

Large-Scale Social and Complex Networks: Design and Algorithms
ECE 232E Summer 2018
Prof Vwani Roychowdhury
UCLA, Department of ECE

Project 2

Social Network Mining

Due on Monday, July 21, 2018 by 11:59 pm

Team Members

Jennifer MacDonald, UID: 604501712
Nguyen Nguyen, UID: 004870721
Sam Yang, UID: 604034791

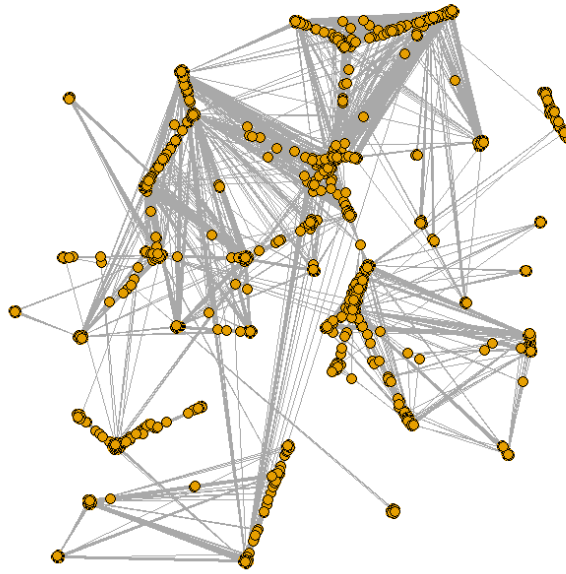
Introduction

The project is designed to explore various interesting properties and meaningful insights of a network's connectivity, degree distribution, as well as characteristics of the personalized network, and neighborhood based measurements. In this project, we explored the social networks of Facebook (undirected) and Google+ (directed).

1. Facebook network

We obtained our Facebook dataset from <http://snap.stanford.edu/data/egonets-Facebook.html> and unzipped the edgelist file facebook_combined.txt.gz and created the Facebook network. The dataset consists a list of connected pairs of nodes, which we used to generate a graph as seen in Figure 1 below. To create the graph, we used the data to assemble an edgelist in igraph library, and then created an undirected graph from the edgelist.

Figure 1: The Facebook network



1. Structural properties of the facebook network

First, we examine two structural properties of the facebook network: connectivity, and degree distribution. Connectivity is a boolean such that it is true if all the nodes of the network are connected. Degree distribution is the measurement of the distribution of the number of neighbors that each node has.

QUESTION 1: Is the facebook network connected? If not, find the giant connected component (GCC) of the network and report the size of the GCC.

Basing on our generated graph above, we found that our facebook network is connected.

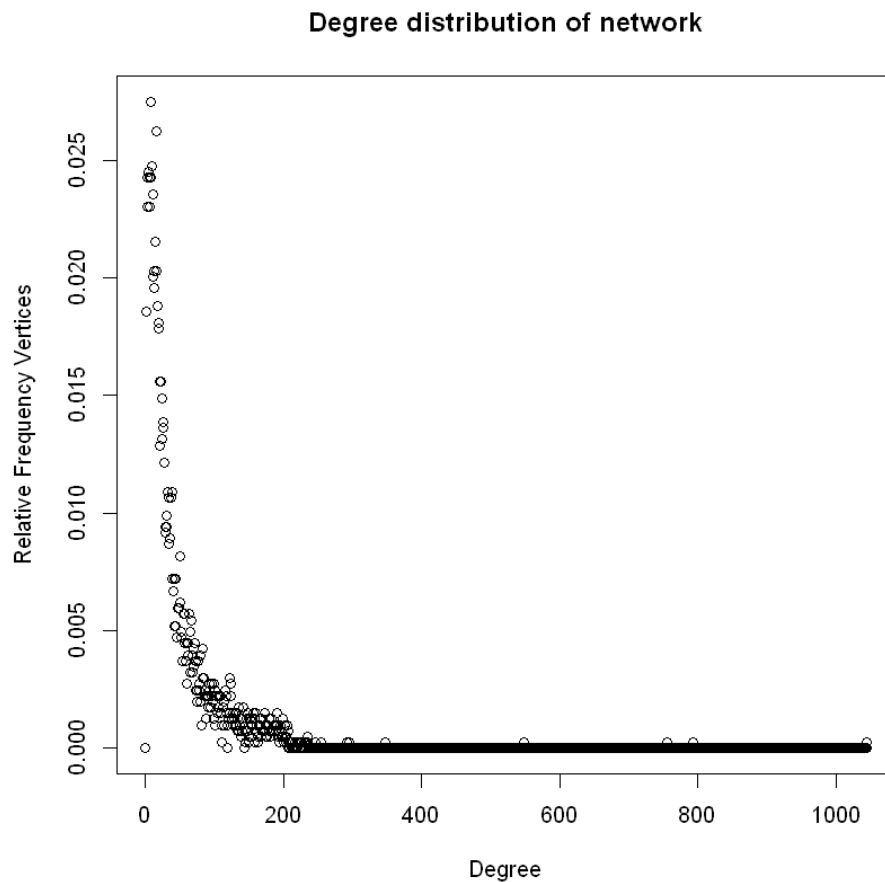
QUESTION 2: Find the diameter of the network. If the network is not connected, then find the diameter of the GCC.

