ECE 232E Project 2
Social Network Mining

Chenchen Kuai	206074833
Hao Wang	405629183
Yuning Yang	705930008

1. Facebook network

1. Structural properties of the Facebook network

QUESTION 1: A first look at the network

QUESTION 1.1: Report the number of nodes and number of edges of the Facebook network.

Answer: Based on the code shows that Facebook network number of nodes: 4039;number of edges: 88234

QUESTION 1.2: Is the Facebook network connected? If not, find the giant connected component (GCC) of the network and report the size of the GCC.

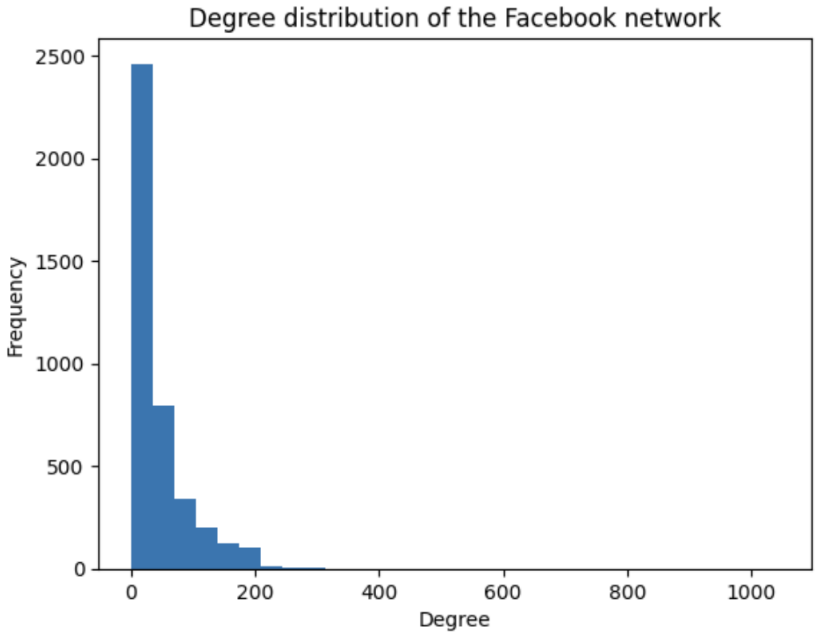
Answer: Yes, by the function is.connected, we know that the Facebook network is connected.

QUESTION 2: Find the diameter of the network. If the network is not connected, then find the diameter of the GCC.

Answer: The diameter is 8.

QUESTION 3: Plot the degree distribution of the facebook network and report the average degree.

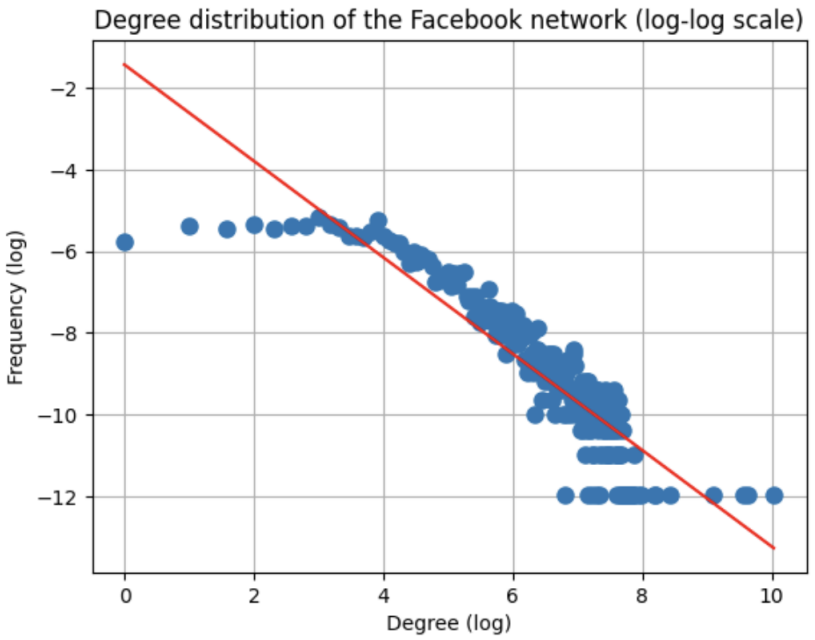
Answer: The degree distribution of the facebook network is shown below. The average degree is 43.691013.




```
mean=43.691013
```

QUESTION 4: 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.

Answer: The plot about degree distribution of Question 3 in a log-log scale is shown below, and the slope is -1.180.




```
Slope: -1.180
```

2. Personalized network

QUESTION 5: Create a personalized network of the user whose ID is 1. How many nodes and edges does this personalized network have? Hint Useful function(s): makeegograph

Answer: This personalized network have number of nodes: 348;number of edges: 2866

QUESTION 6: What is the diameter of the personalized network? Please state a trivial upper and lower bound for the diameter of the personalized network.

Answer: The diameter of the personalized network is 2, and the trivial upper bound is 2 and lower bound is 1.

QUESTION 7: 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)?

Answer: In the context of the personalized network, the diameter represents the longest shortest path between any two nodes in the network. In this case, the personalized network is

an ego network centered around a user and their neighbors. If the diameter of the personalized network = 2(upper bound), it means that all neighbors of the central user are only directly connected to the central user and not to each other. In other words, the network is not fully connected, and any two neighbors need to pass through the central user to connect, making the personalized network a star-like structure. If the diameter of the personalized network =1(lower bound), assuming there are more than 3 nodes in the personalized network, it means that all neighbors are directly connected to each other as well as to the central user. In this case, the personalized network forms a clique , where any two nodes are directly connected by an edge, and the central user is highly integrated within their network.

3. Core node's personalized network

QUESTION 8: How many core nodes are there in the Facebook network. What is the average degree of the core nodes?

Answer: There are 41 core nodes in the Facebook network, and the average degree of the core nodes is 277.439024.

QUESTION 9: For each of the above core node's personalized network, find the community structure using Fast-Greedy, Edge-Betweenness, and Infomap community detection algorithms. Compare the modularity scores of the algorithms. For visualization purpose, display the community structure of the core node's personalized networks using colors. Nodes belonging to the same community should have the same color and nodes belonging to different communities should have different color. In this question, you should have 15 plots in total. Hint Useful function(s): clusterfastgreedy , clusteredgebetweenness , clusterinfomap

Answer:

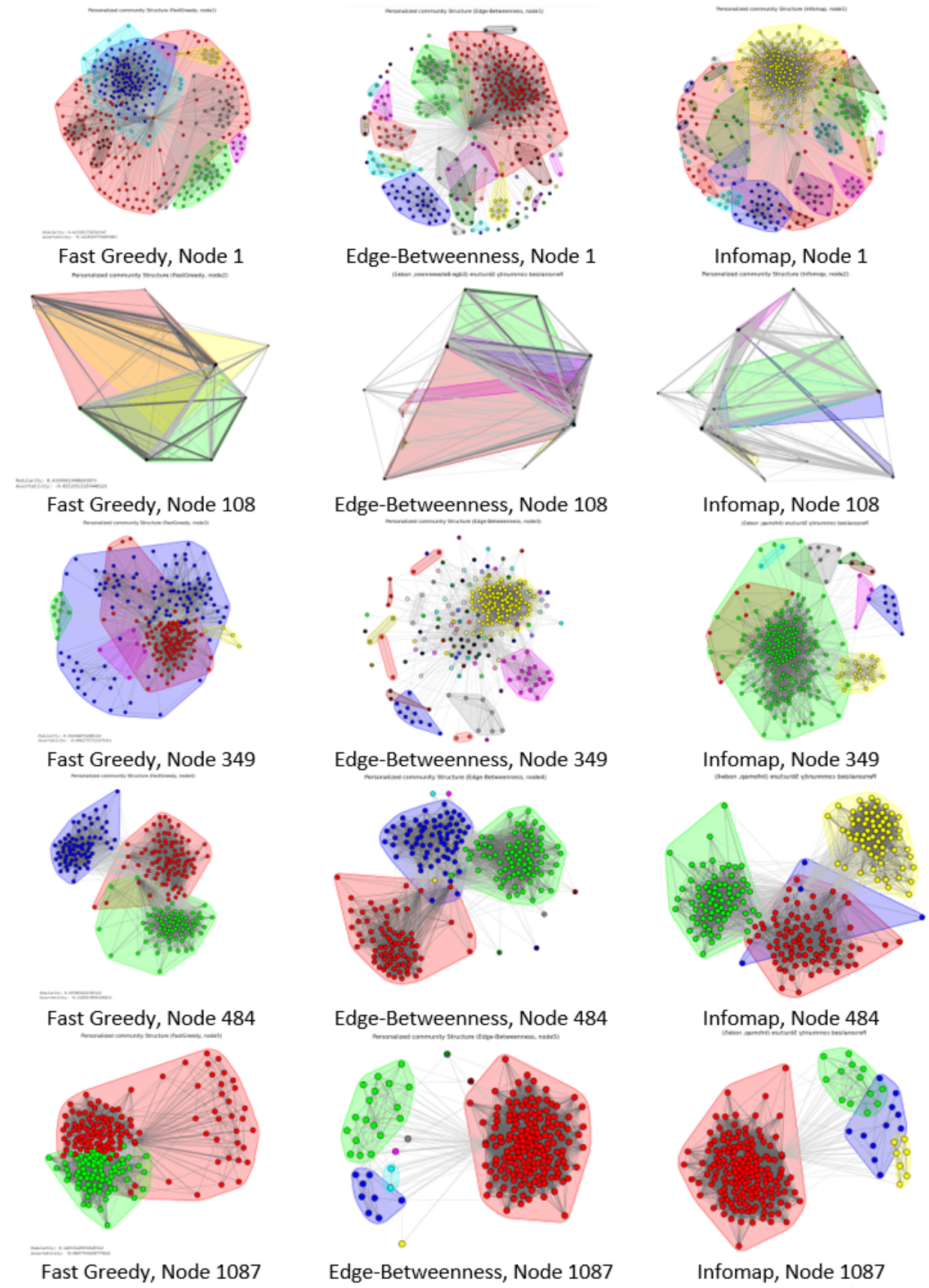
1. Fast-Greedy Algorithm works in a hierarchical manner to optimize modularity. Modularity is a measure of how well a network is partitioned into communities, with higher values indicating better partitioning.

2. The Edge-Betweenness algorithm is based on the betweenness centrality of network edges. Betweenness centrality is a measure of the importance of an edge in terms of the number of shortest paths between nodes that pass through it.

3. Infomap Algorithm:

The Infomap algorithm is based on the idea of compressing information about random walks on a network. It uses the concept of the Minimum Description Length (MDL) principle to find the optimal community structure.

Based on the three algorithms, the plots of personalized networks are generated.



Personalized community structure with different network structures

Modularity scores of 3 community detection algorithms (with core node)

Node ID	*Fast Greedy*	*Edge-Betweenness*	*Infomap*
1	*0.413*	*0.353*	*0.389*
108	*0.436*	*0.507*	*0.508*
349	*0.252*	*0.134*	*0.095*
484	*0.507*	*0.489*	*0.515*
1087	*0.146*	*0.028*	*0.027*

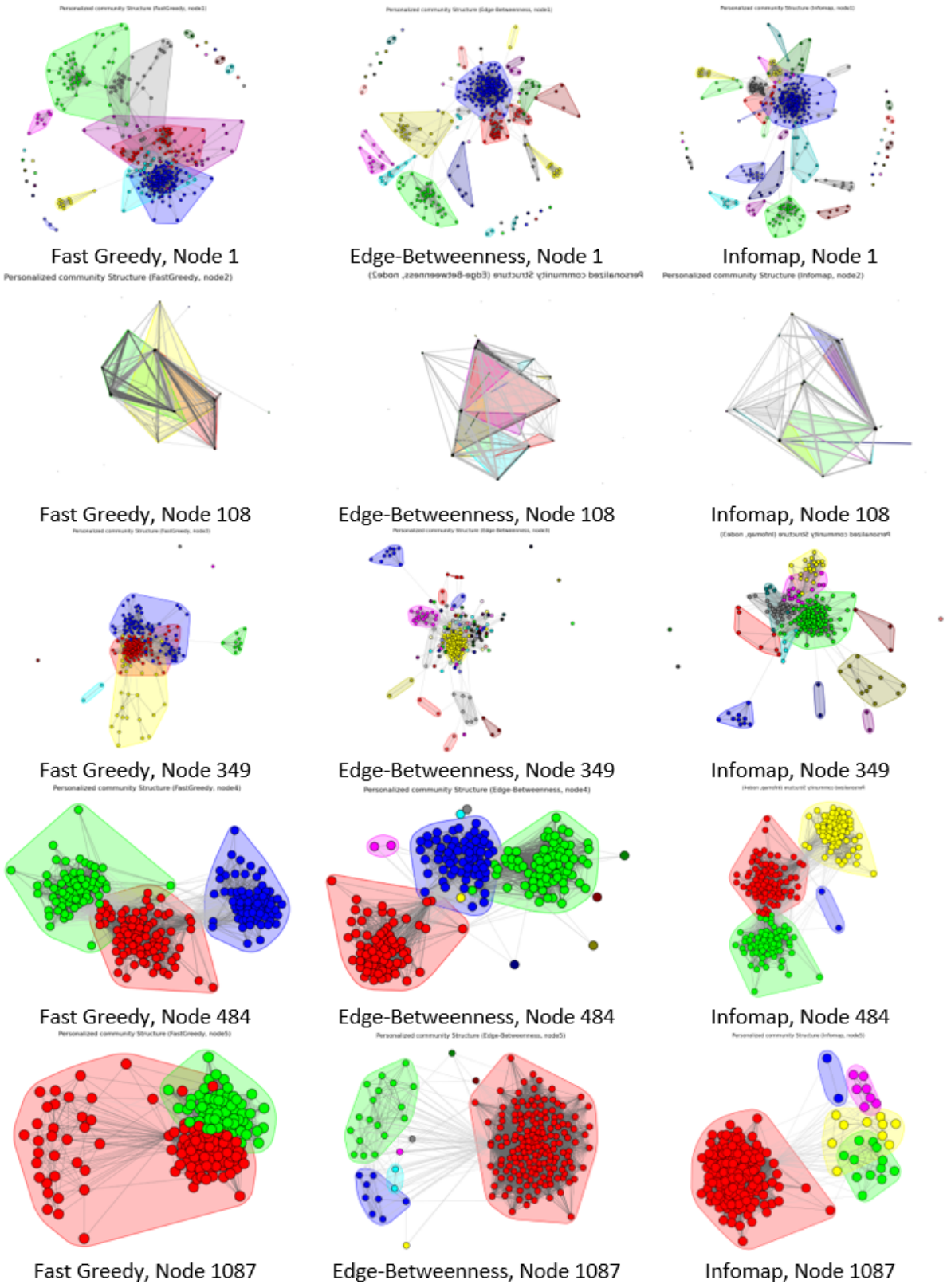
For community detection algorithms, Fast-Greedy generally performs better in terms of modularity compared to Edge-Betweenness and Infomap. This could be due to that this algorithm is designed to find the higher modularity in each step. However, for Node ID 108, which has the largest number of nodes in the personalized network, Edge-Betweenness and Infomap have higher modularity scores than Fast-Greedy. This suggests that Fast-Greedy may be more suitable for smaller networks, while Edge-Betweenness and Infomap are better suited for larger networks. Additionally, Fast-Greedy and Infomap algorithms are faster than the Edge-Betweenness algorithm.

