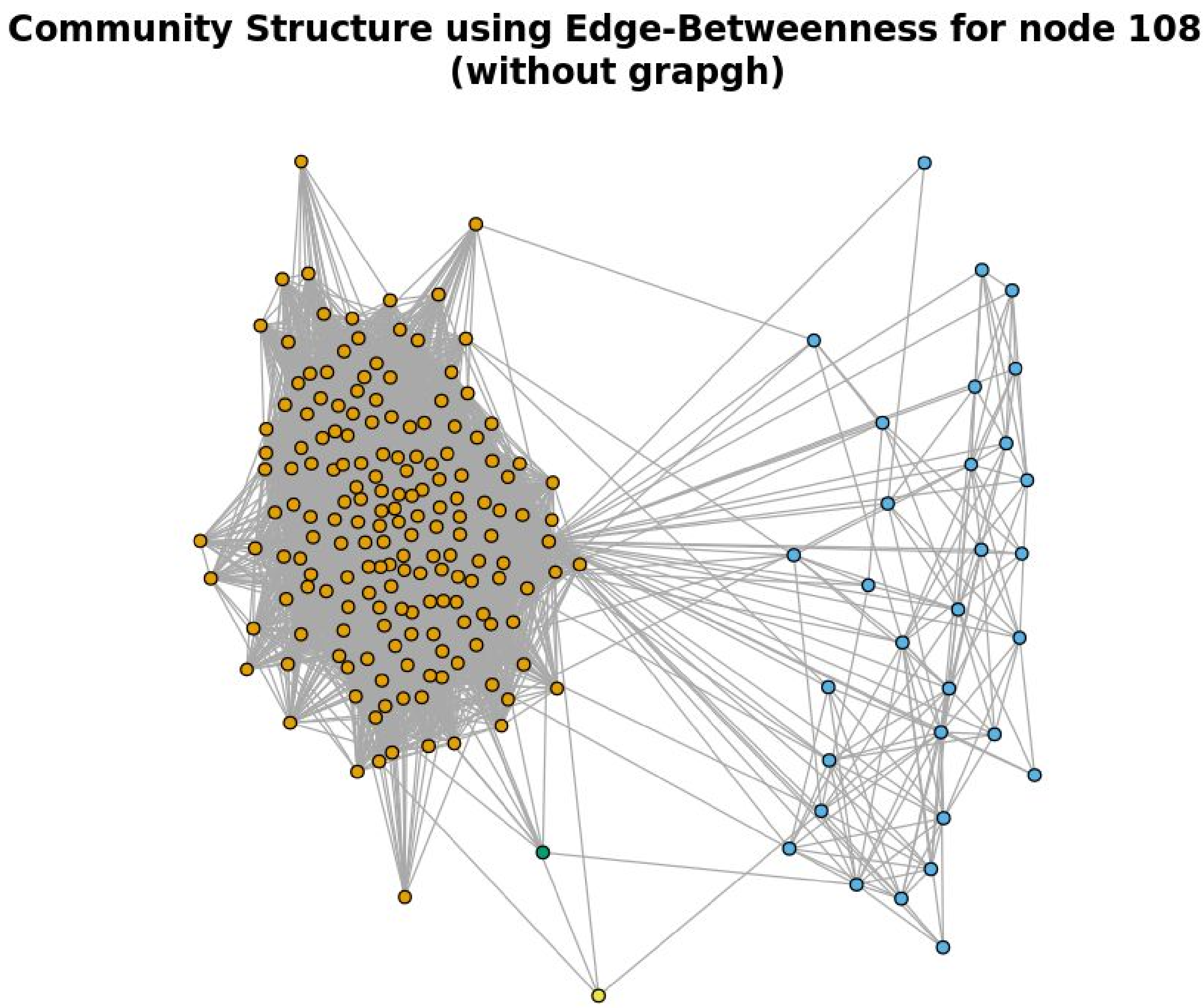

Figure 1.31 Community Structure using Edge-Betweenness Algorithm for the Modified Personalized Network Using Node 1087 as Core Node

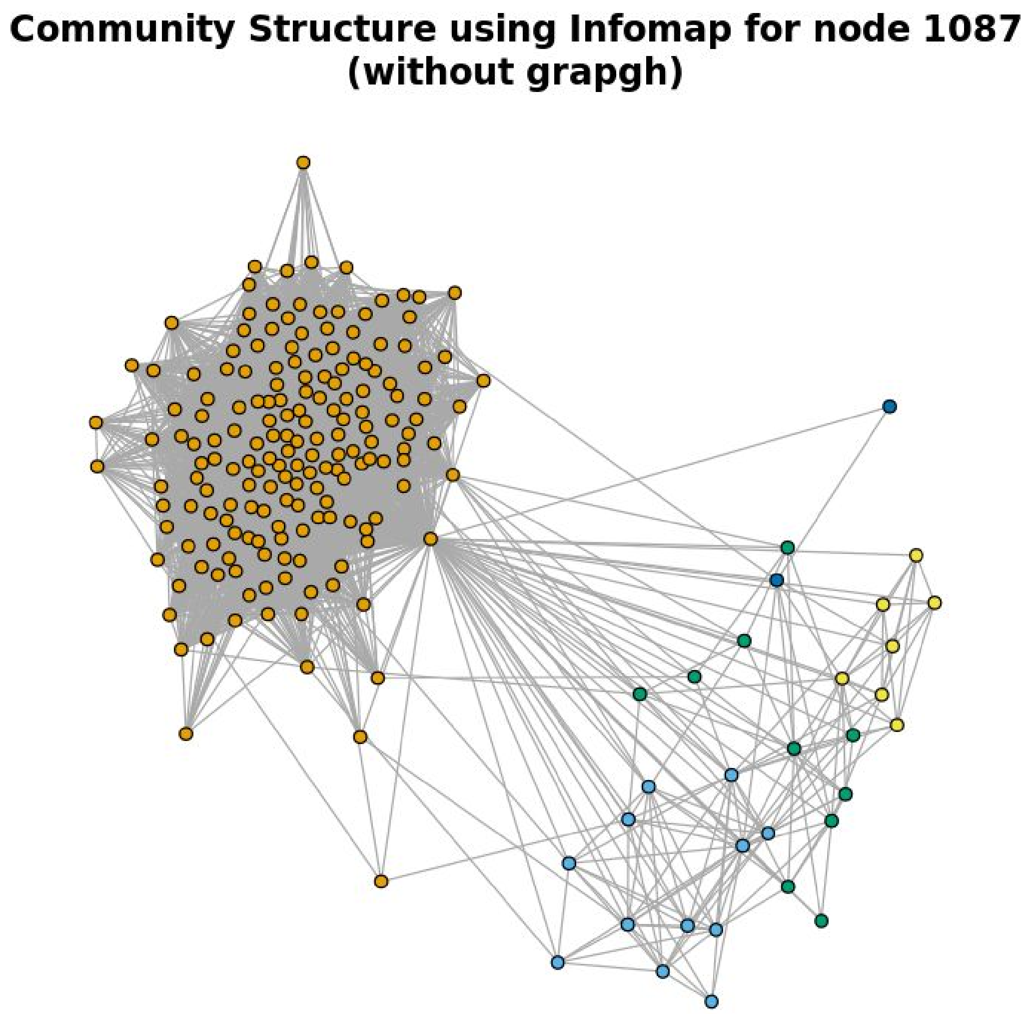

Figure 1.32 Community Structure using Infomap Community Detection Algorithm for the Modified Personalized Network Using Node 1087 as Core Node

Question 11.

Because for this part, we were focusing on the subgraph which contains one core node and all other node are the neighbours of that core node. Thus, in this case, the mutual friends a node shares with the core node is actually all its neighbours except the core nodes. Thus, the relation between the embeddedness and the degree can be shown as follows:

$$Embeddedness(node) = degree(node) - 1$$

By using this equation, we can calculate embeddedness more easily.

Question 12.

For this question, we are focusing on the embeddedness and dispersion distribution of the core node's personalized network. For the dispersion calculation, if the distance between two nodes is infinite, we set it as $diameter + constant$, where $diameter$ is the diameter of the original core node's personalized network, and constant is equal to 2 for this question.

For the core node's personalized network using node id 1 as core node, the embeddedness and dispersion can be shown as Fig. 1.33 and Fig. 1.34 respectively as follows.

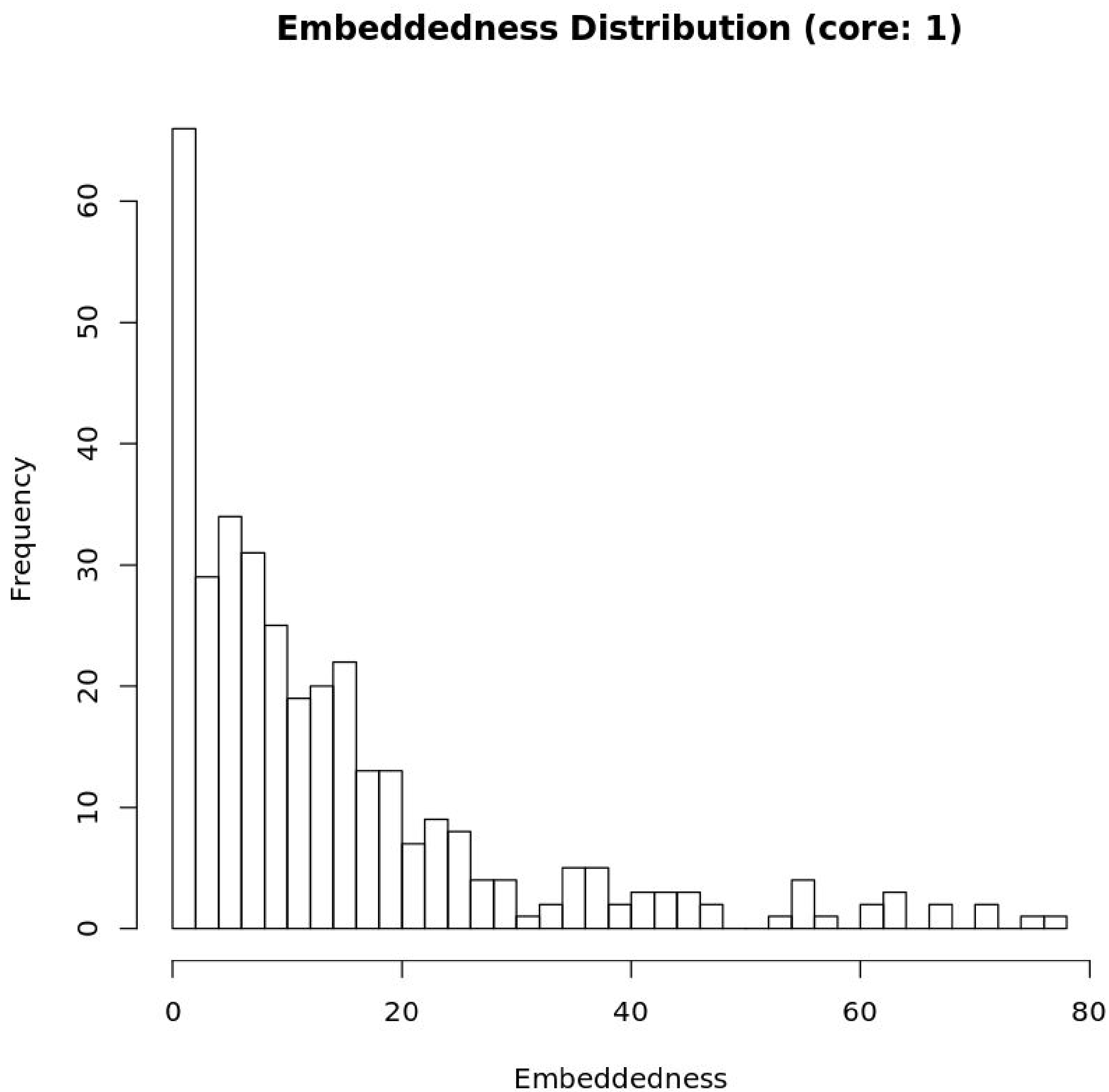

Figure 1.33 The Embeddedness Distribution for the Core Node's Personalized Network Using Node Id 1 as Core Node

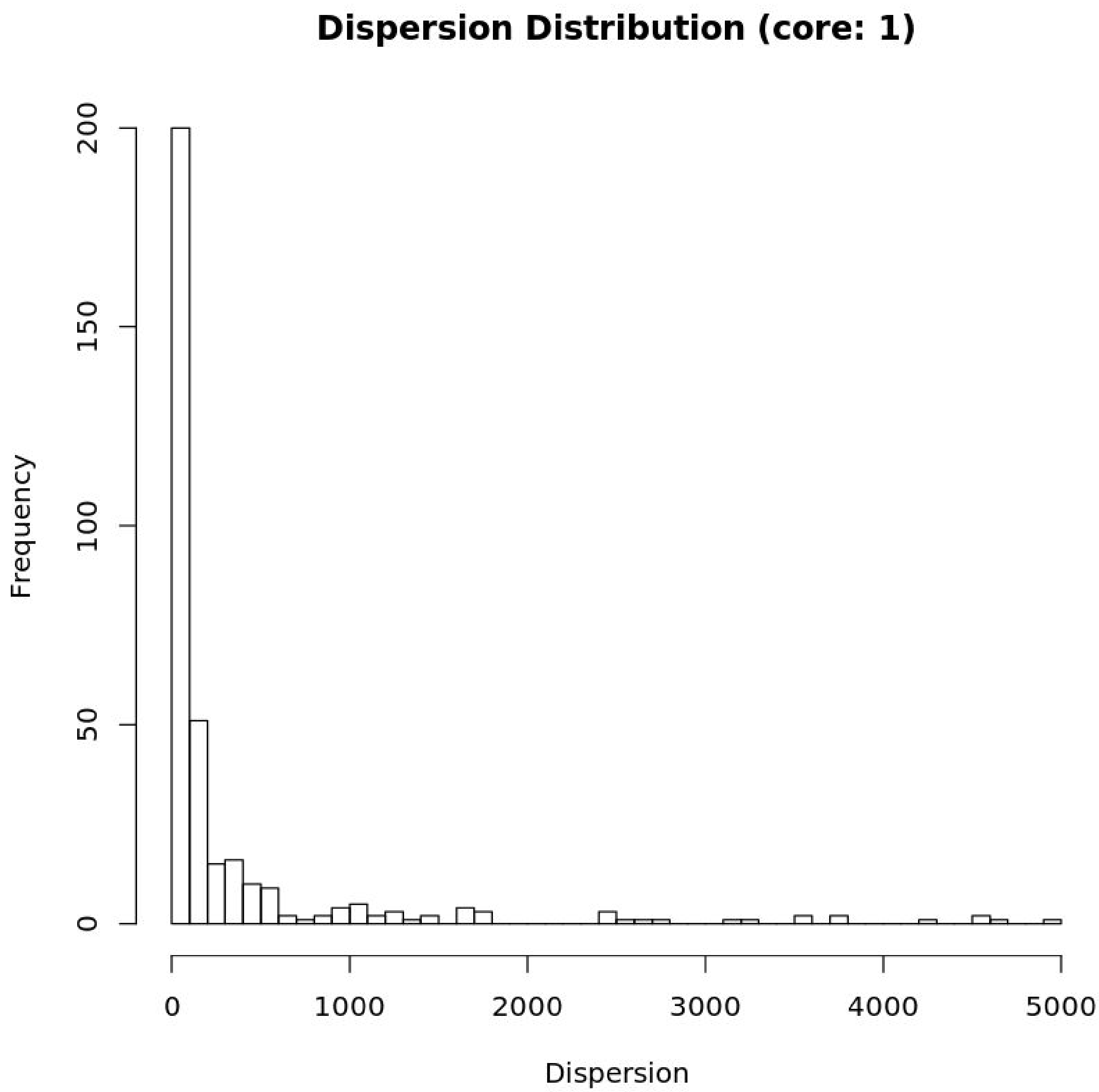

Figure 1.34 The Dispersion Distribution for the Core Node's Personalized Network Using Node Id 1 as Core Node

For the core node's personalized network using node id 108 as core node, the embeddedness and dispersion can be shown as Fig. 1.35 and Fig. 1.36 respectively as follows.