We measured the diameter of the network and found that the diameter of the network is 8.

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

We plotted the degree distribution, seen below in Figure 2.

Figure 2: Degree distribution of the Facebook network



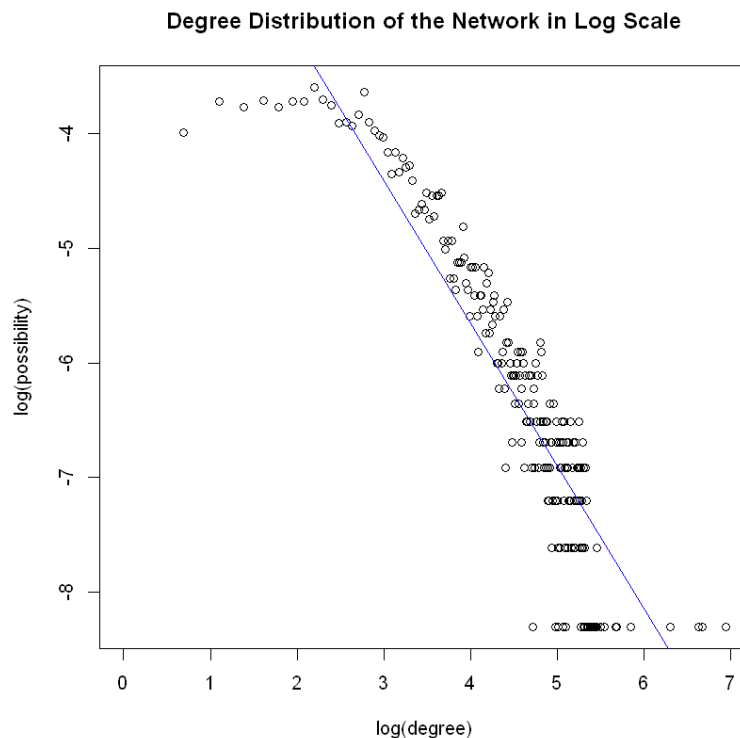
The plot shows that the nodes with a number of degrees close to 1 is comparatively high, with a steep drop off for the degrees less than 100, and then it levels out around 200 where the relative frequency vertex value is close to 0. This means that all our users are connected somehow (since graph is all connected), but most of the users are only aware of their direct friends or possibly at most some of their friends' friends.

The average network degree is 522.5, though this is not a good measurement as seen that the plot is skewed to the left.

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.

The degree distribution of the network in log-log scale can be seen in Figure 3. The best-fit line of the data was added in blue. In order to create the line, we removed all degree and degree distribution pairs with non-real numbers and used the remaining data create a best-fit line. The best-fit line appears to have a slope of roughly -1.3.