In the context of core nodes' personalized networks, Node ID 484 achieves the highest modularity score among all nodes, indicating a clear community structure within its personalized network. In contrast, Node ID 1087 has the lowest modularity score, suggesting a more ambiguous community structure in its personalized network.

QUESTION 10: For each of the core node's personalized network (use same core nodes as Question 9), remove the core node from the personalized network and find the community structure of the modified personalized network. Use the same community detection algorithm as Question 9. Compare the modularity score of the community structure of the modified personalized network with the modularity score of the community structure of the personalized network of Question 9. For visualization purpose, display the community structure of the modified personalized network using colors. In this question, you should have 15 plots in total.

Answer:

In the case that the core node is removed in the personalized networks, the community structures are shown in Figure.



Modified personalized community structure with different network structures

Modularity scores of 3 community detection algorithms

Node ID	Core node	Fast Greedy	Edge-Betweenness	Infomap
1	With	0.413	0.353	0.389
	Without	0.442	0.416	0.418
108	With	0.436	0.507	0.508
	Without	0.458	0.521	0.521
349	With	0.252	0.134	0.095
	Without	0.246	0.151	0.245
484	With	0.507	0.489	0.515
	Without	0.534	0.515	0.543
1087	With	0.146	0.028	0.027
	Without	0.148	0.032	0.027

According to the table, the modularity increases without the core node, compared to that with the core node. This could be due to the following reasons:

Reduction in inter-community connections: A core node often serves as a bridge or hub between different communities, meaning it has many connections to nodes in different communities. When removing the core node, it can lead to a clearer separation of communities within the network. This increased separation will result in higher modularity.

Changes in intra-community connections: When the core node is removed, its direct connections with other nodes within its own community are also removed. This might lead to a reorganization of connections within the community. In some cases, this reorganization can result in tighter connections within communities, contributing to an increase in modularity.

QUESTION 11: 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.

Answer:

In the personalized network of the core node, the mutual friends of the node are the nodes that have a direct edge to this node, except the core node. Thus, the embeddedness can be calculated as the degree of the node in the personalized network of the core node minus one.

The expression is as follows:

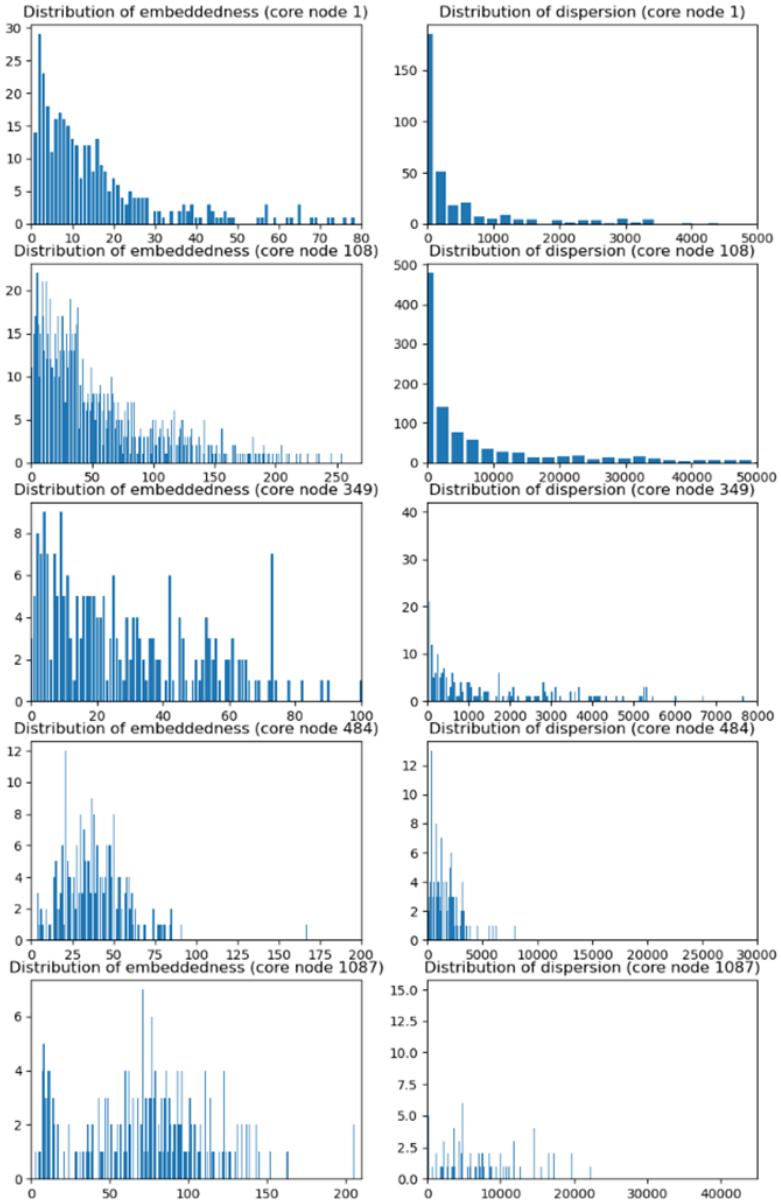
$$E(v\_i,v\_core\ )=degree(v\_i\ )-1$$

QUESTION 12: 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.

Answer:

The embeddedness follows the expression in question 11.

The dispersion is calculated in the network where the node and the core node are deleted in the personalized network of the core node. Dispersion of a node is defined as the sum of distances between every pair of the mutual friends the node shares with the core node.

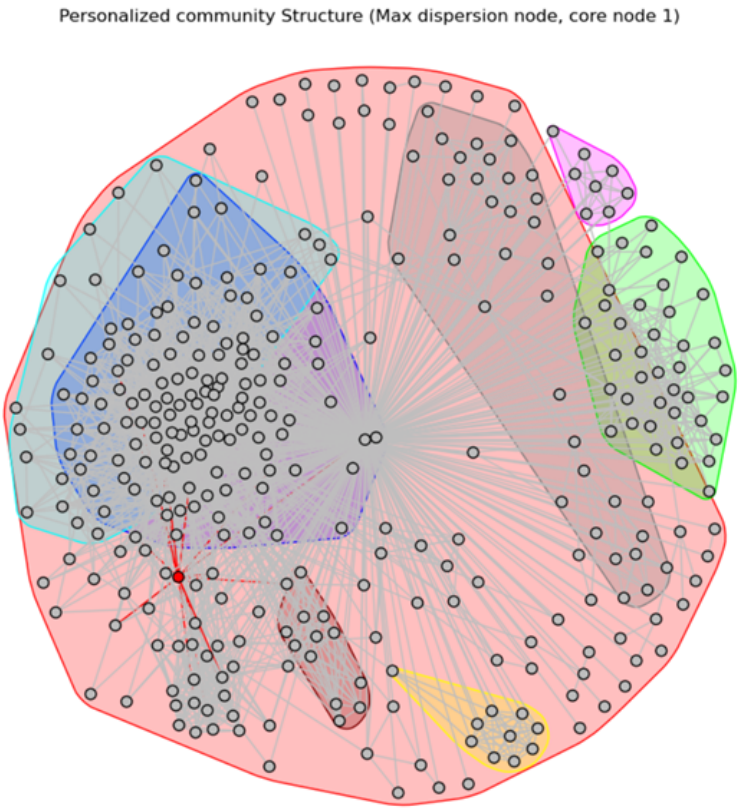


In the figure, we observe that the embeddedness of nodes in the personalized networks of core nodes 108 and 1087 is considerably higher than that of the other three, with a maximum of over 200 common friends. This suggests that the connections between nodes in these two networks are stronger (i.e., they have more robust ties), or that these networks contain larger friend clusters that correspond to well-defined areas of interaction in their lives, such as colleagues or classmates. Additionally, we notice that the dispersion of nodes in the personalized networks of core nodes 108, 484, and 1087 is significantly greater than the remaining two, with a maximum sum of distance exceeding 30,000. This indicates that nodes within these three networks possess broader social circles, and their shared acquaintances are more widely distributed.