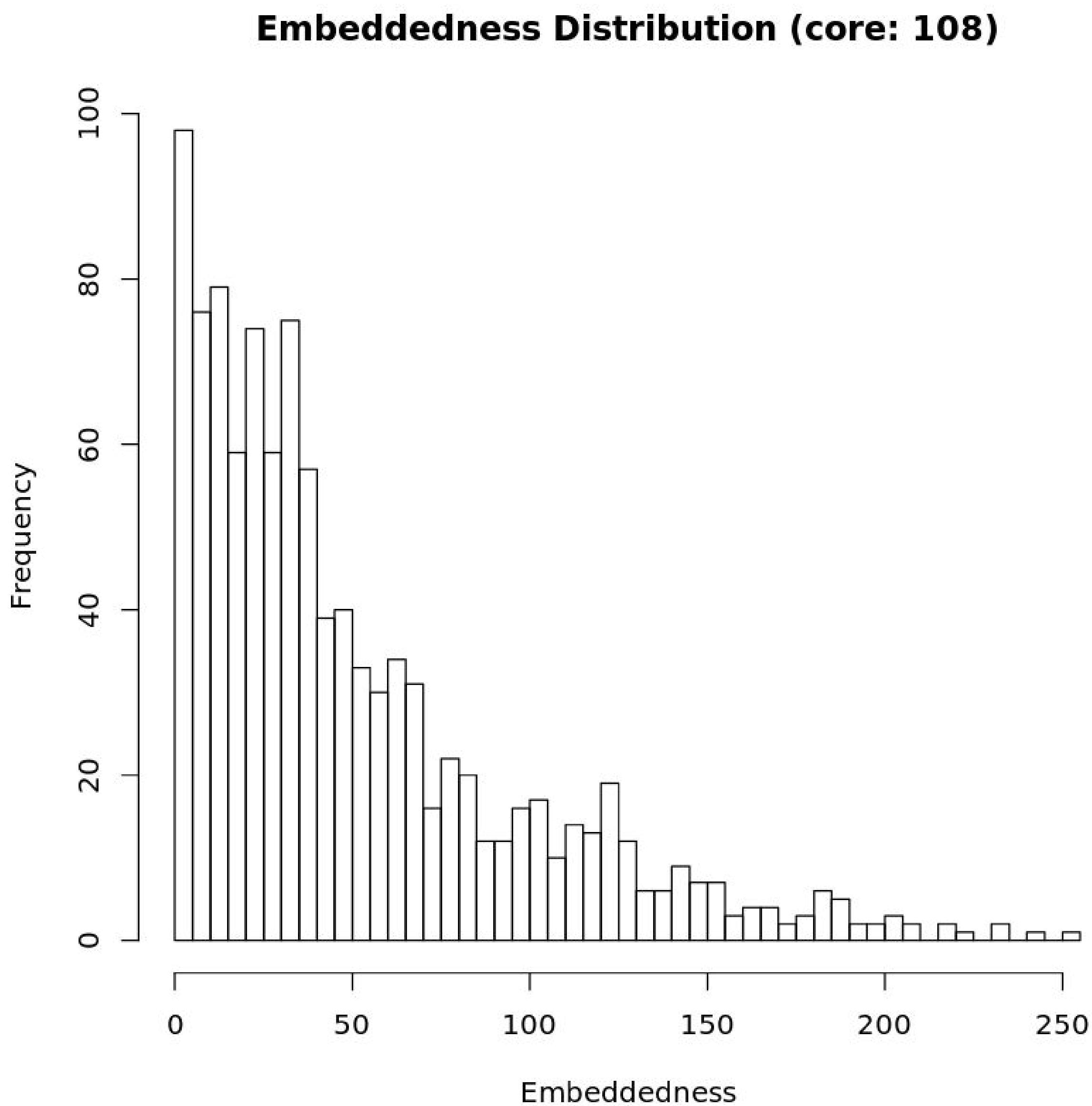

Figure 1.35 The Embeddedness Distribution for the Core Node's Personalized Network Using Node Id 108 as Core Node

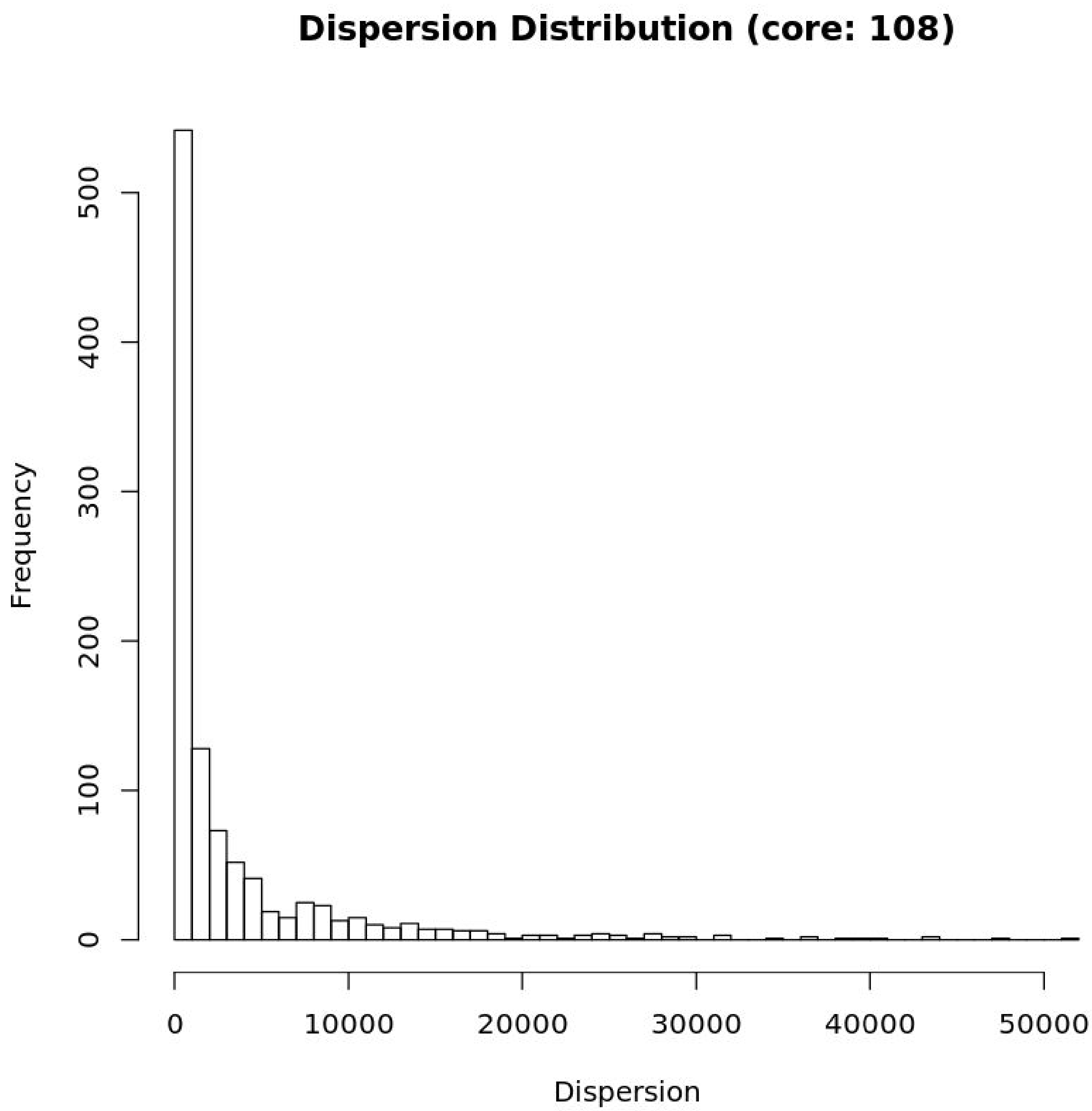

Figure 1.36 The Dispersion Distribution for the Core Node's Personalized Network Using Node Id 108 as Core Node

For the core node's personalized network using node id 349 as core node, the embeddedness and dispersion can be shown as Fig. 1.37 and Fig. 1.38 respectively as follows.

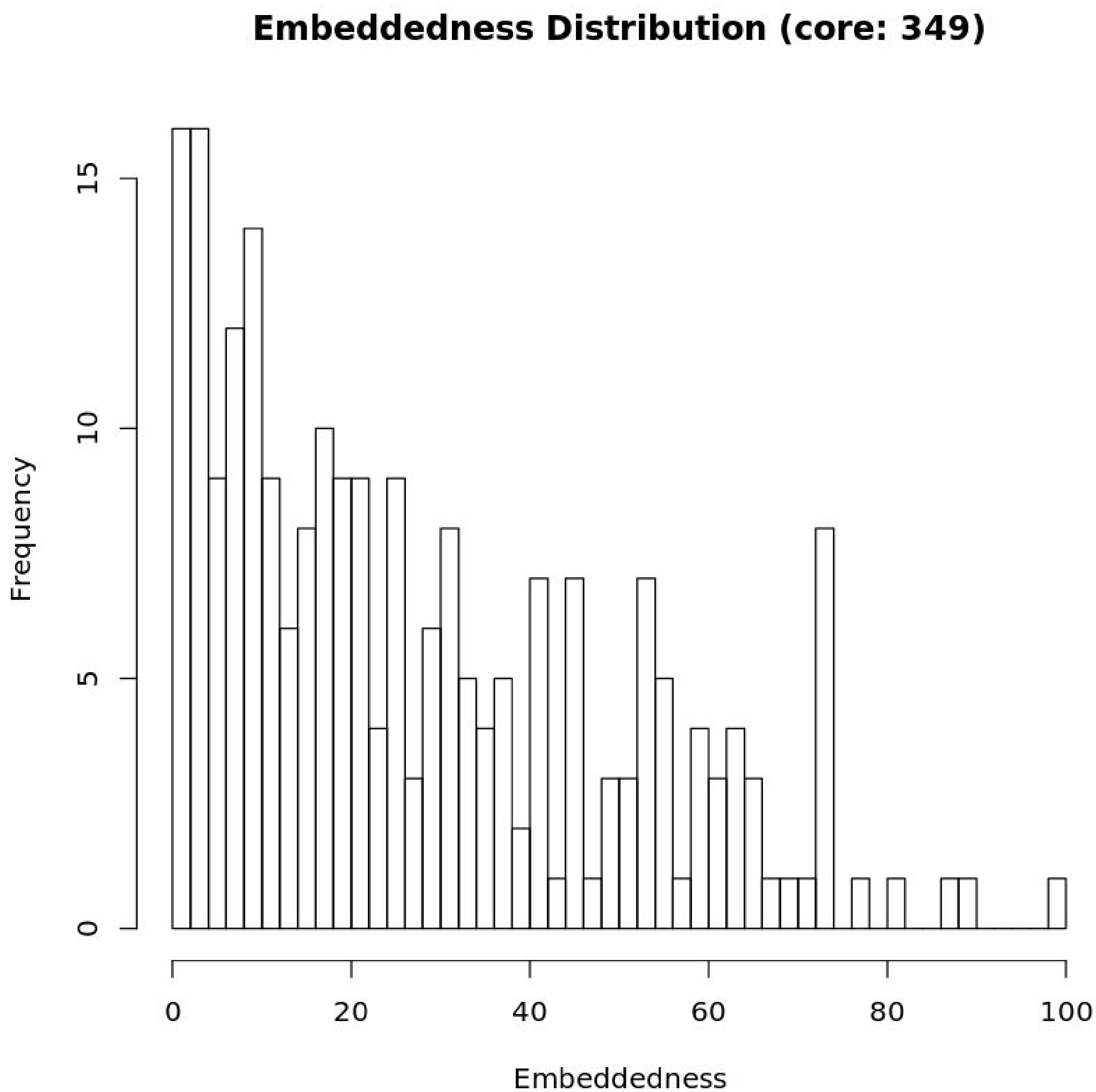

Figure 1.37 The Embeddedness Distribution for the Core Node's Personalized Network Using Node Id 349 as Core Node

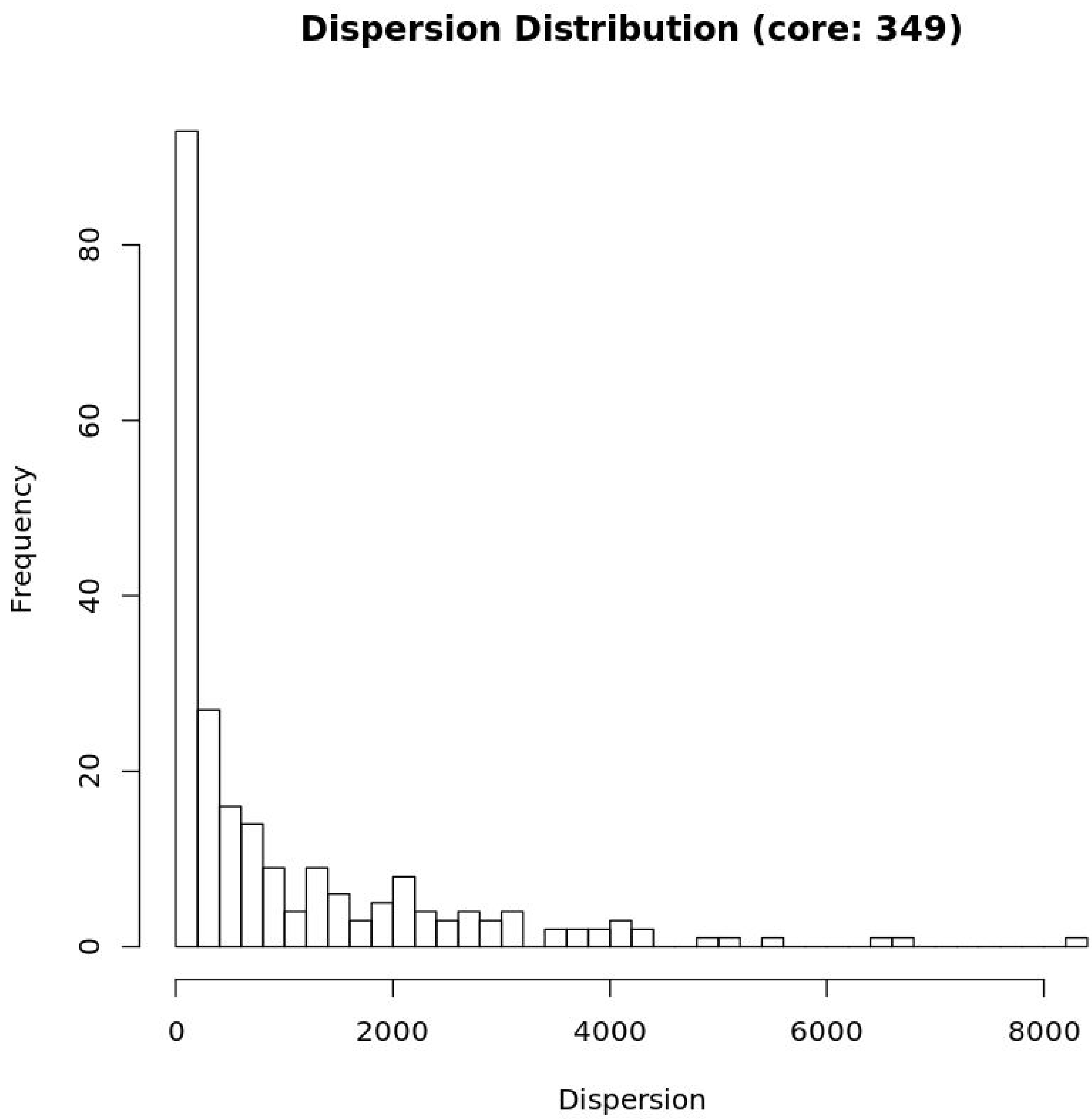

Figure 1.38 The Dispersion Distribution for the Core Node's Personalized Network Using Node Id 349 as Core Node

For the core node's personalized network using node id 484 as core node, the embeddedness and dispersion can be shown as Fig. 1.39 and Fig. 1.40 respectively as follows.