Figure 3: Degree distribution of the network in log-log scale



2. Personalized network

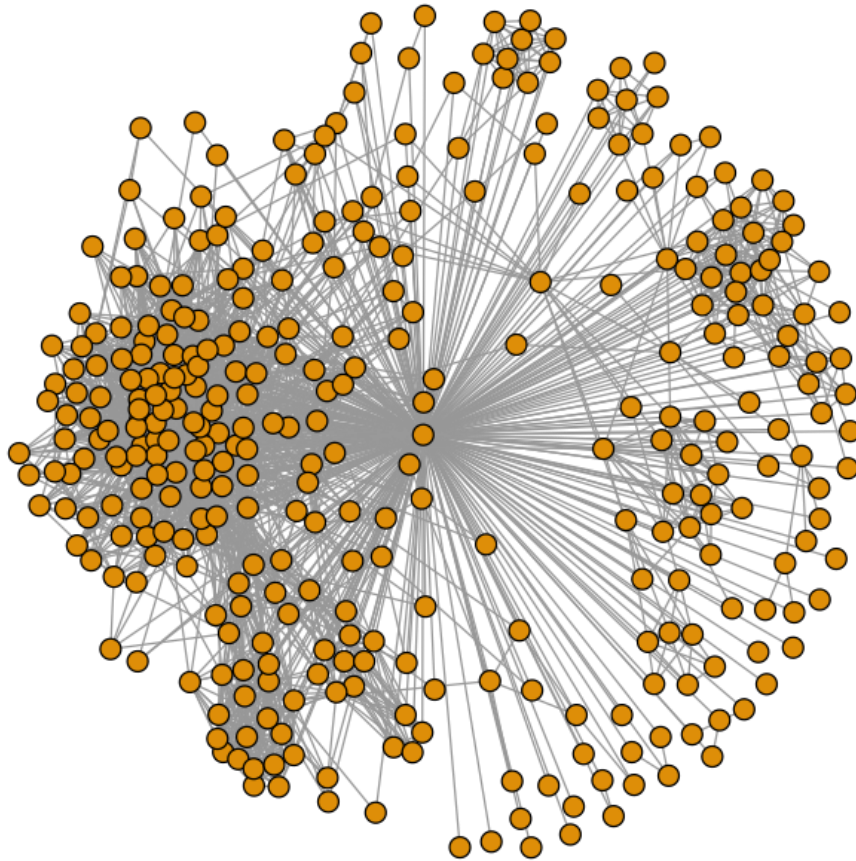
In this part of the analysis, we studied some of the structural properties of the personalized network of the users. First, we defined a personalized network to be a subgraph of the network created from a chosen user and their neighbors (or “friends” in Facebook terms). The ID’s we referenced are that of the graph node ID from our original network. Here, we are referring to a node ID such that the ID is 1 + node ID in edgelist.

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): `make_ego_graph`

We created the personalized network of the user whose ID is 1 by using the built-in function `make_ego_graph`. The network is shown below in Figure 4.

Figure 4: Personalized network of user with ID 1



We can see the relationships of the user whose ID is 1 in the center and its connection to all the other nodes. This personalized network has 18 nodes and 74 edges.

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.

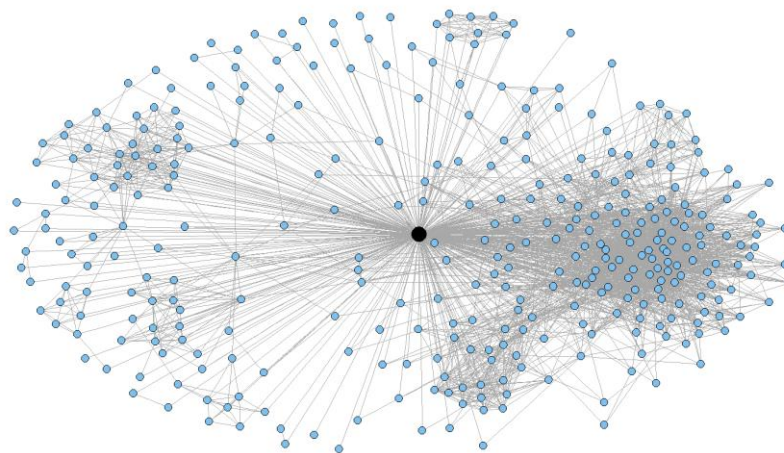
The diameter of the personalized network is 2. A trivial upper bound for the diameter of the personalized network is 2 and a trivial lower bound for the diameter of the personalized network is 1.

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?

We know the diameter to be the “longest shortest path” between any two vertices. So, a trivial upper bound for the diameter would have to be two since the shortest longest path would be from a friend to a friend, and would only require the original person to be the third node, making a diameter of 2. The trivial lower bound for the diameter would simply be the diameter from the original user to the friend, making a diameter of 1. This would happen if the original user had only one friend.

3. Core node's personalized network

We also defined a core node as a node that has more than 200 neighbors. An example of this can be seen below. In this part of our analysis, we looked at the properties that core nodes have in this next section.



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

When we counted the number of neighborhoods where the number of nodes were more than 200, we found that the number of core nodes in the Facebook network is 40. The average degree of those core nodes is 279.375.

3.1. Community structure of core node's personalized network

We also looked at the community structure of the personalized five network of the following core nodes:

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

Qualitatively, a community is defined as a subset of nodes within the graph such that connections between nodes are denser than connections with the rest of the network.