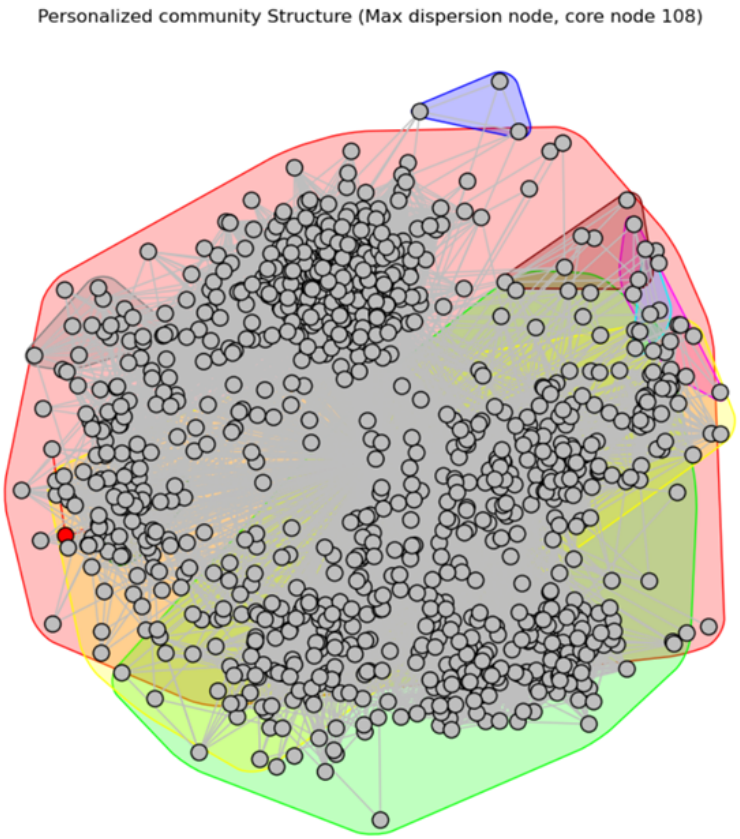
QUESTION 13: For each of the core node's personalized network, plot the community structure of the personalized network using colors and highlight the node with maximum dispersion. Also, highlight the edges incident to this node.

Answer:

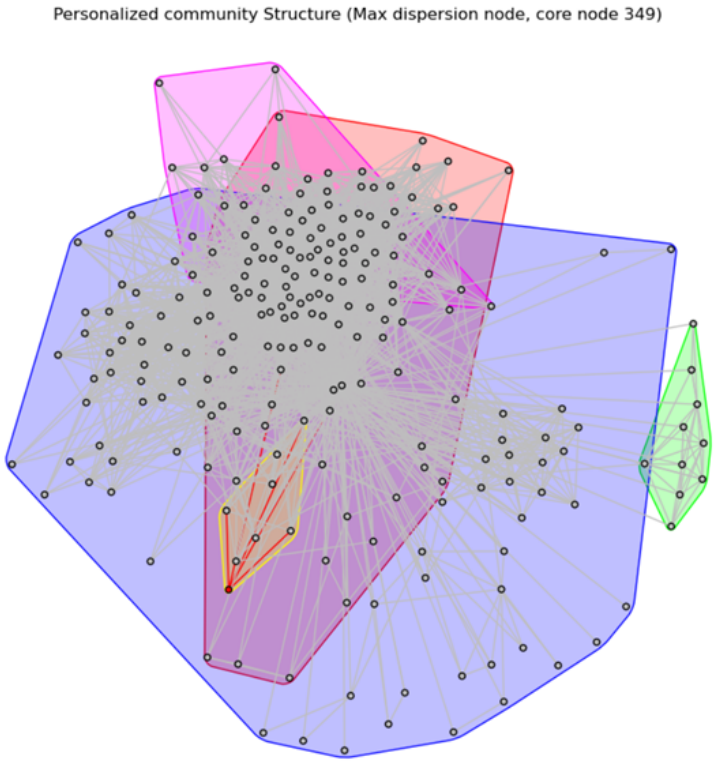
Below are community structures of the five core nodes' personalized networks with maximum dispersion node and the incident edges highlighted: (the node with maximum dispersion and its edges are highlighted in red)



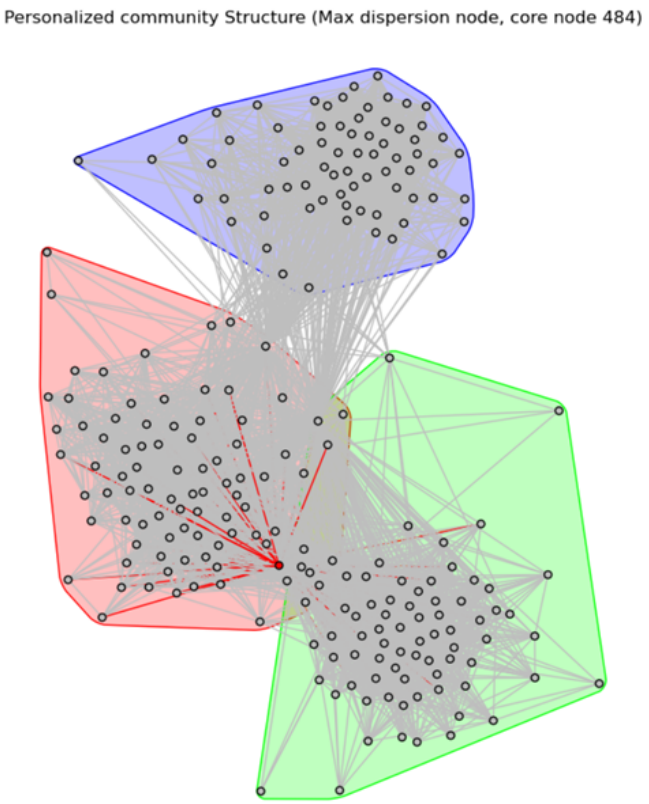
Community structure (core node 1, max dispersion shown)



Community structure (core node 108, max dispersion shown)



Community structure (core node 349, max dispersion shown)



Community structure (core node 484, max dispersion shown)

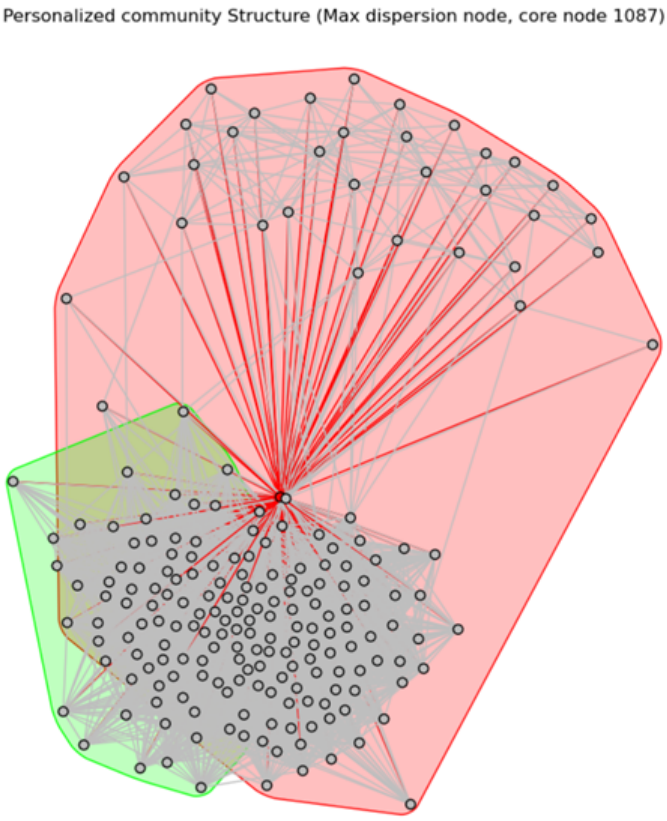
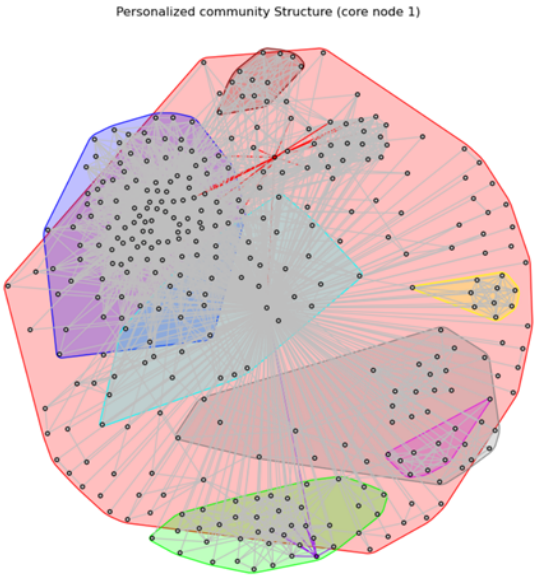