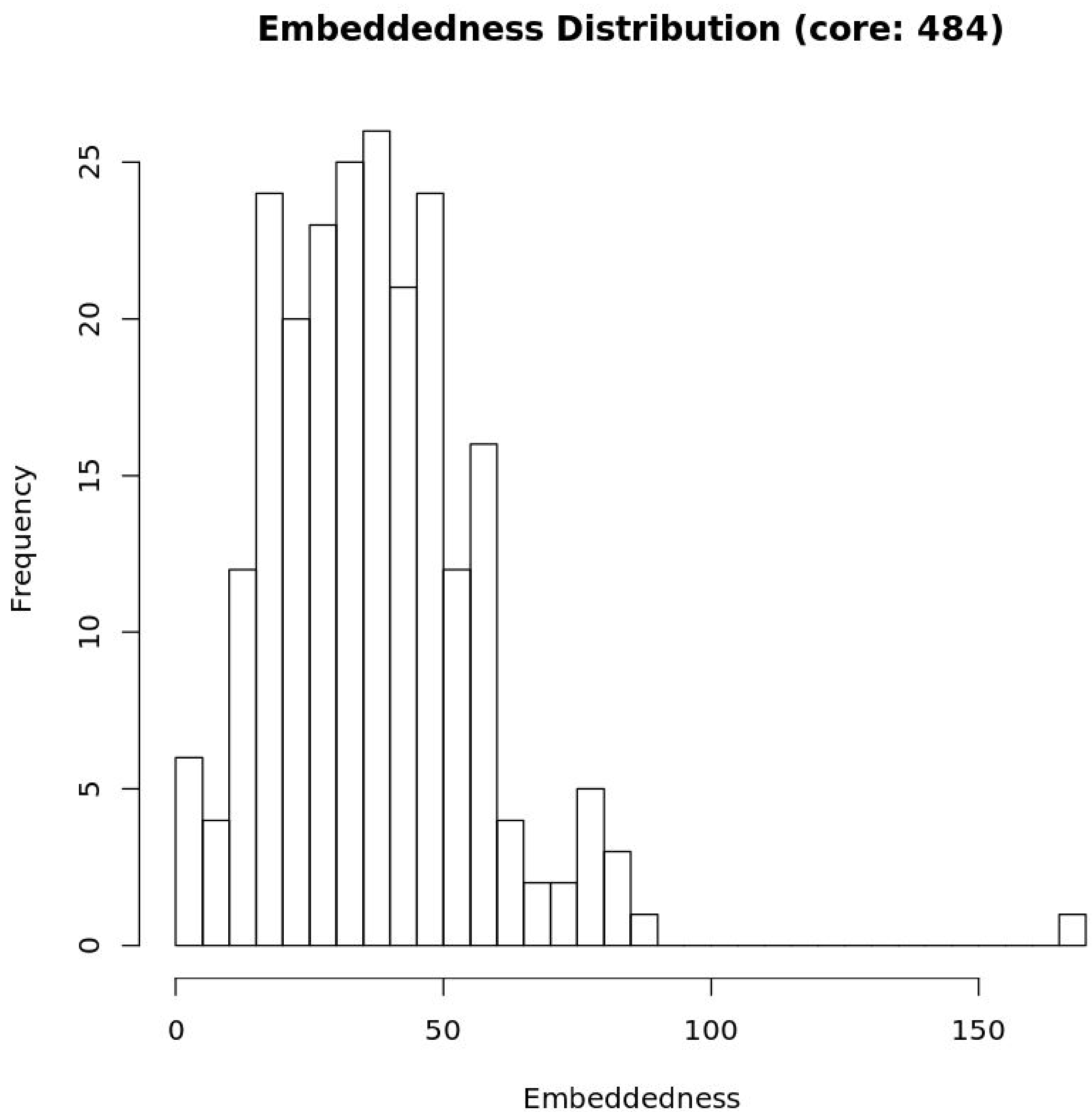

Figure 1.39 The Embeddedness Distribution for the Core Node's Personalized Network Using Node Id 484 as Core Node

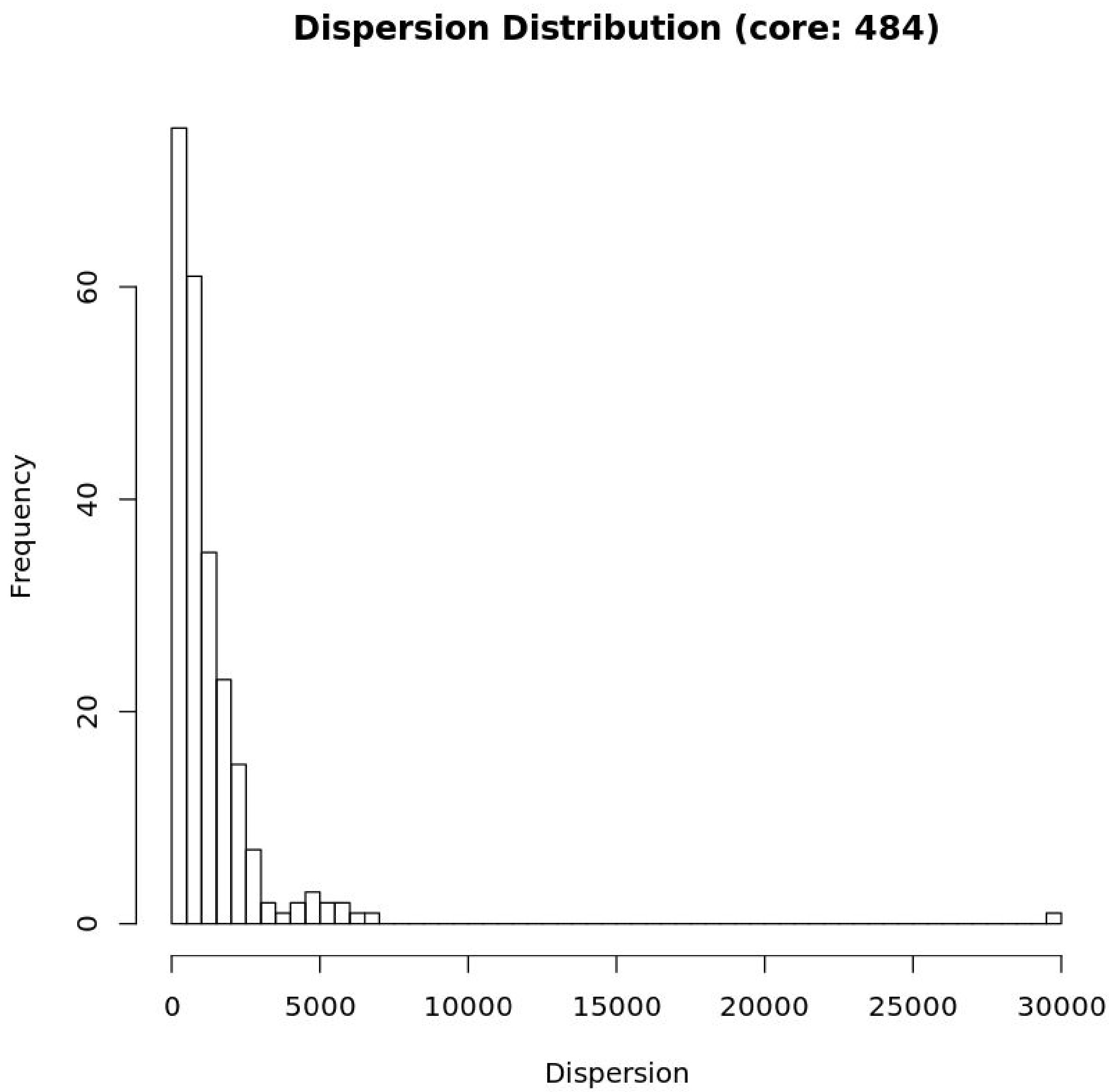

Figure 1.40 The Dispersion Distribution for the Core Node's Personalized Network Using Node Id 484 as Core Node

For the core node's personalized network using node id 1087 as core node, the embeddedness and dispersion can be shown as Fig. 1.41 and Fig. 1.42 respectively as follows.

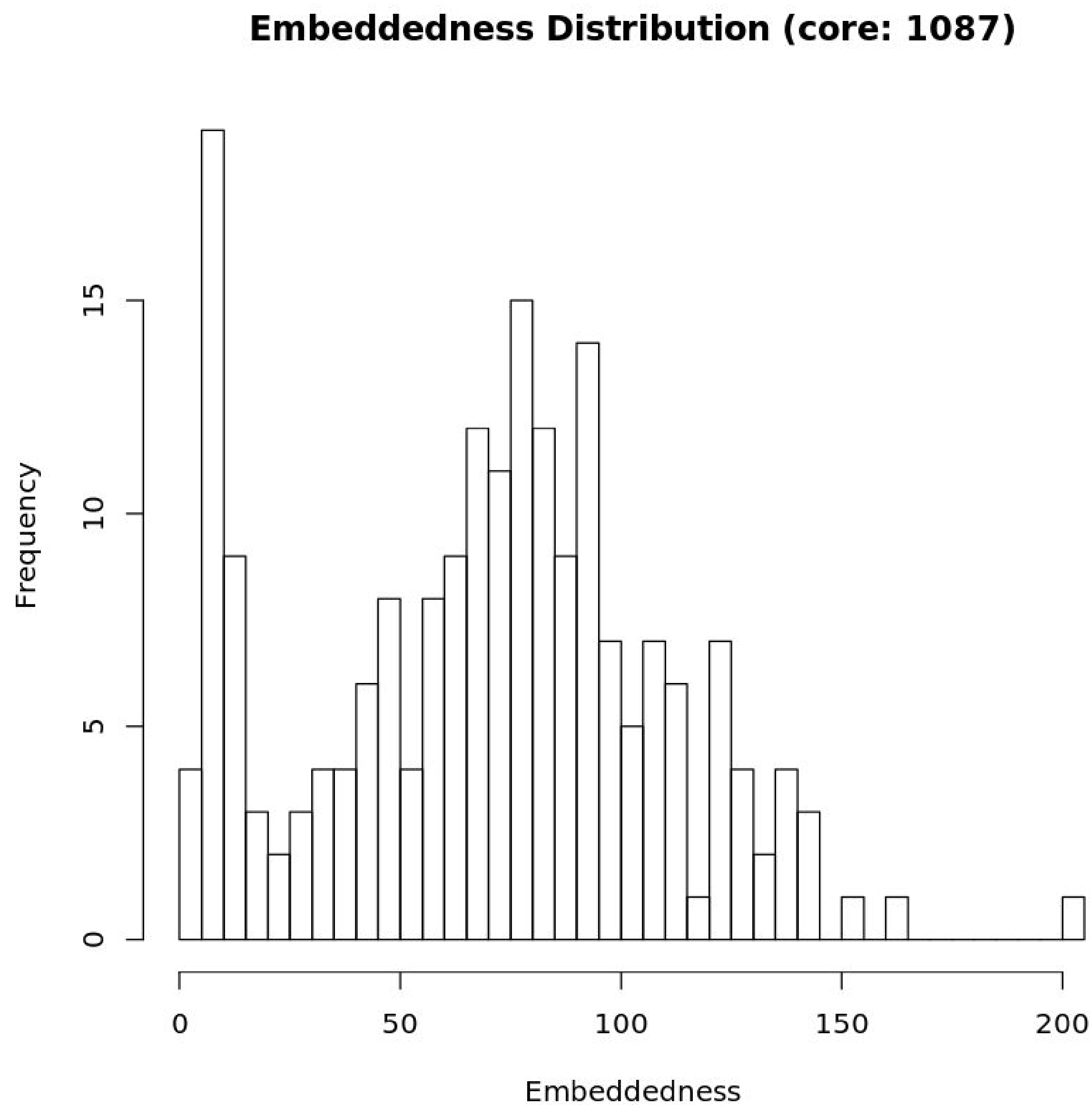

Figure 1.41 The Embeddedness Distribution for the Core Node's Personalized Network Using Node Id 1087 as Core Node

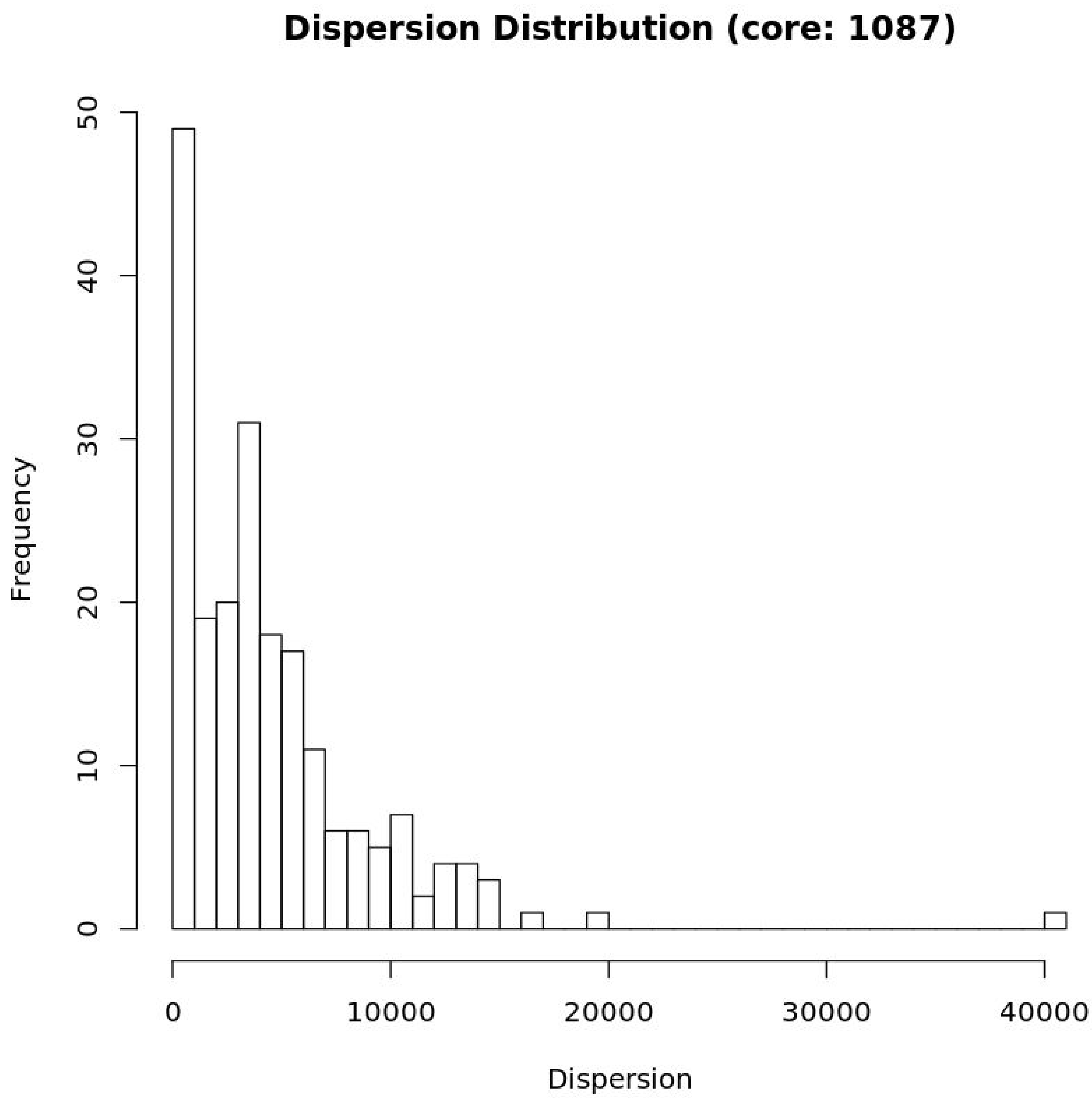

Figure 1.42 The Dispersion Distribution for the Core Node's Personalized Network Using Node Id 1087 as Core Node

Question 13.