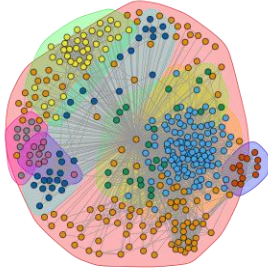
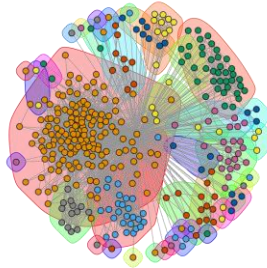
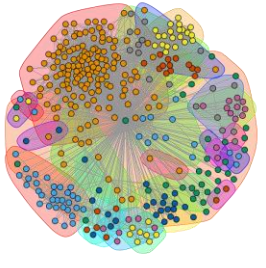
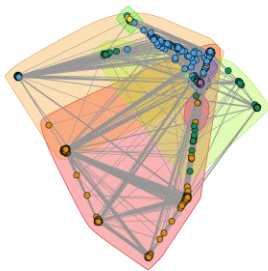
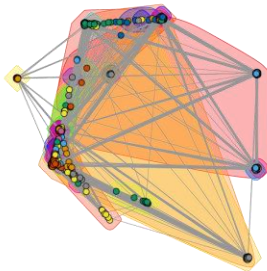
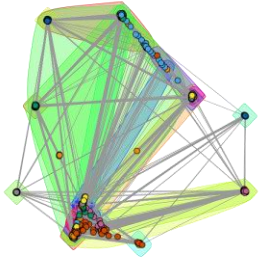
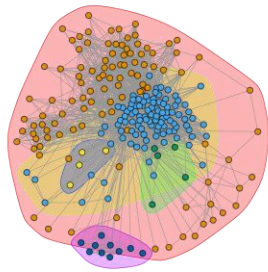
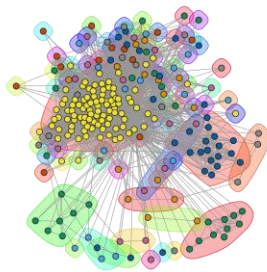
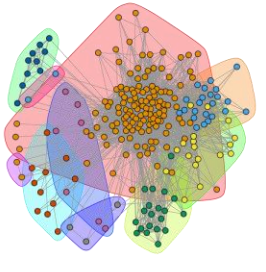
QUESTION 9: For each of the above core node's personalized network, find the community structure using Fast-Greedy, Edge-Betweenness, and Infomap community detection algorithms. Compare the modularity scores of the algorithms. For visualization purpose, display the community structure of the core node's 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.

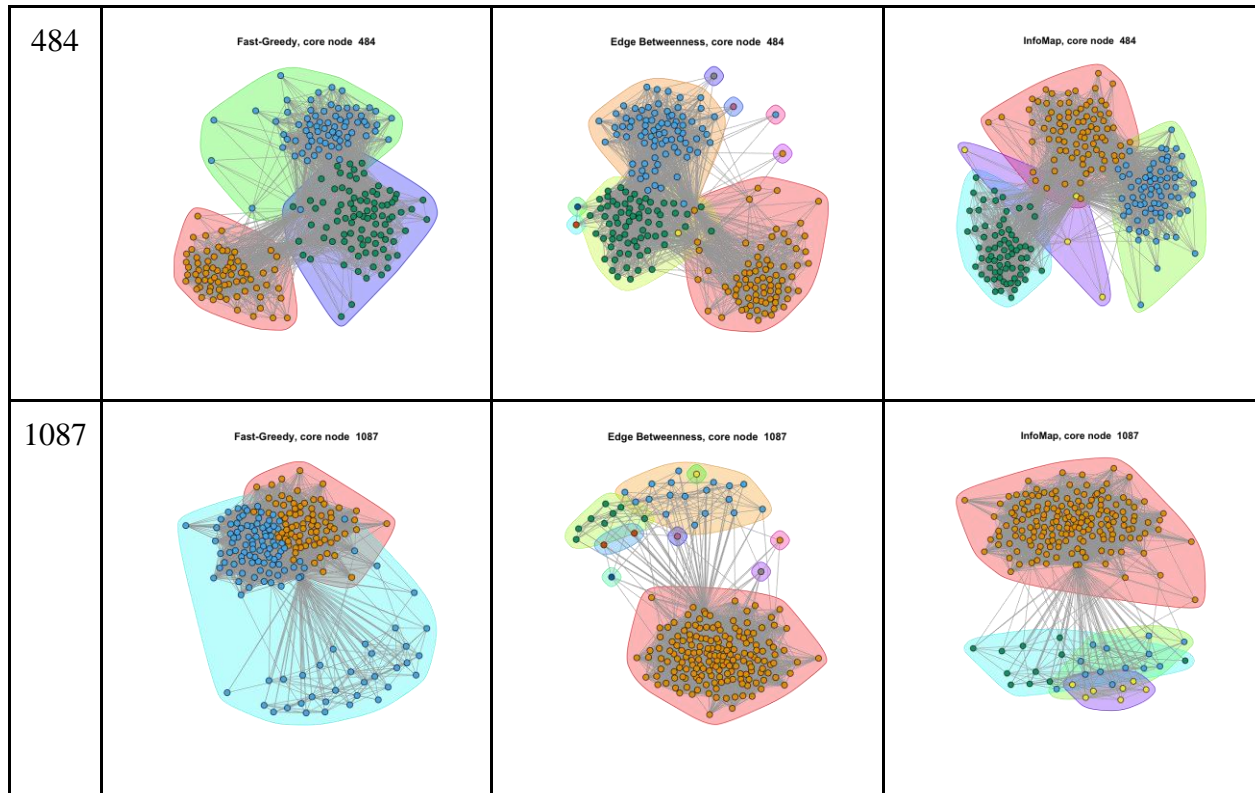
Hint Useful function(s): `cluster_fast_greedy`, `cluster_edge_betweenness`, `cluster_infomap`