Figure 8. Community structure (core node 1087, max dispersion shown)

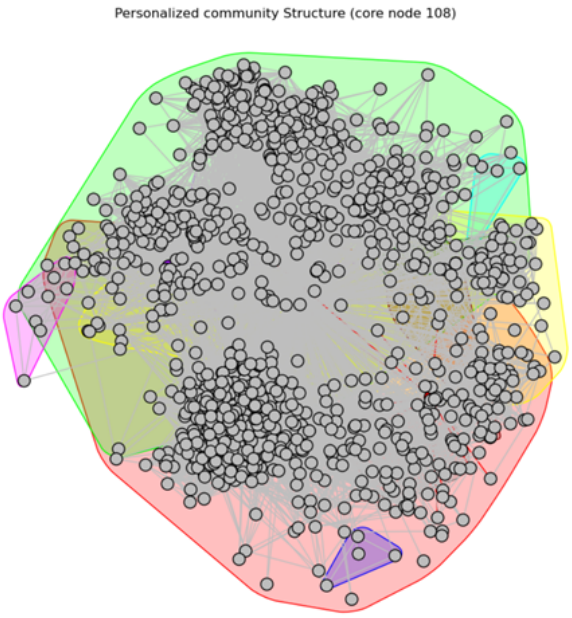
QUESTION 14: Repeat Question 13, but now highlight the node with maximum embeddedness and the node with maximum dispersion/embeddedness (excluding the nodes having zero embeddedness if there are any). Also, highlight the edges incident to these nodes. Report the id of those nodes.

Answer:

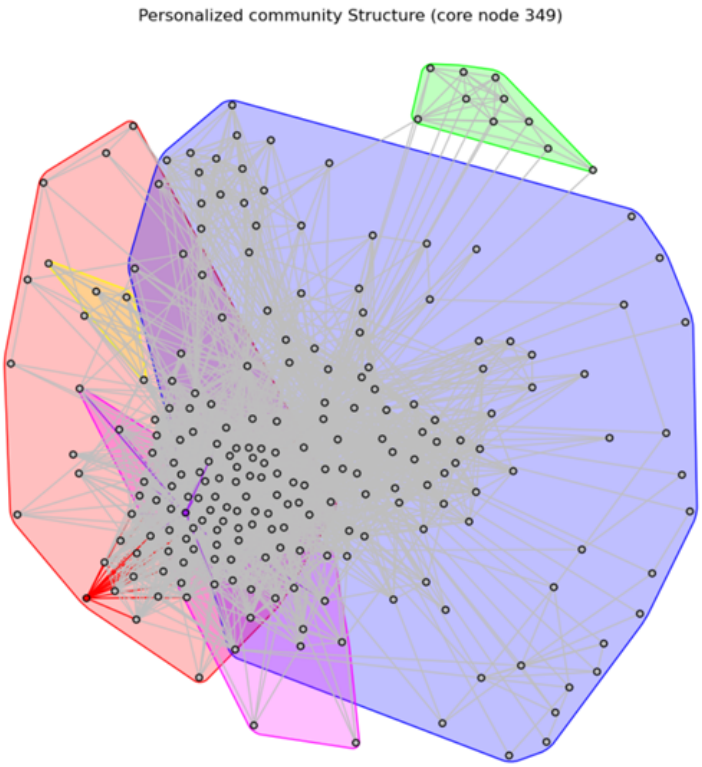
Below are community structures of the five core nodes' personalized networks with maximum dispersion node and the incident edges highlighted: (the node with maximum embeddedness and its edges are highlighted in red, the node with maximum dispersion/embeddedness and its edges are highlighted in red)



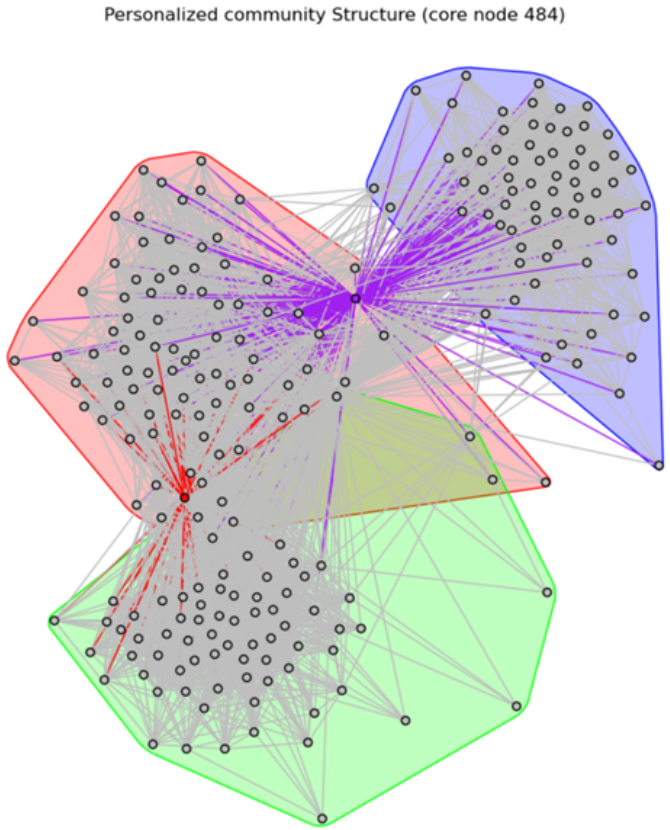
Community structure (core node 1, max embeddedness, dispersion/embeddedness)



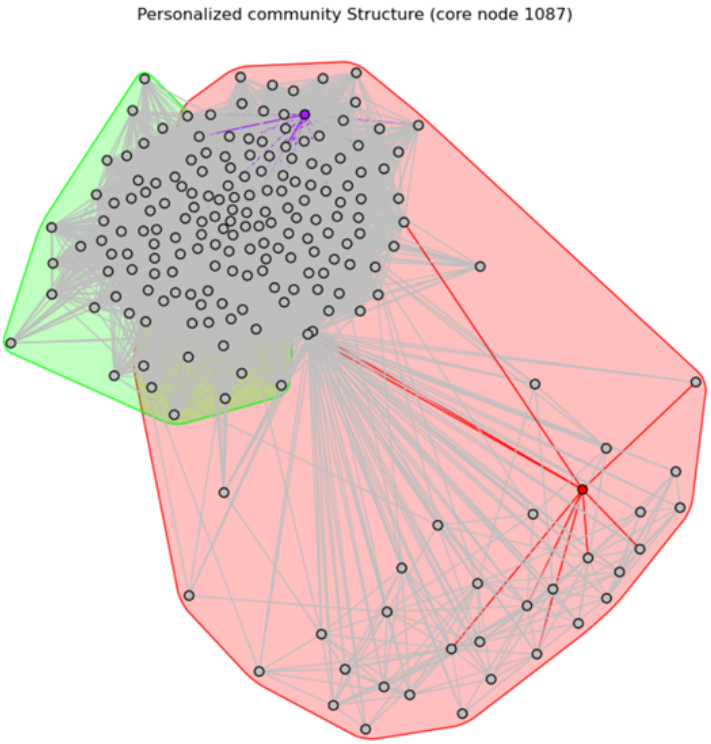
Community structure (core node 108, max embeddedness, dispersion/embeddedness)



Community structure (core node 349, max embeddedness, dispersion/embeddedness)



Community structure (core node 484, max embeddedness, dispersion/embeddedness)



Community structure (core node 1087, max embeddedness, dispersion/embeddedness)

QUESTION 15: Use the plots from Question 13 and 14 to explain the characteristics of a node revealed by each of this measure.