For this question, we plotted the community structure (using Fast-Greedy algorithm) of the personalized network using colors and highlight the node with maximum dispersion. We tests the core node's personalized network (use the same core nodes as Question 9). The result can be shown from Fig. 1.43 to Fig. 1.47.

In each graphs, the red nodes represents the node with maximum dispersion and the green nodes represents the core node. Besides, the blue edges represents represents the edges incident to the node with maximum dispersion.

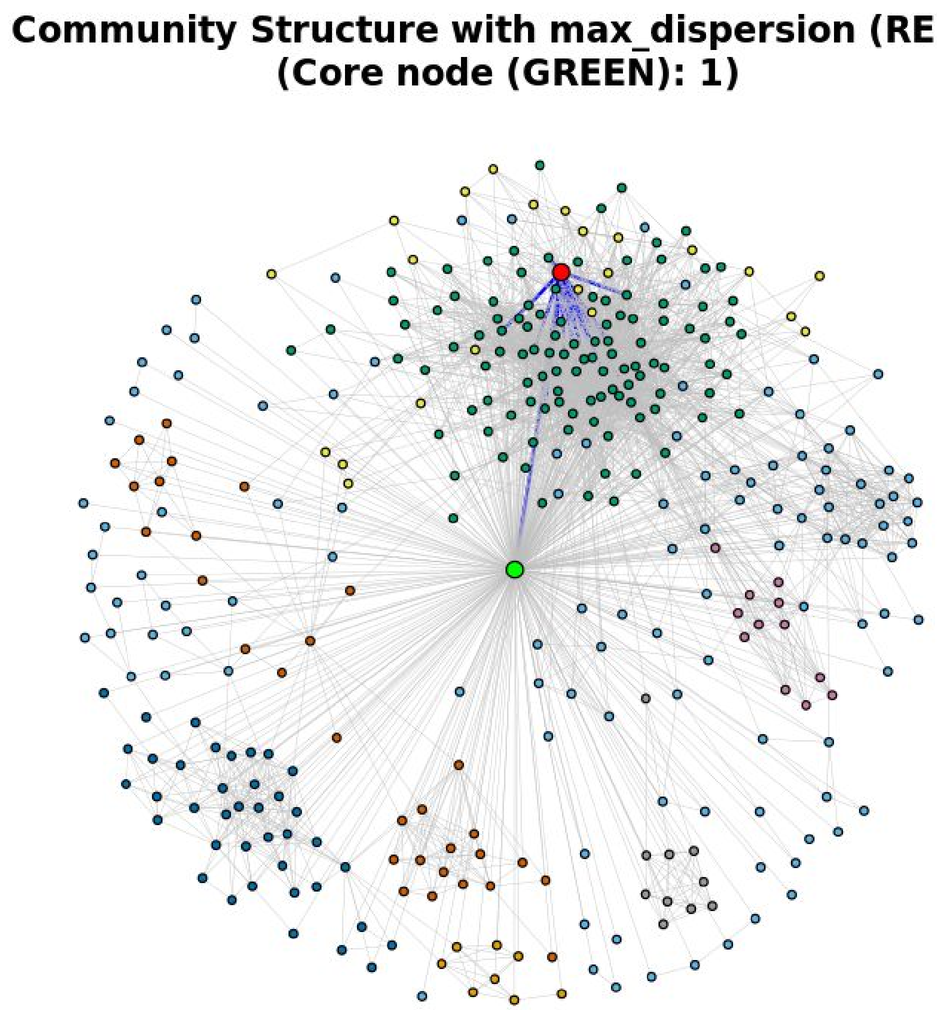

Figure 1.43 The Community Structure of the Core Node's Personalized Network Using Node Id 1 as Core Node (The Node with Maximum Dispersion is Highlighted)

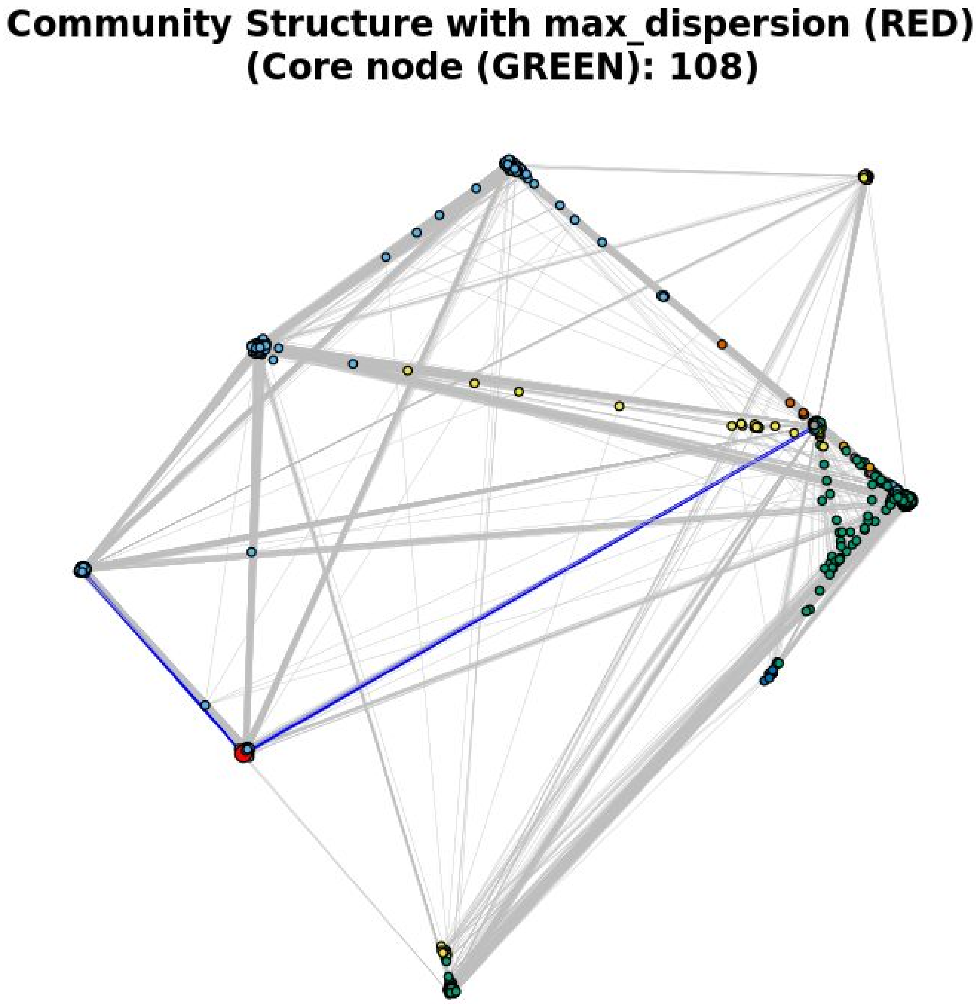

Figure 1.44 The Community Structure of the Core Node's Personalized Network Using Node Id 108 as Core Node (The Node with Maximum Dispersion is Highlighted)

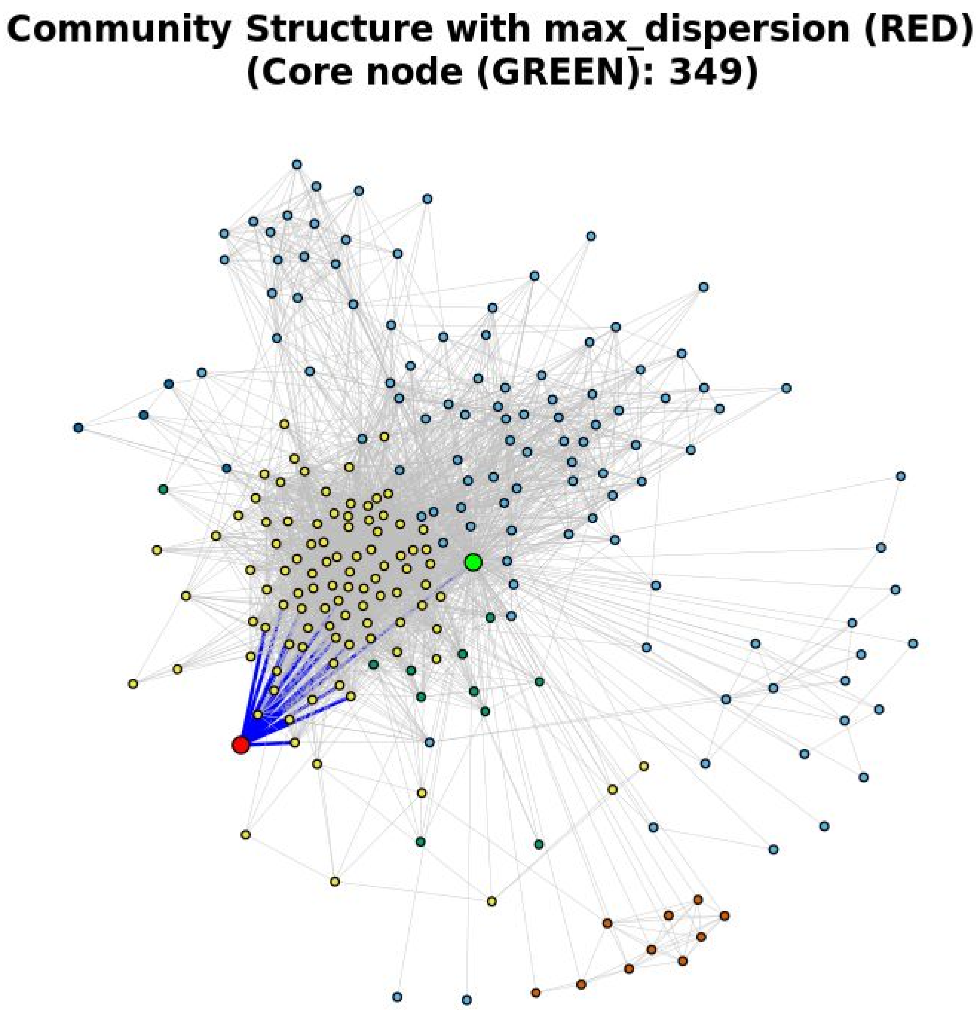

Figure 1.45 The Community Structure of the Core Node’s Personalized Network Using Node Id 349 as Core Node
(The Node with Maximum Dispersion is Highlighted)

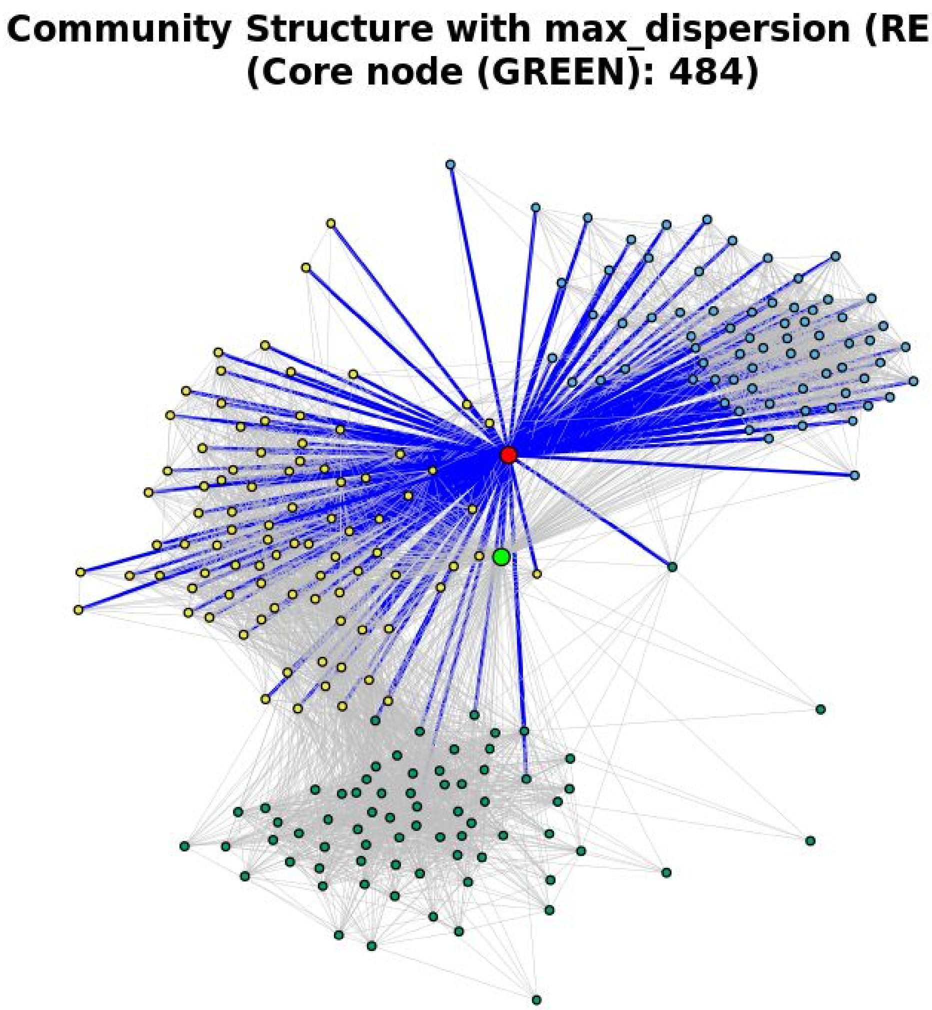

Figure 1.46 The Community Structure of the Core Node’s Personalized Network Using Node Id 484 as Core Node
(The Node with Maximum Dispersion is Highlighted)