For each core node listed above, we used the `cluster_fast_greedy` method, the `cluster_edge_betweenness` method, and the `cluster_infomap` method from `igraph` to find the community structures created using each strategy.

The Fast Greedy method implements the fast greedy modularity optimization algorithm for finding community structure. Edge Betweenness method calculates the edge betweenness of the graph by removing the edge with the highest edge betweenness score, recalculate edge betweenness of the edges and remove the one with the highest score. The idea of the edge betweenness based on community structure detection such that edges connecting modules would have high edge betweenness since all the shortest paths from one module to another must traverse through them. The Infomap method finds the community structure that minimizes the expected description length or a random walk. It uses the probability flow of random walks on a network as a proxy for information flows, and decompose the network into modules. The community structures can be seen below in Table 1.

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

ID	Fast-Greedy	Edge-Betweenness	Infomap
1	<p>Fast-Greedy, core node 1</p> 	<p>Edge Betweenness, core node 1</p> 	<p>InfoMap, core node 1</p> 
108	<p>Fast-Greedy, core node 108</p> 	<p>Edge Betweenness, core node 108</p> 	<p>InfoMap, core node 108</p> 
349	<p>Fast-Greedy, core node 349</p> 	<p>Edge Betweenness, core node 349</p> 	<p>InfoMap, core node 349</p> 



We also recorded the modularity of each community structure, seen below in Table 2.

Table 2: Modularity of core nodes using various community detection algorithms

ID	Fast-Greedy	Edge-Betweenness	Infomap
1	0.413	0.353	0.394
108	0.436	0.507	0.508
349	0.250	0.134	0.211
484	0.507	0.489	0.515
1087	0.146	0.028	0.027

The Fast-Greedy algorithm was most often to yield the highest modularity. The node with the ID of 484 tended to have the best modularity score overall, while the node with the ID of 1087 tended to have the worst. This means that the community detection algorithms had the easiest time creating distinct communities with 484 and the hardest with 1087.

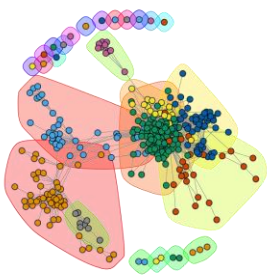
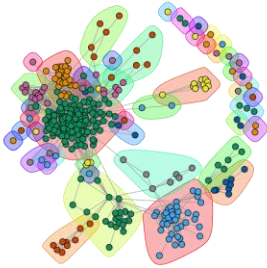
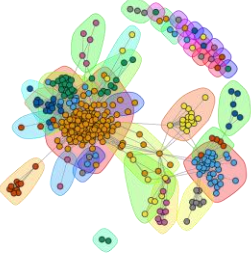
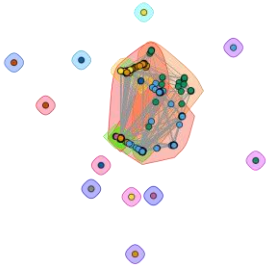
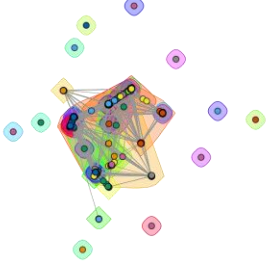
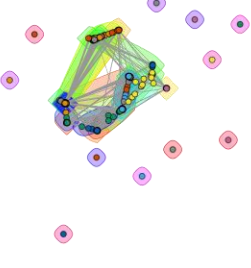
3.2. Community structure with the core node removed