Answer:

For the node with maximum dispersion:

Nodes with maximum dispersion have mutual friends that are not well-connected or are from different circles, suggesting that the core node and the target node may have mutual friends from diverse backgrounds or areas, leading to a high distance between them.

For the node with maximum embeddedness:

Nodes with maximum embeddedness share many common friends with the core node, indicating a close relationship or belonging to the same social circle. However, high embeddedness alone does not guarantee a close personal relationship, as it could also result from belonging to a well-connected group where everyone knows each other.

Node with maximum dispersion/embeddedness:

The normalized value represents a balance between dispersion and embeddedness. Nodes with a high normalized value have a large dispersion value and a small embeddedness value, meaning they share fewer common friends with the core node, and those mutual friends are likely from different areas or not well-connected. This metric could be used to assess the

likelihood of two nodes being in a romantic relationship, with higher values indicating a higher probability of a potential romantic tie.

4. Friend recommendation in personalized networks

QUESTION 16: What is |Nr|, i.e. the length of the list Nr?

Answer:

The network is a personalized network of node ID 415. The vertices that have degree of 24 in this network are listed as:

[30, 52, 74, 89, 92, 101, 117, 132, 133, 135, 136]

Thus |N_r| = 11.

QUESTION 17: Compute the average accuracy of the friend recommendation algorithm that uses:

• Common Neighbors measure

• Jaccard measure

• Adamic Adar measure

Based on the average accuracy values, which friend recommendation algorithm is the best?

Answer:

The graph applies the personalized network of node 415. Edges of a node are deleted with a probability of 0.25, and the new network is tested if it can restore the deleted nodes to calculate the accuracy of three measures. The test result for a single node is shown as follows:

```
Number of edges deleted: 6
Deleted edges (using vertex IDs): [15, 73, 101, 124, 135, 155]
Nodes with the top R highest mutual friend counts (not neighbors) with node 74: [15, 73, 101, 124, 135, 155]
Nodes with the top R highest Jaccard (not neighbors) with node 74: [73, 101, 121, 124, 135, 155]
Nodes with the top R highest AAM (not neighbors) with node 74: [15, 73, 101, 124, 135, 155]
Accurancy of common neighbours:1.0
Accurancy of Jaccard measure:0.8333333333333334
Accurancy of AAM measure:1.0
```

Also, the result indicating the accuracy of the three measures is shown in the table.

Average accuracy of three measures

Common Neighbors Measure	0.85393
Jaccard Measure	0.79800
Adamic Adar Measure	0.83982

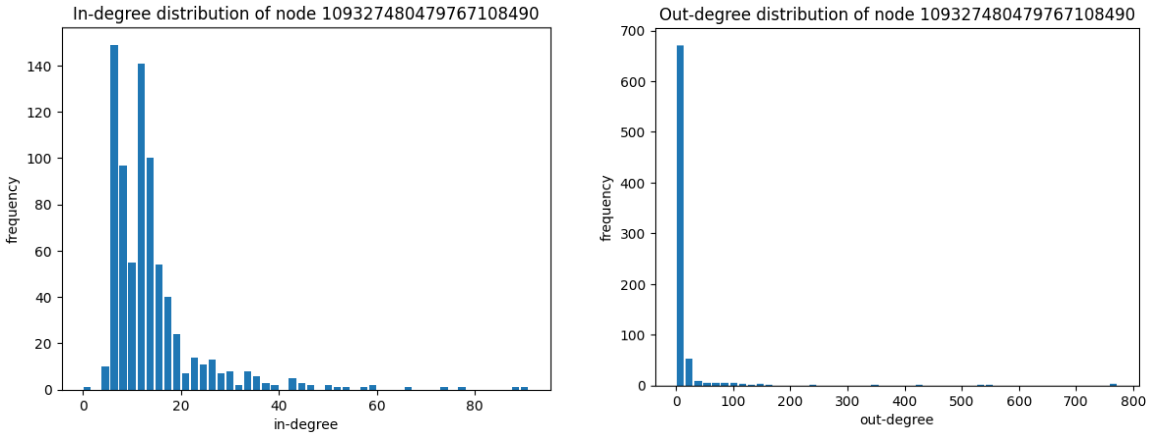
According to the result, the average accuracy of Common Neighbors Measure has the highest accuracy and the performance of Adamic Adar Measure is close to Common Neighbors Measure. Meanwhile, the performance of Jaccard Measure is the worst.

QUESTION 18: How many personal networks are there?

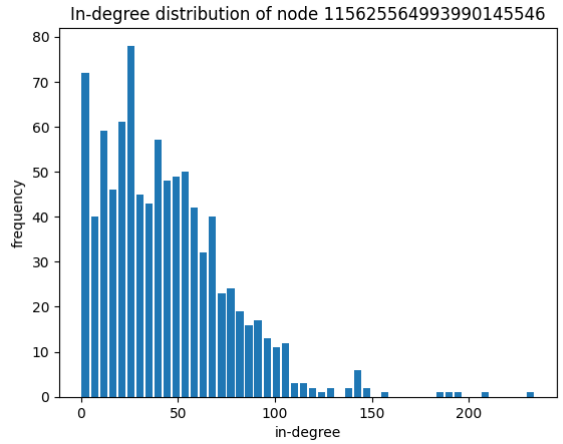
Answer: For this question we go over all the circles and find the circles with more than 2 people (2 lines) and find a total of 57 personal networks.

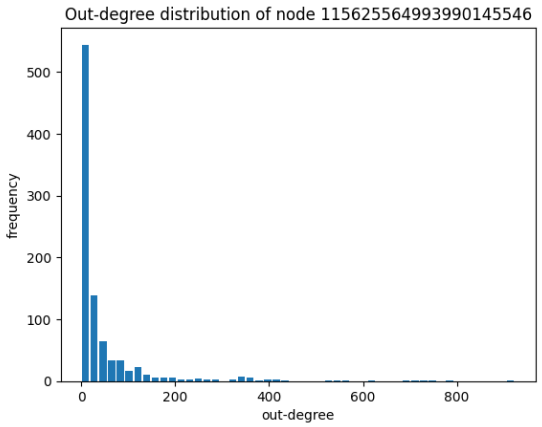
QUESTION 19: For the 3 personal networks (node ID given below), plot the in-degree and out-degree distribution of these personal networks. Do the personal networks have a similar in and out degree distribution? In this question, you should have 6 plots.