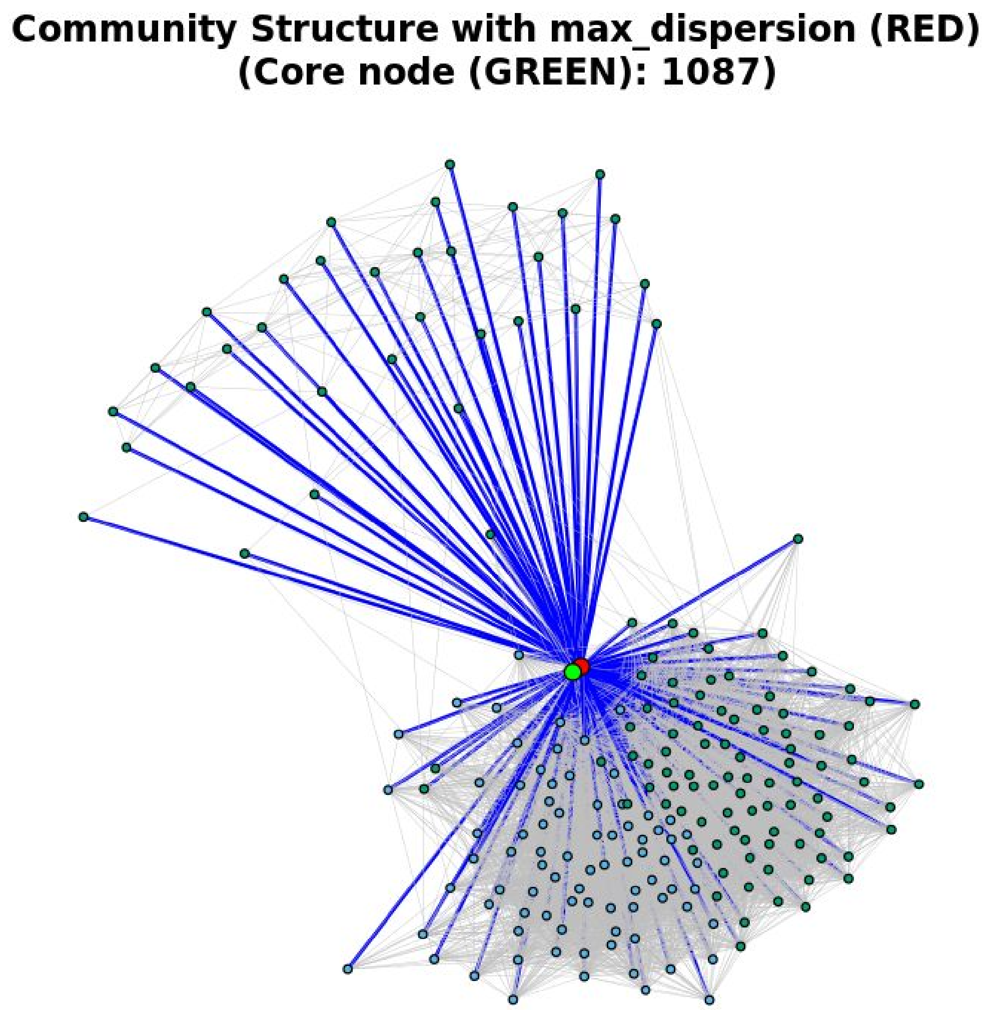

Figure 1.47 The Community Structure of the Core Node's Personalized Network Using Node Id 1087 as Core Node (The Node with Maximum Dispersion is Highlighted)

Question 14.

For this question, we plotted the community structure (using Fast-Greedy algorithm) of the personalized network using colors and highlight the node with maximum embeddedness and the node with maximum $\frac{dispersion}{embeddedness}$.

We first plotted the community structure with node withimum embeddedness highlighted (The result can be shown from Fig. 1.48 to Fig. 1.52). In each graphs, the red nodes represents the node with maximum embeddedness and the green nodes represents the core node. Besides, the blue edges represents represents the edges incident to the node with maximum embeddedness.

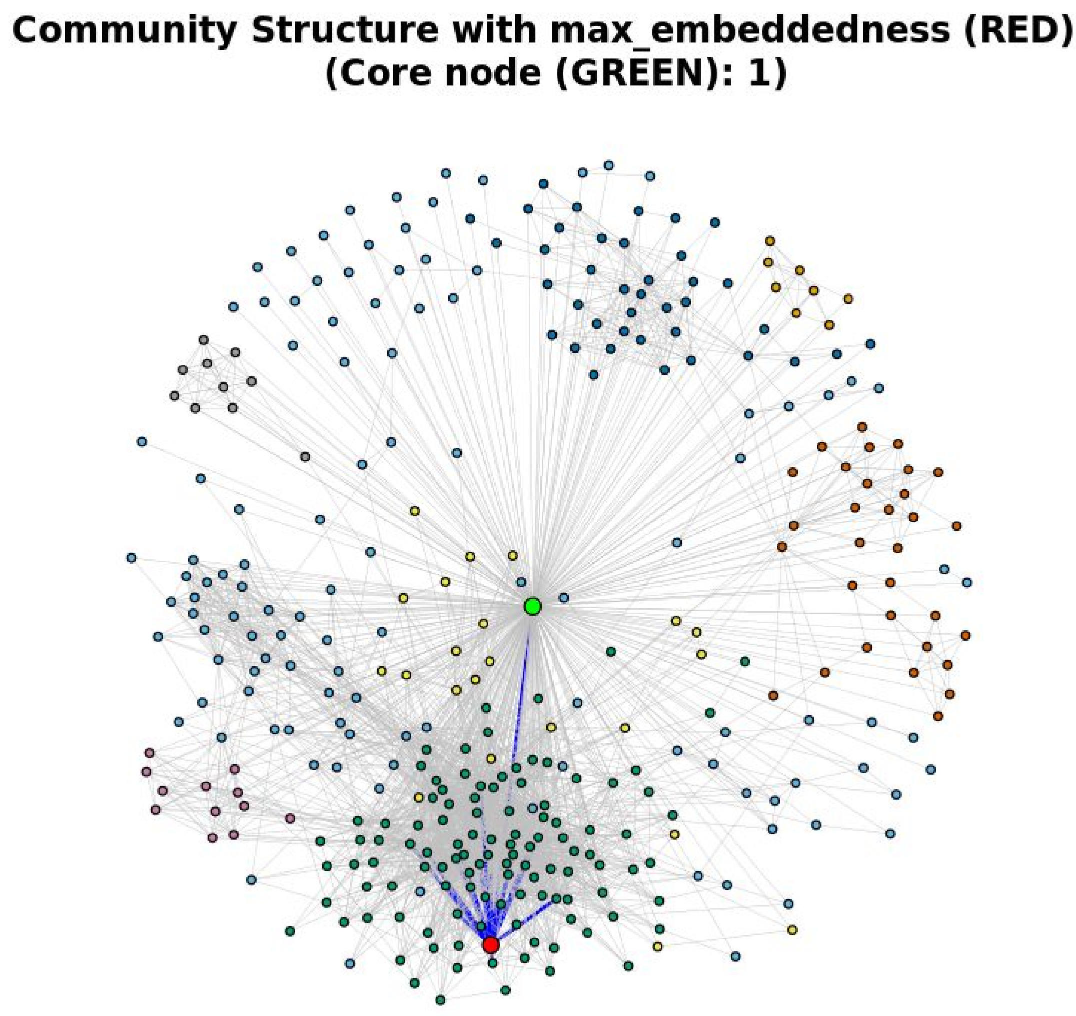

Figure 1.48 The Community Structure of the Core Node's Personalized Network Using Node Id 1 as Core Node (The Node with Maximum Embeddedness is Highlighted)

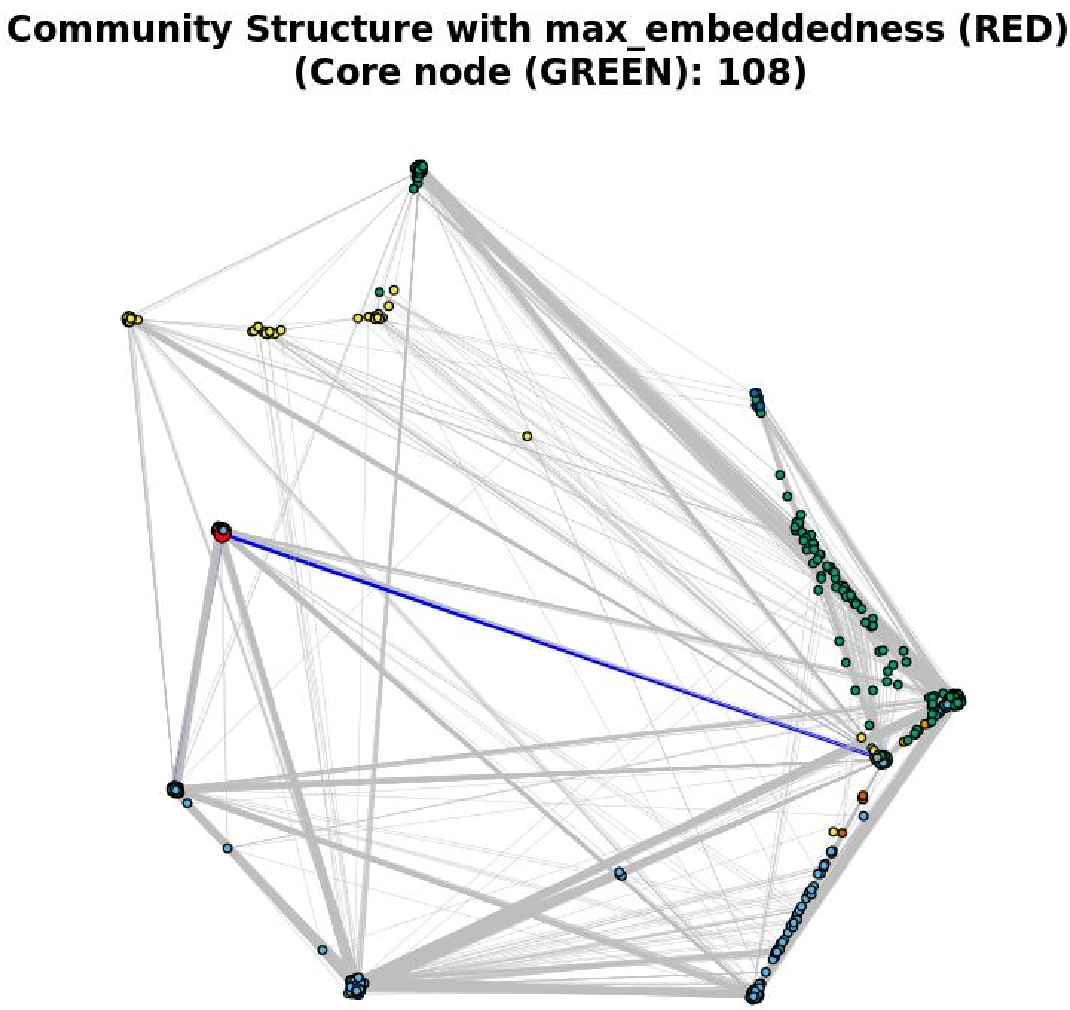

Figure 1.49 The Community Structure of the Core Node's Personalized Network Using Node Id 108 as Core Node (The Node with Maximum Embeddedness is Highlighted)

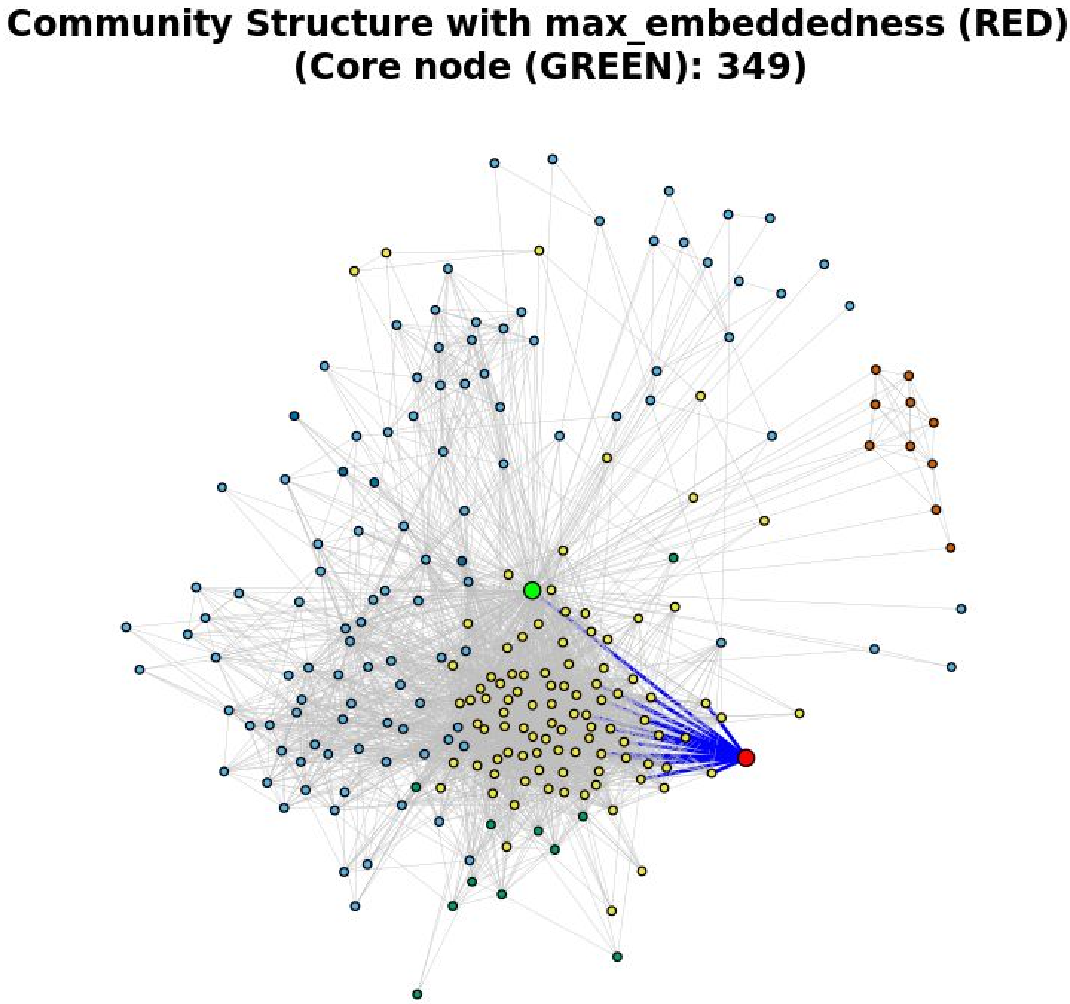

Figure 1.50 The Community Structure of the Core Node's Personalized Network Using Node Id 349 as Core Node (The Node with Maximum Embeddedness is Highlighted)