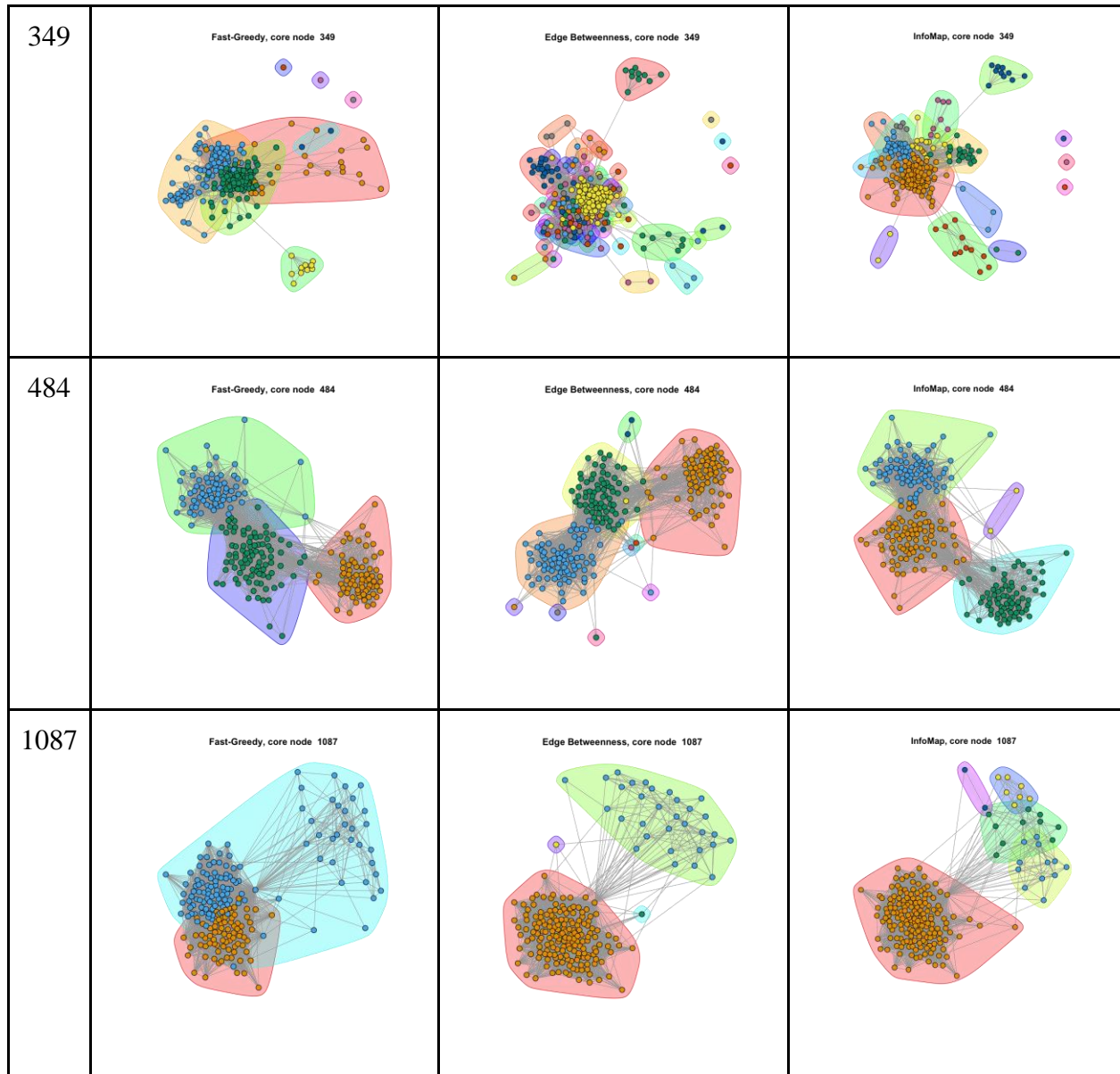
We also looked at the community structures after the code node is removed from that personalized network using the same five networks as the previous problems.

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

We used the same method as we did in Question 9, the only difference being that we only included the neighbors of the node we were using as the original user, but not the original user itself. The community structures can be seen below in Table 3.

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

ID	Fast-Greedy	Edge-Betweenness	Infomap
1			
108			



We also recorded the modularity of each community structure, seen below in Table 4.