Answer: The following figure are the in and out degree distribution of node 109327480479767108490

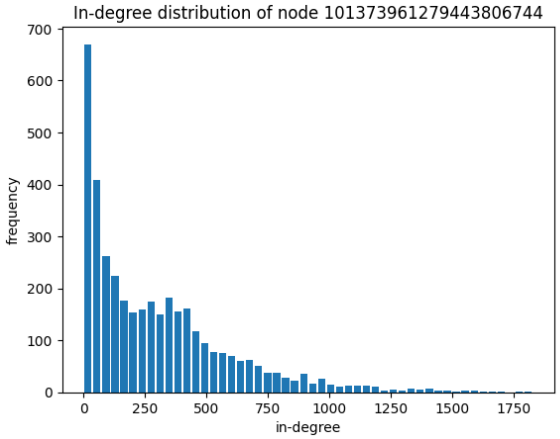


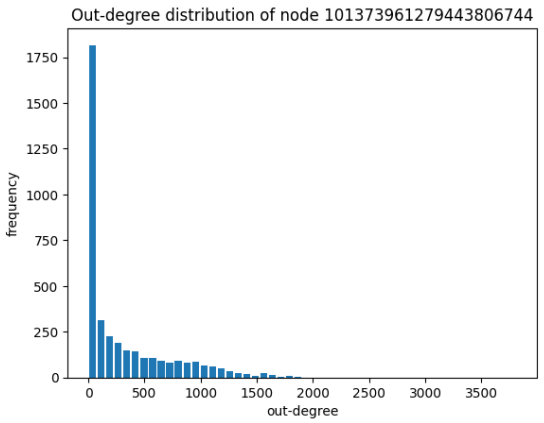
Here are the figures of node 115625564993990145546





Here are the figures of node 10137396127944380674 4





From the figures above, we can see that they have similar in and out degree distribution. The first network has the smallest in-degree and out-degree while the third network has the biggest in and out degree. which means that the first network is simplest and the third network is the most complex.

We can calculate the mean and the variance of the indegree and outdegree and the result is as follow:

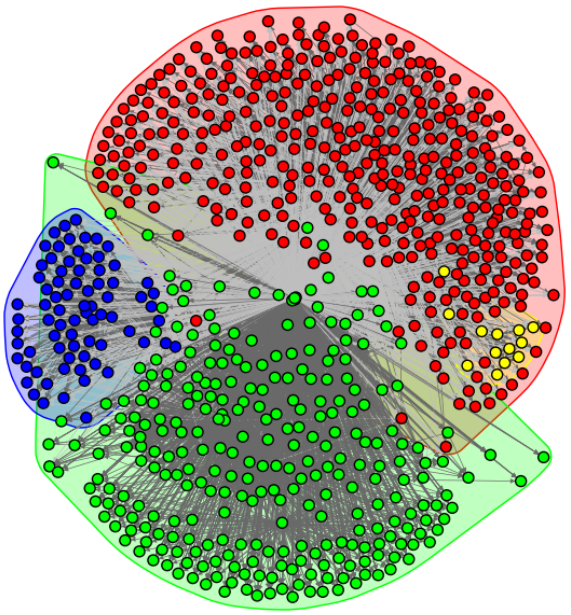
	mean(in)	var(in)	mean(out)	var(out)
109327480479767108490	14.062	95.877	14.062	4582.249
115625564993990145546	43.640	1019.516	43.640	9341.183
10137396127944380674 4	298.118	86386.120	298.118	166143.175

We can see that the graph3 has far more degrees than 1 and 2, because graph 3 is larger and more complicated. We can see that the variance of the out degree is always bigger than in degree. Because out degree means the node follow other node, and the central node follows

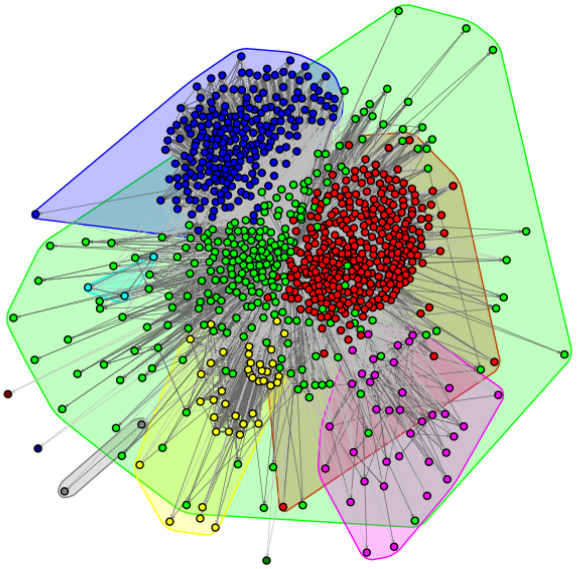
each node in this graph (by defintion). So there is an outstanding number of our degree distribution so that the variance is much larger.

QUESTION 20: 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.

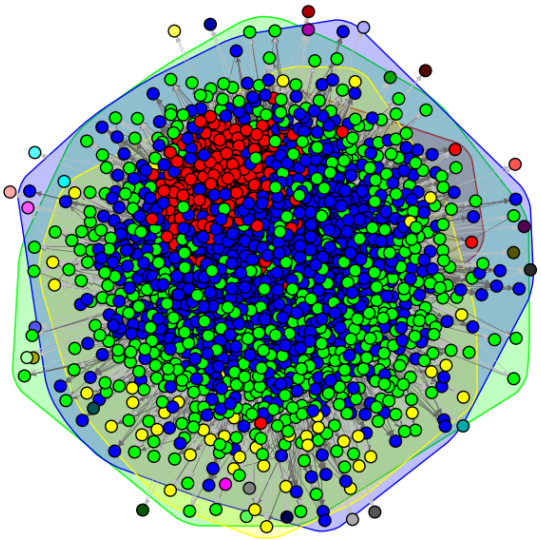
Answer: For network 109327480479767108490: modularity score is 0.25287, communities is as follows:



For network 115625564993990145546: modularity score is 0.31996, communities is as follows:



For network 101373961279443806744: modularity score is 0.19204, communities is as follows:



The modularity score ranges from -1 to 1, with higher values indicating better community structures with dense connections within communities and sparse connections between communities. For these three community, the modularity scores are relatively low (below 0.5), which suggests that the community structures are not particularly strong. However, these scores are positive, indicating that there are some community structures present in each personal network. The values are close to each other, indicating that the three personal networks have somewhat similar community structures, community 1 has the best community structure among them, but none of them have very distinct or strong community divisions.

QUESTION 21: Based on the expression for h and c, explain the meaning of homogeneity and completeness in words.

Answer: Homogeneity (H) and Completeness (C) are both derived by subtracting the ratio of conditional entropy to entropy.

Homogeneity (H) represents the reduction in uncertainty of the true class labels, given the clustering results. In other words, it measures how much knowing the cluster assignments helps in predicting the true class labels. A high homogeneity score indicates that each cluster is comprised mostly of data points belonging to a single class. Value 1 indicates that each cluster contains data points belonging to just one class.

Completeness (C) represents the reduction in uncertainty of the clustering results, given the true class labels. It measures how much knowing the true class labels helps in predicting the cluster assignments. A high completeness score indicates that all data points from the same class are grouped together in the same cluster. Value 1 indicates that all data points from the same class are grouped together in the same cluster.

QUESTION 22: 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?

	Homogeneity	Completeness
109327480479767108490	0.5352	0.5589
115625564993990145546	0.1482	0.3857
101373961279443806744	0.0008	0.0004

For 109327480479767108490, the relatively higher homogeneity and completeness scores means the clustering results are more consistent with the true class labels. The clusters are mostly composed of data points belonging to the same class (homogeneity) and most data points of the same class are grouped together in the same cluster (completeness).

For 115625564993990145546, it has a lower homogeneity score, which means the clusters are less pure in terms of class composition. However, the completeness score is moderate, indicating that the data points of the same class are somewhat grouped together in the same cluster.

For 101373961279443806744, it has very low homogeneity and completeness scores, suggesting poor clustering results. The clusters are not composed of data points belonging to the same class, and data points of the same class are not grouped together in the same cluster.