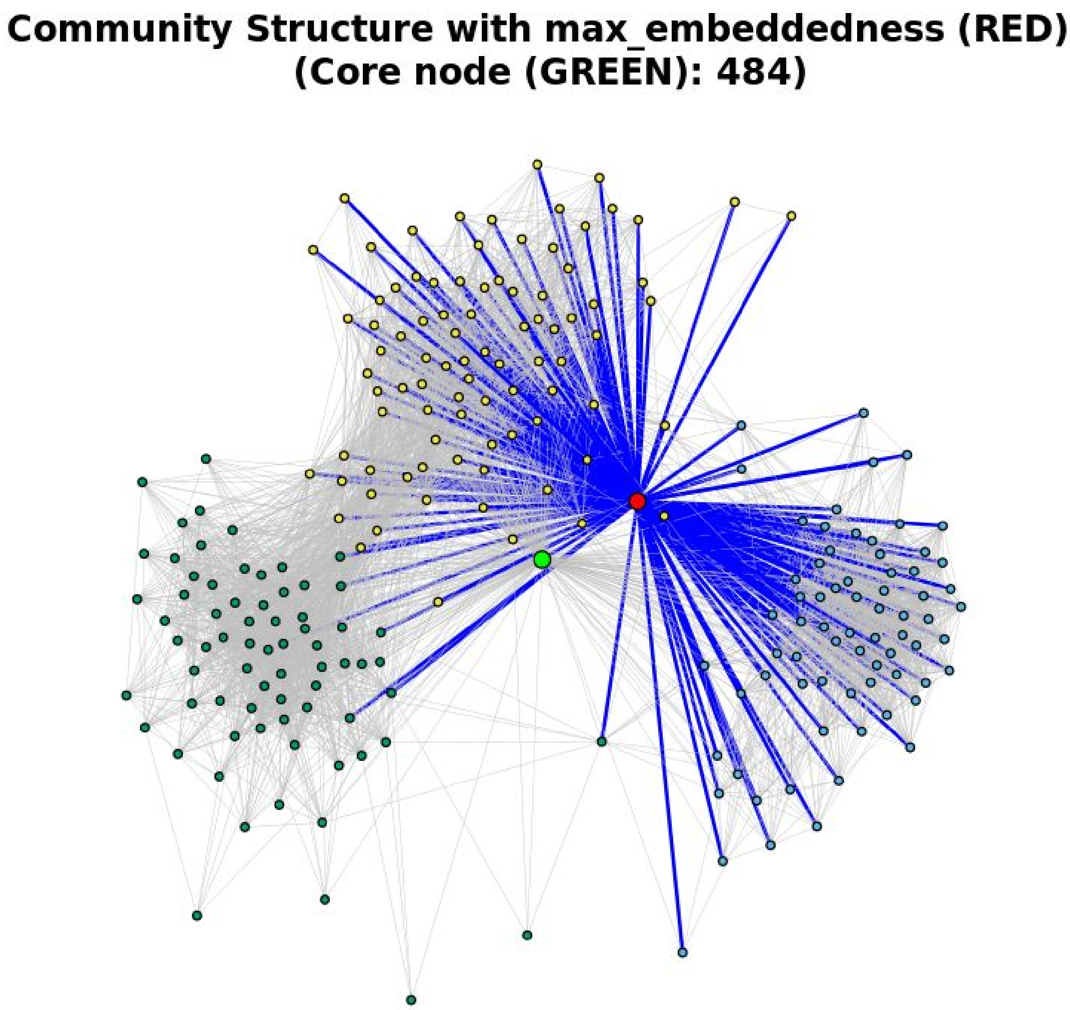

Figure 1.51 The Community Structure of the Core Node’s Personalized Network Using Node Id 484 as Core Node (The Node with Maximum Embeddedness is Highlighted)

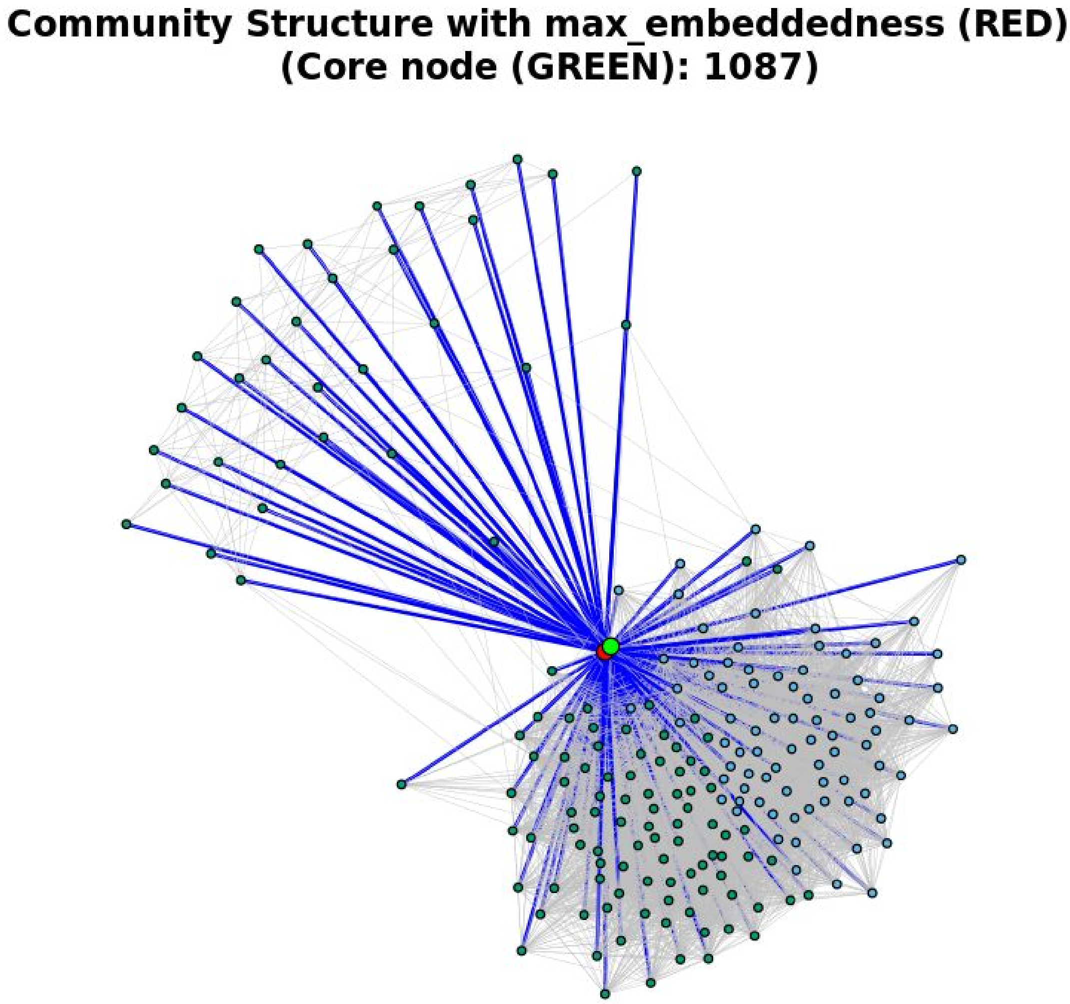

Figure 1.52 The Community Structure of the Core Node's Personalized Network Using Node Id 1087 as Core Node (The Node with Maximum Embeddedness is Highlighted)

Then, we plotted the community structure with node withimum $\frac{dispersion}{embeddedness}$ highlighted (The result can be shown from Fig. 1.43 to Fig. 1.57). In each graphs, the red nodes represents the node with maximum $\frac{dispersion}{embeddedness}$ and the green nodes represents the core node. Besides, the blue edges represents represents the edges incident to the node with maximum $\frac{dispersion}{embeddedness}$.

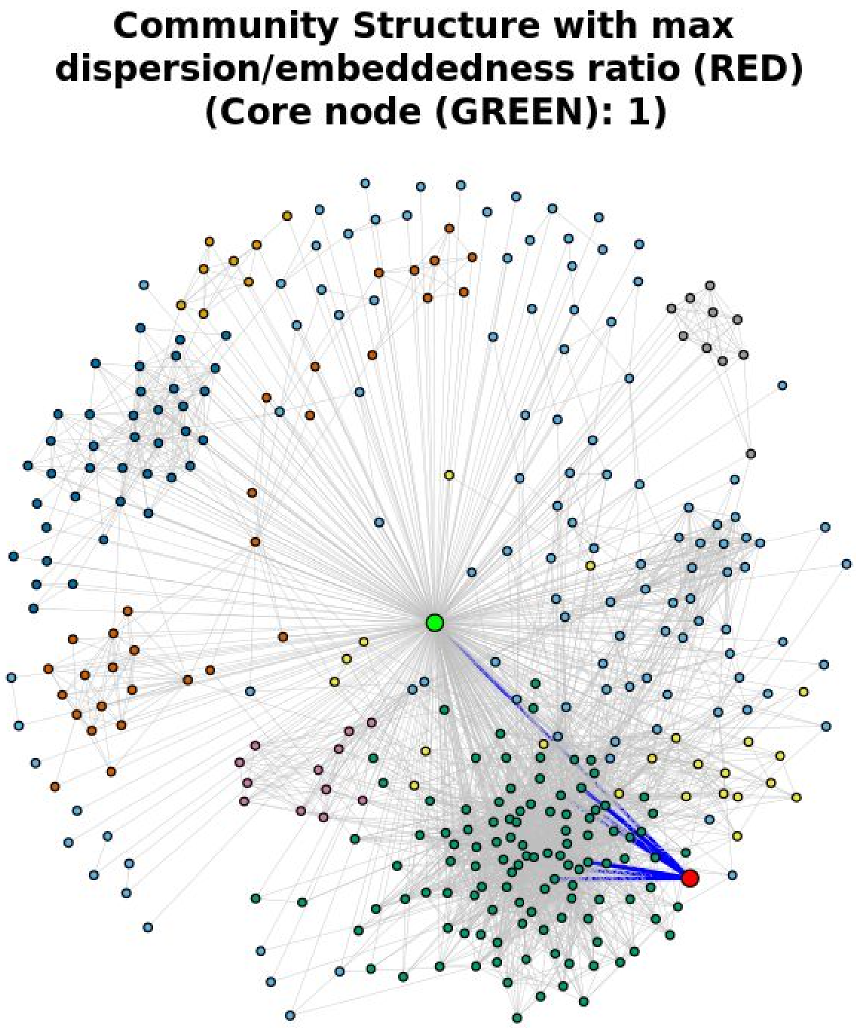

Figure 1.53 The Community Structure of the Core Node's Personalized Network Using Node Id 1 as Core Node (The Node with Maximum Dispersion/Embeddedness is Highlighted)

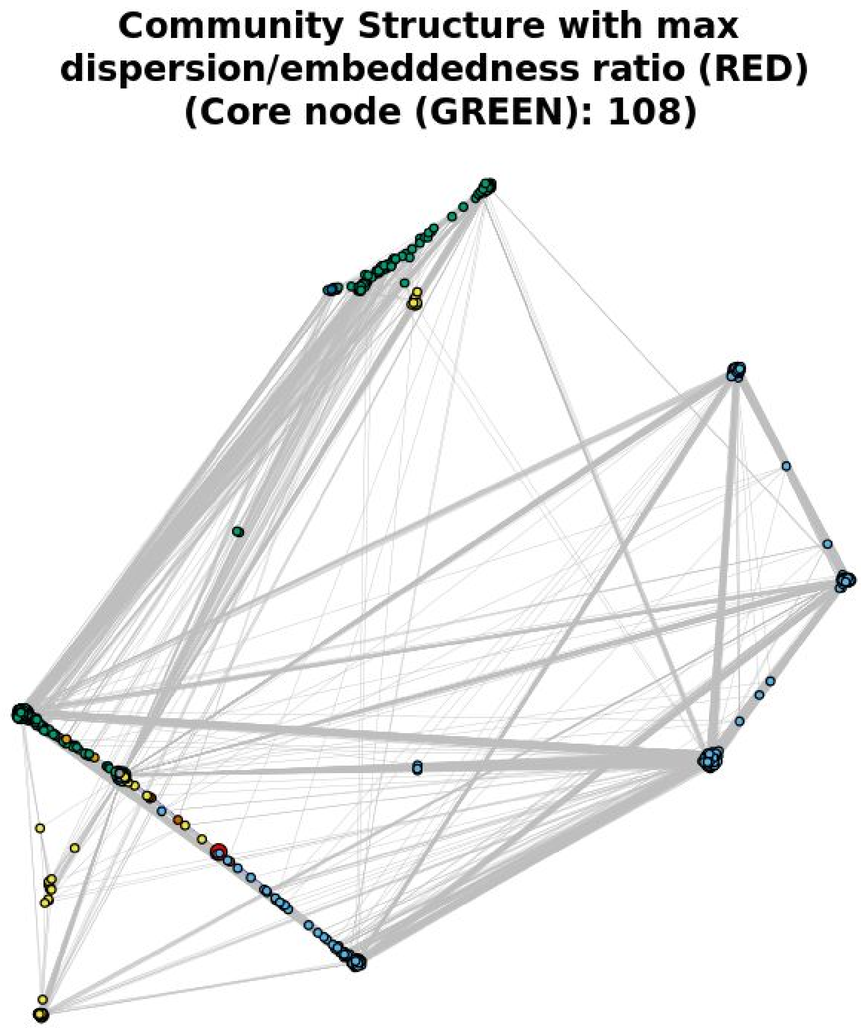

Figure 1.54 The Community Structure of the Core Node's Personalized Network Using Node Id 108 as Core Node (The Node with Maximum Dispersion/Embeddedness is Highlighted)

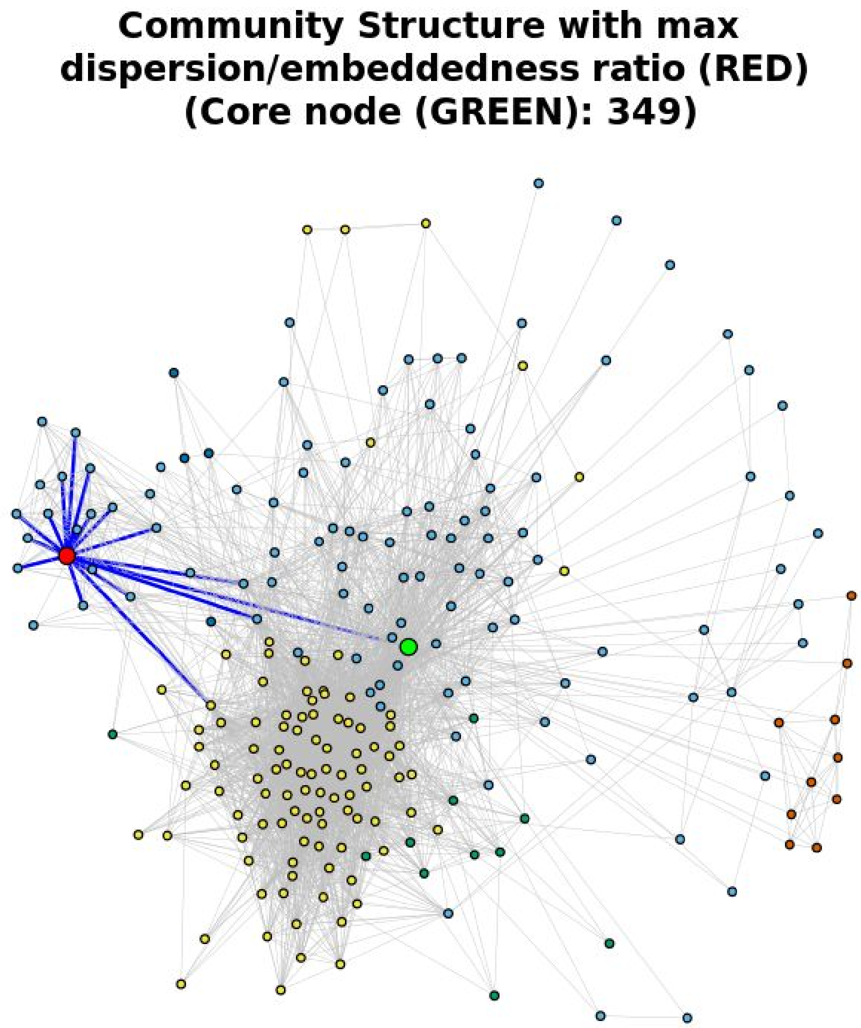

Figure 1.55 The Community Structure of the Core Node's Personalized Network Using Node Id 349 as Core Node (The Node with Maximum Dispersion/Embeddedness is Highlighted)