Table 4: Modularity of core nodes using various community detection algorithms with code node removed

ID	Fast-Greedy	Edge-Betweenness	Infomap
1	0.442	0.416	0.418
108	0.458	0.521	0.521
349	0.246	0.151	0.234

484	0.534	0.515	0.543
1087	0.148	0.032	0.027

The modularity values were found to be very similar to that of Question 9, although in almost all cases the modularity value increased a little bit. This makes sense since trying to classify the core node would have made it difficult to cleanly assign nodes to communities. In addition, the nodes that were only neighbors with the core node were unconnected with the rest of the graph and became their own communities.

3.3 Characteristic of nodes in the personalized network

We examined the characteristics of nodes using two more measures: embeddedness, or the number of mutual friends that a node shares with a core node; and, dispersion, or the sum of the distances between every pair of mutual friends that the node shares with the core node in a graph where the node and core node are removed.

More information on embeddedness and dispersion was found in the paper at <http://arxiv.org/abs/1310.6753>.

QUESTION 11: Write an expression relating the Embeddedness of a node to its degree.

$$\text{card}(\text{degree}(v_{\text{core}}) \cap \text{degree}(v_i)),$$

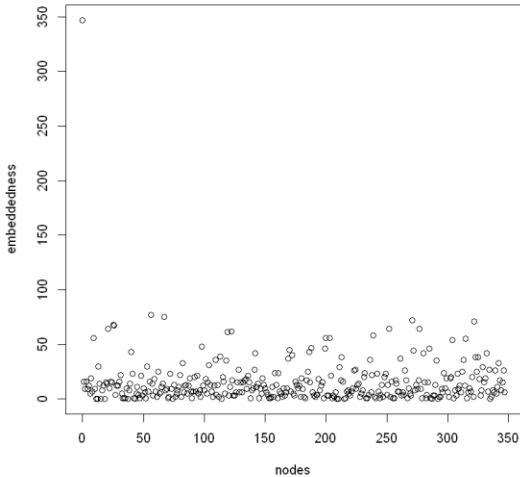
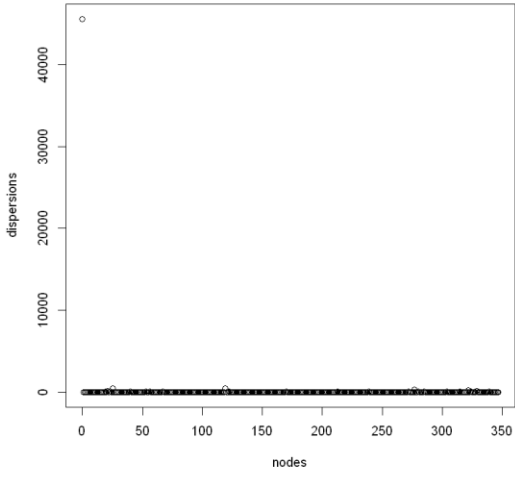
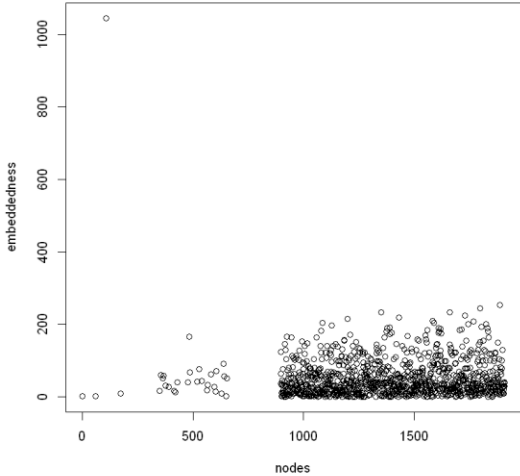
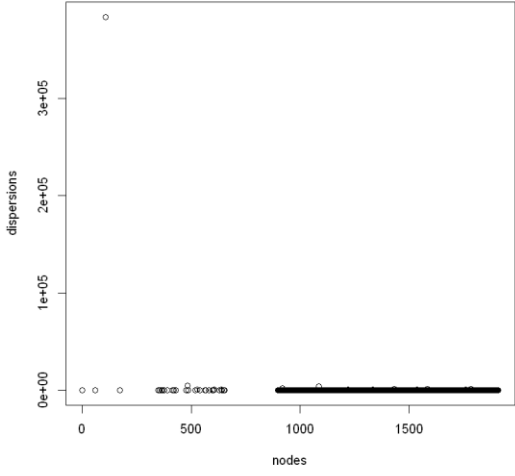
Where card is the cardinality of the set, \cap is the intersection of that set, v_{core} is the core node, and v_i is the non-core node being compared.