There are no negtative value because Both homogeneity and completeness are derived from conditional entropy and entropy measures. Since entropy is always non-negative and conditional entropy is always less than or equal to the entropy, the values of homogeneity and completeness will always be between 0 and 1.

QUESTION 23: Idea 1

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

Answer: In this question, we use mainly pytorch to build our network, and we use some of the useful functions in sklearn and scipy.

Firstly, we use the class provided code to get the feature matrix ***X*** with shape (2708, 1433) and the adjacency matrix ***A*** with shape (2708, 2708). The adjacency matrix is a sparse matrix with 10556 stored elements. (There were original 5429 citations so when build the undirected graph, each citation should make both A*ij* and A*ji* equals to one, which should get us 10858 elements in the adjacency matrix ***A***. However, there are 302 pairs of circulated citations in our dataset, such as paper 35 citate paper 210871 and paper 210871 also citate paper 35. So there are a total of 10556 elements in the adjacency matrix ***A***. We use the following code to find all the circulated citations.)

```
[22] edge_list2 = []
     for edge in edge_list:
         a = edge[0]
         b = edge[1]
         edge_list2.append((b, a))

     edge_list3 = edge_list + edge_list2
     repeat_items = [key for key,value in Counter(edge_list3).items()if value > 1]
     print(repeat_items[:5])
     print(len(repeat_items))
```

```
[('35', '210871'), ('130', '39403'), ('506', '89416'), ('910', '5462'), ('910', '5869')]
302
```

```
[20] ('35', '210871') in edge_list
```

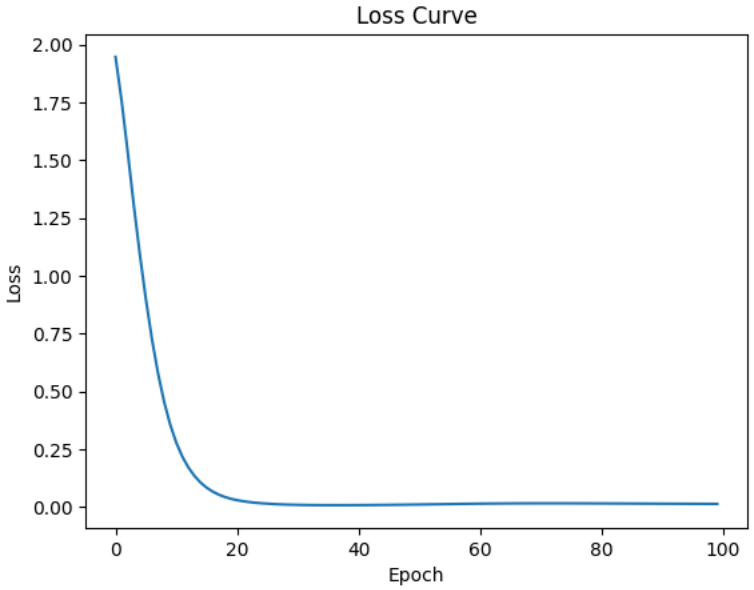
```
True
```

```
[21] ('210871', '35') in edge_list
```

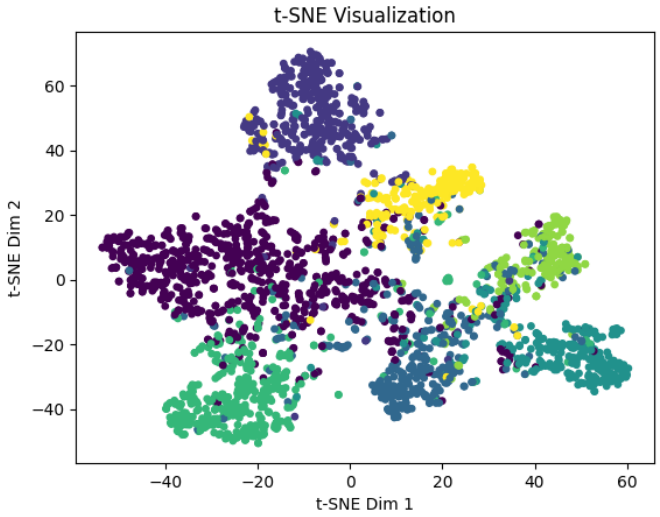
```
True
```

Then, we choose 20 items from each class and masked them as the training set. use torch to build the GCN network after changing the adjacency matrix ***A*** into edge indexes. The GCN network is a two-layer network with input dim 1433 and out dim 7 and a hidden dim 32. We use the CrossEntropyLoss as a common classification loss function, the Adam optimizer with learning rate 0.01 and weight decay 5e-4 to optimize the parameters. Specifically, due to the characteristic of the GCN network, all of the input feature should be calculated in the network at the same time. So we use the input mask to calculate loss cause we don't want the model to train on test data. And we use the test mask to calculate the test accuracy.

The following figure shows the loss curve of the training of GCN network. We can see that the network converged after 20 epochs of training.



the test accuracy is 0.8104 and the following figure shows the TSNE visualization of the seven classes of all the output (including the training data)



QUESTION 24: Idea 2

Extract structure-based node features using Node2Vec [2]. 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)?

Answer: Node2Vec is a graph embedding technique that learns a low-dimensional representation for each node in a graph based on the graph's topology. It uses a random walk strategy to sample sequences of nodes in the graph, and then applies a skip-gram model to learn node embeddings that preserve the network structure. Intuitively, the Node2Vec

embedding was generated by random walk, so it represent the structural information of a graph.

We use a two layer NN to build the classifier. The hidden dim is 32 and output dim is 7. We use the CrossEntropyLoss as a common classification loss function, the Adam optimizer with learning rate 0.01 and weight decay 5e-4 to optimize the parameters, the same as idea1. For Node2Vec only, we choose the embedding size as 32. For text feature, the embedding size is 1433, and we also tried combining the two features, which give us input dimension of 1465. The following table shows the result of each trial.

	structure_only	text_only	combine
test accuracy	0.6819	0.5448	0.7403

We can see that the structure information outperforms the text information. probably because our training set is small. The Node2Vec algorithm reduce dimensionality while preserving the dataset's graph structure. This can help alleviate the curse of dimensionality and overfitting problems. Using only text features to classify documents ignores the document's relationship with other documents and may not capture important contextual information.

Combining the text and structure information can get us best classification accuracy as 74.03%. Because we are leveraging both graph-based and content-based information, which can help improve classification accuracy and reduce the risk of overfitting.

QUESTION 25: Idea 3

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.

Answer: For this question, we don't use the helper code but write all the different situations in a single loop. For part (a), the typical random walk is used and the transition matrix is just the adjacency matrix. For part (b), we use sklearn.metrics.pairwise.cosine_similarity(X, X) to calculate the cosine similarity matrix, then calculate the exp of the matrix and multiply the matrix with A to disable the walk to a non-neighbor. We don't need to explicitly normalize the transition probability matrix since the random choice function can do it. As for teleportation, we use the following code to implement and set pout to zero to indicate no teleportation. There are a total of 140 seeds and walk 1000 times for each seeds and we get 140,000 results starting from 7 different classes. Then for each test sample we find which class is most likely to end on this case.

```
if random.random() < pout:
    next_node = random.choices(train_indexs[label])[0]  # 20 seeds
else:
    probs = transition_probs[current_node]
    next_node = random.choices(range(N), probs)[0]  # neighbors
current_node = next_node
```

The following table is the result of test accuracy and f1 score of the random walk classifier. We can see that with teleportation 0.1 the accuracy is always the highest. For teleportation 0.2, part (b) out performs part (a) which shows the text feature is useful.

	no teleportation	teleportation 0.1	teleportation 0.2
Part (a) accuracy	0.7251	0.7391	0.7259
Part (a) f-1	0.7263	0.7407	0.7273
Part (b) accuracy	0.7231	0.7352	0.7317
Part (b) f-1	0.7244	0.7373	0.7333