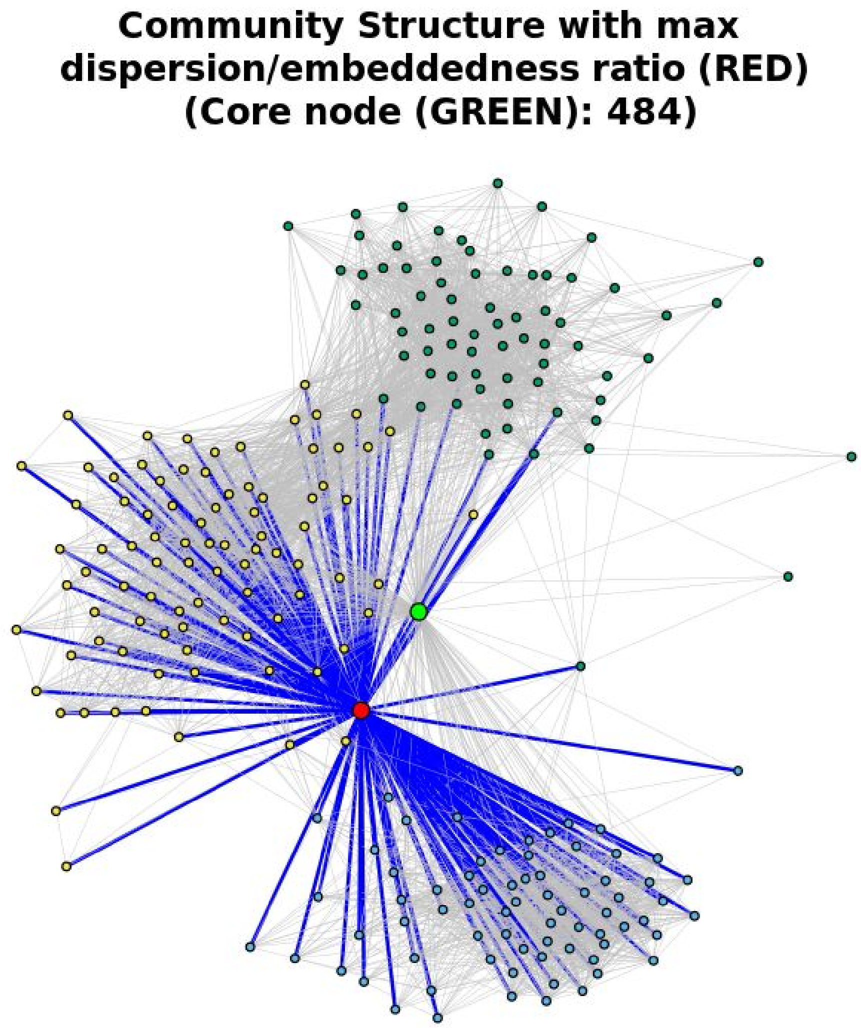

Figure 1.56 The Community Structure of the Core Node's Personalized Network Using Node Id 484 as Core Node
(The Node with Maximum Dispersion/Embeddedness is Highlighted)

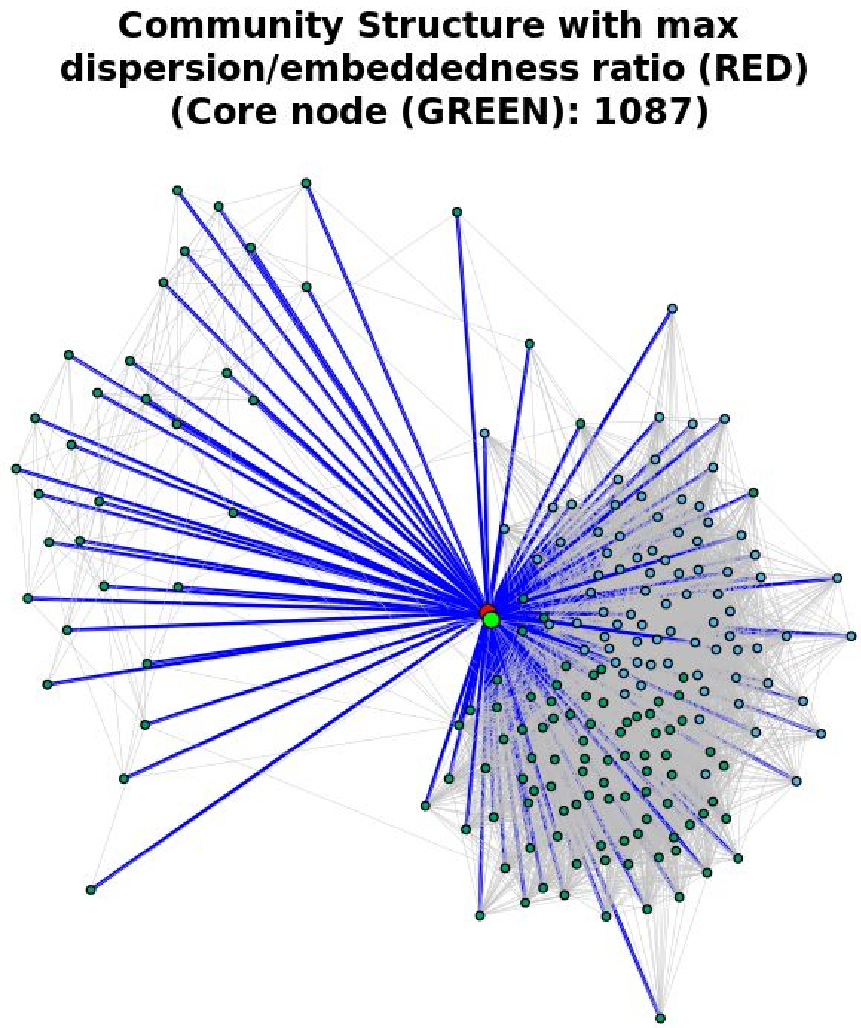

Figure 1.57 The Community Structure of the Core Node's Personalized Network Using Node Id 1087 as Core Node (The Node with Maximum Dispersion/Embeddedness is Highlighted)

Question 15.

We first examined all these measurement through definition.

The embeddedness is the number of mutual friends. This reveals how much two nodes' social circle overlap, using this to indicate the relevance of two nodes. However, this do have a limitation, for example, a node with more neighbours tends to be more relevant comparing to a node with little neighbours. For example, A and B are couples, but B are very shy, and thus have little mutual friends with A. Now considering that C is very outward, and have a lot of friend with A. In this case, A and C are more relevant comparing to A and B.

The dispersion is another measurement. It is better than embeddedness because it looks not only the mutual friends but also the network of the structure. Two people are more relevant in the case if they have the same colleagues, classmates, private members, and so on than the case that they only have the same colleagues. The limitation for dispersion is the same as embeddedness. Because the dispersion is proportion to the embeddedness, which means the more mutual friends they have, the more dispersion they have, which we show it is not reasonable sometimes in previous paragraph.

Thus, to better measure the relevance, we $\frac{dispersion}{embeddedness}$. In this case, the limitation of only using embeddedness or dispersion can be solved. Because they measure the measure the dispersion per mutual friend.

After examining the measurements through definition, we saw how things work on our graph shown in Q13 and Q14. First, we can see that for both five core node (1, 108, 349, 484, 1087), the max embeddedness and max dispersion node looks similar. This show that our explanation is correct, which says that the more mutual friends they have, the more dispersion they have.

Besides, by comparing with the max $\frac{dispersion}{embeddedness}$ with other two types of graph (max dispersion and max embeddedness), we can see that they are indeed more relevant to their core nodes, closer to the core nodes and their mutual friend are more separated (which indicate that they are indeed more relevant).

In summary, in the performance of finding the relevance of two node, the measurement using $\frac{dispersion}{embeddedness}$ performs the best, then the measurement using the dispersion. The measurement using the embeddedness performs the worst.

Question 16.

In the fourth problem of the first part, we want to predict future links between nodes in the network, which is equivalent to recommend new friends in the Facebook network. We have three neighborhood-based measures for friend recommendation, which are common neighbor measure, Jaccard measure, Adamic-Adar measure. These three measures need two neighbors set of nodes in the network as input. Common neighbor measure is to calculate the length of the intersection of two neighbor sets. Jaccard measure is to calculate the result of the length of the intersection of two neighbor sets divided by the length of the union of two neighbor sets. Adamic-Adar measure is to calculate the sum of the result that 1 divided by log of neighbor set of the node, which is all node contained in the intersection of the first two neighbor sets. Then we state the recommendation method below. We want to recommend t new friends to some user k in the network using Jaccard measure. The first step is that for each node in the network that is not a neighbor of k, we compute the Jaccard measure between the node k and the node not in the neighborhood of k. The second step is we pick t nodes that have the highest Jaccard measure with node k and recommend these nodes as friends to node k.
In this question, we want to specifically analyze the personalized network of node with ID 415. We create the list of users who we want to recommend new friends to, all nodes which are degree 24. We denote this list as Nr, and Nr is an 11-element array with user node ID 31, 53, 75, 90, 93, 102, 118, 133, 134, 136, and 137.

Question 17.

In this question, we will apply the 3 different types of friend recommendation algorithms to recommend friends to the users in the list Nr. We will define an average accuracy measure to compare the performances of the friend recommendation algorithms. In order to compute the average accuracy of the friend recommendation algorithm, we perform this task in two steps: the first step is to compute the average accuracy for each user in the list Nr; the second step is to compute the average accuracy of the algorithm by averaging across the accuracies of the users in the list Nr. For the first step, we complete it in the following procedure. Firstly, we remove each edge of node at random with probability 0.25, and the list of friends deleted are denoted as R. Then we use those three methods stated above to recommend the same number of new friends as R to the user k. Those recommend friends are denoted as P. The accuracy for the user k for this iteration is given by the length of the intersection between P and R divided by the length of R. By iterating over the above steps for 10 times and then take the average, we can get the average accuracy of user k. The final result of the average accuracy by using those three methods are:

0.827644 for the common neighbor method, 0.7966263 for the Jaccard method, and 0.8138633 for the Adamic Adar method.

Part 2: Google+ Network

In this part, we will explore the structure of the Google+ network. The dataset for creating the network can be found in the link in http://snap.stanford.edu/data/egonets-Gplus.html.

Create directed personal networks for users who have more than 2 circles. The data required to create such personal networks can be found in the file named gplus.tar.gz.

Question 18.

By counting the number of circles in each .circle file, we found out that there are 57 personal networks that have more than 2 circles.

Question 19.

For this question, we examined the in and out degree distribution for the following node:

- 109327480479767108490
- 115625564993990145546
- 101373961279443806744

The in and out degree distribution for the net 109327480479767108490 can be shown in Fig 2.1.

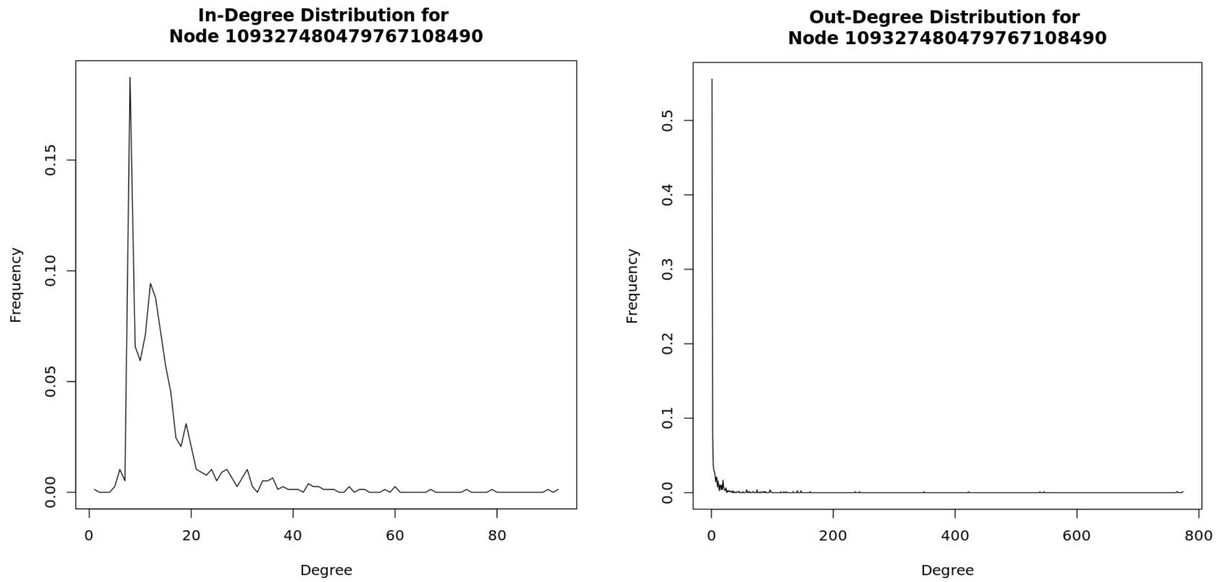

Figure 2.1 The In (left) and Out (right) Degree Distribution for Node 109327480479767108490

The in and out degree distribution for the net 115625564993990145546 can be shown in Fig 2.2.

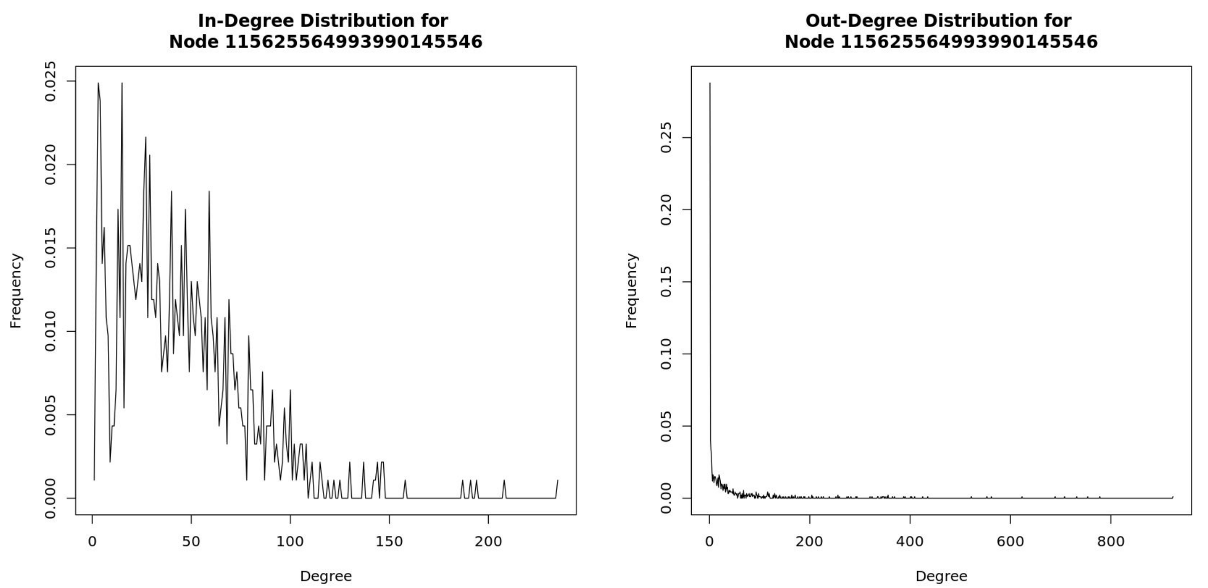

Figure 2.2 The In (left) and Out (right) Degree Distribution for Node 115625564993990145546

The in and out degree distribution for the net 101373961279443806744 can be shown in Fig 2.3.

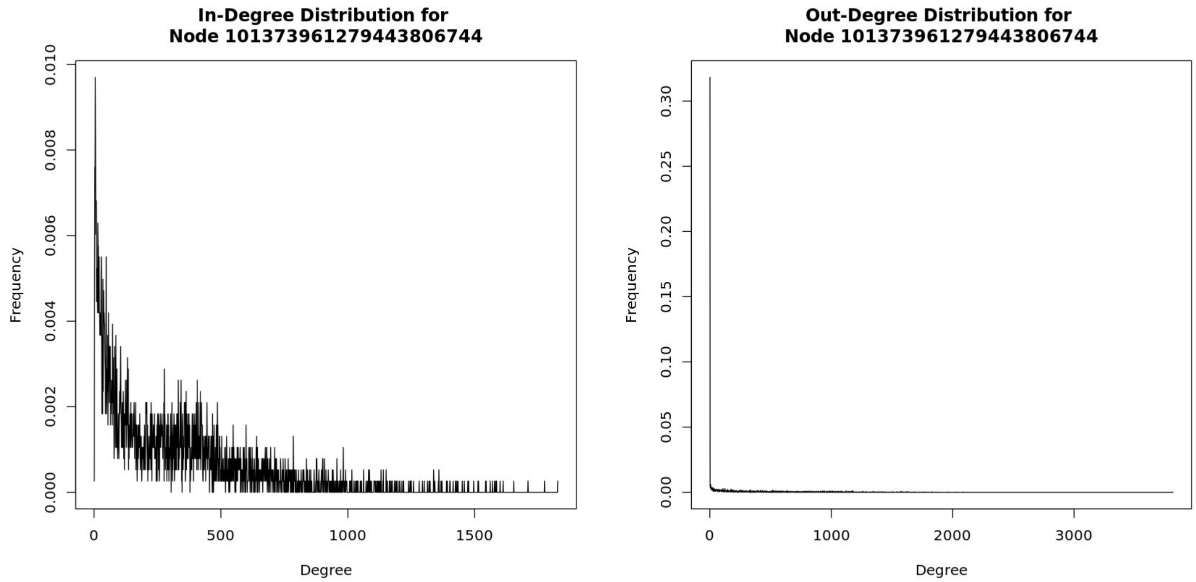

Figure 2.3 The In (left) and Out (right) Degree Distribution for Node 101373961279443806744

From Fig 2.1, Fig 2.2, and Fig 2.3, we observed that these nodes personal networks have a similar in and out degree distribution. For in degree distribution, all the three personal networks show the same pattern, there is a peek in 20-50, before the peek the frequency is increasing while after the peek the frequency is decreasing. For the out degree, in all three personal network, when the degree is low, the frequency is very high, and the frequency keeps decreasing as degree increases.

Question 20.

For this part, we used the Walktrap community detection algorithm to find the community structure of the personal networks. We tested the algorithms on different personal networks (109327480479767108490, 115625564993990145546, and 101373961279443806744) and the modularity scores can be shown in table 2.1.

Personal Network ID	Modularity Scores Using Walktrap Community Detection Algorithm
109327480479767108490	0.252765
115625564993990145546	0.319473

101373961279443806744	0.191090

Table 2.1 The Modularity Scores for Different Personal Networks Using Walktrap Community Detection Algorithm

From table 2.1, it can be shown that the modularity scores are not similar to each other.

After calculating the modularity scores, we plotted the communities for the personal networks. The result can be shown in Fig 2.4, Fig 2.5, and Fig 2.6.

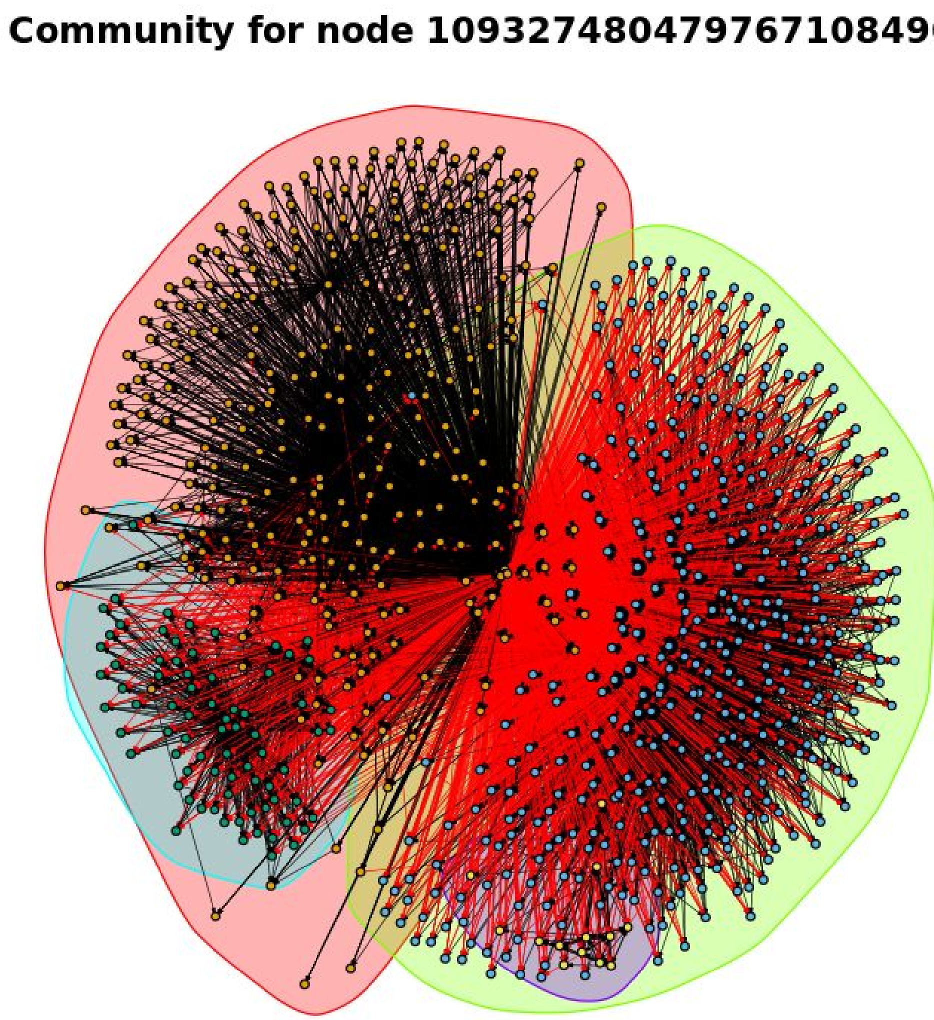

Figure 2.4 The Community Graph for Node 109327480479767108490

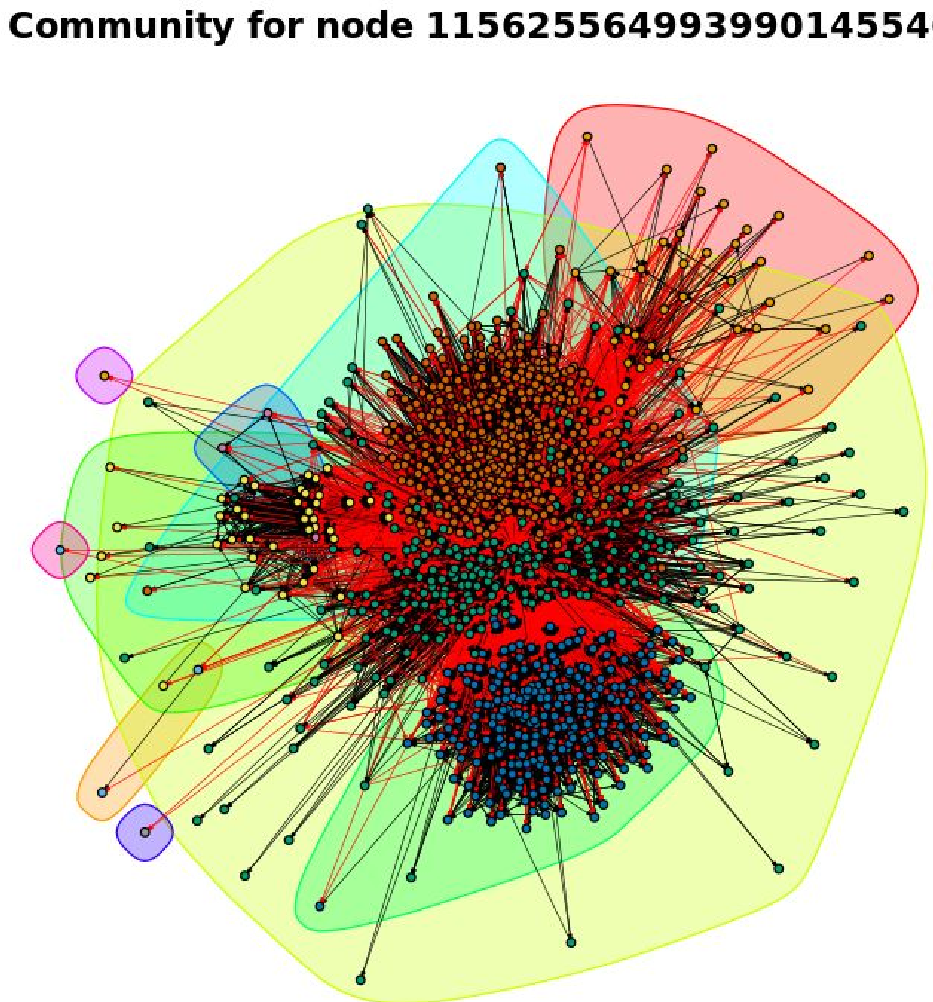

Figure 2.5 The Community Graph for Node 115625564993990145546

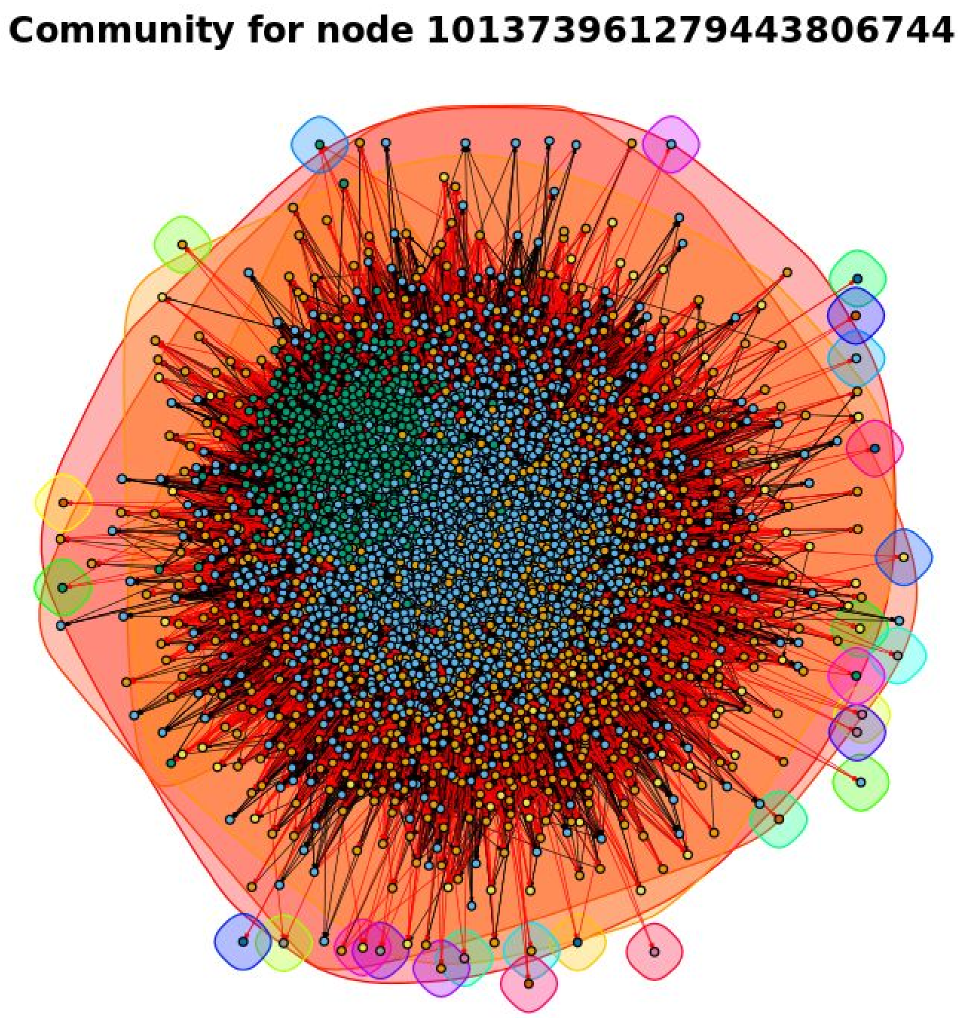

Figure 2.6 The Community Graph for Node 101373961279443806744

Question 21.

Homogeneity measures the score of people who are in a circle that’s assigned to the same community at the same time. Completeness measures the score of people who are assigned to the community but not in the same circle at the same time.

Question 22.

Here we calculated the homogeneity and completeness of the given three IDs. The results are shown as blow:

Node ID	109327480479767108490	115625564993990145546	101373961279443806744
Homogeneity	0.884913	0.7970808	-0.679265
Completeness	0.5088401	-0.1283379	-1.259542

Table 2.2 Homogeneity & Completeness of the three given nodes

From the table above, we could see that the first node has the highest homogeneity and positive completeness. Since there are fewer circles and communities in the first node than the other two and the relationship between circles and communities are almost one to one, as we see from the figures of question 20, the results show us higher homogeneity and completeness. For the second node, homogeneity is positive while the completeness is negative. We could see from the figures in question 20, that there are many circles across different communities which decreases completeness and even makes completeness negative. But in the same community, there is almost one circle or very few different circles assigned to it, which shows us relative high homogeneity. And for the third node, the homogeneity and completeness are both negative. As we could see from the figure 2.6, there are many circles across different communities as well as many different circles within the same community, which decreases both value of homogeneity and completeness.