QUESTION 12: For each of the core node's personalized network (use the same core nodes as Question 9), plot the distribution of embeddedness and dispersion. In this question, you will have 10 plots.

Hint Useful function(s): *neighbors*, *intersection*, *distances*

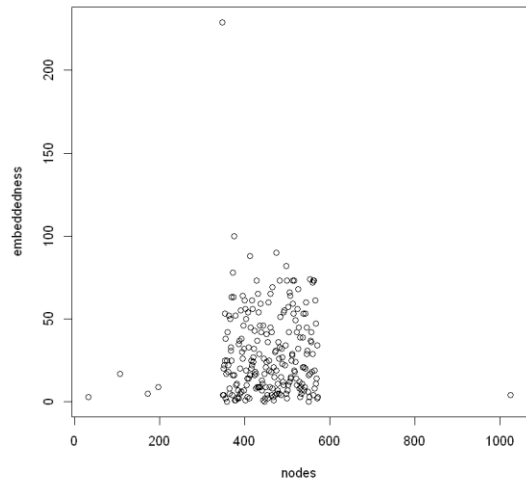
For each personalized network, we found the embeddedness and dispersion values. For embeddedness, we took the length of the intersection of the neighbors of the core node and the node we were examining. For dispersion, we deleted the node and core node from the graph and then took the distances of each mutual friend to another. If the distance for each was greater than 2, we added that to the dispersion total. The results can be seen below in Table 5.

Table 5: Distributions for embeddedness and dispersion

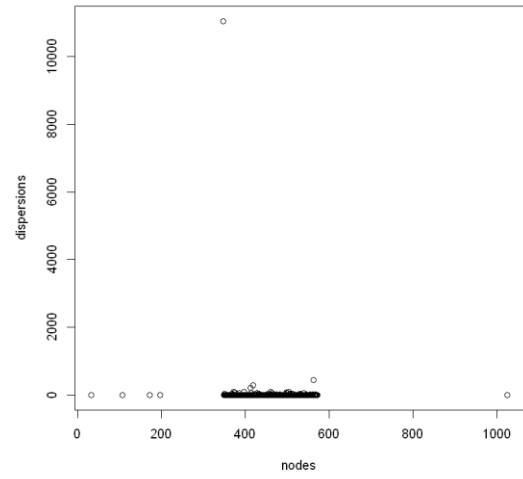
ID	Embeddedness	Dispersion
1	<p>Distribution of Embeddedness of core node: 1</p> 	<p>Distribution of Dispersion of core node: 1</p> 
108	<p>Distribution of Embeddedness of core node: 108</p> 	<p>Distribution of Dispersion of core node: 108</p> 

349

Distribution of Embeddedness of core node: 349

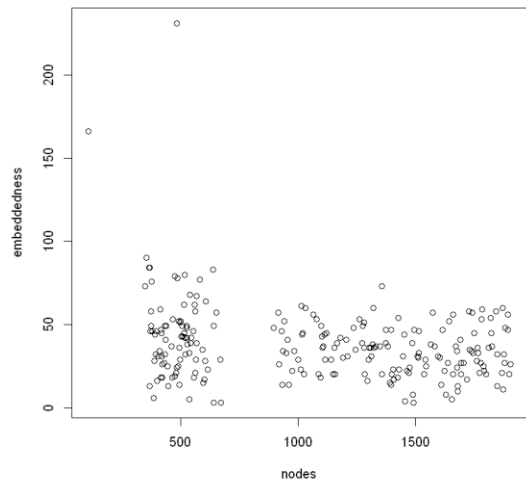


Distribution of Dispersion of core node: 349

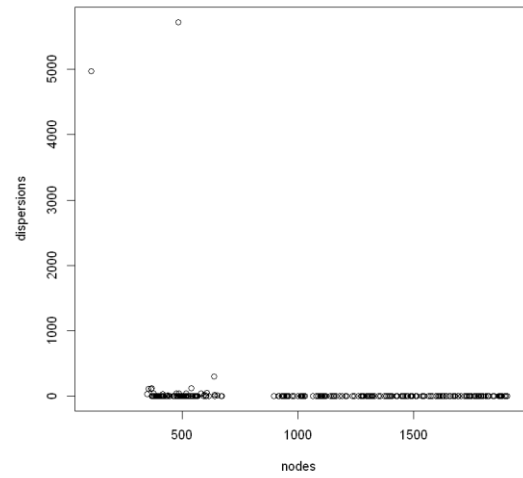


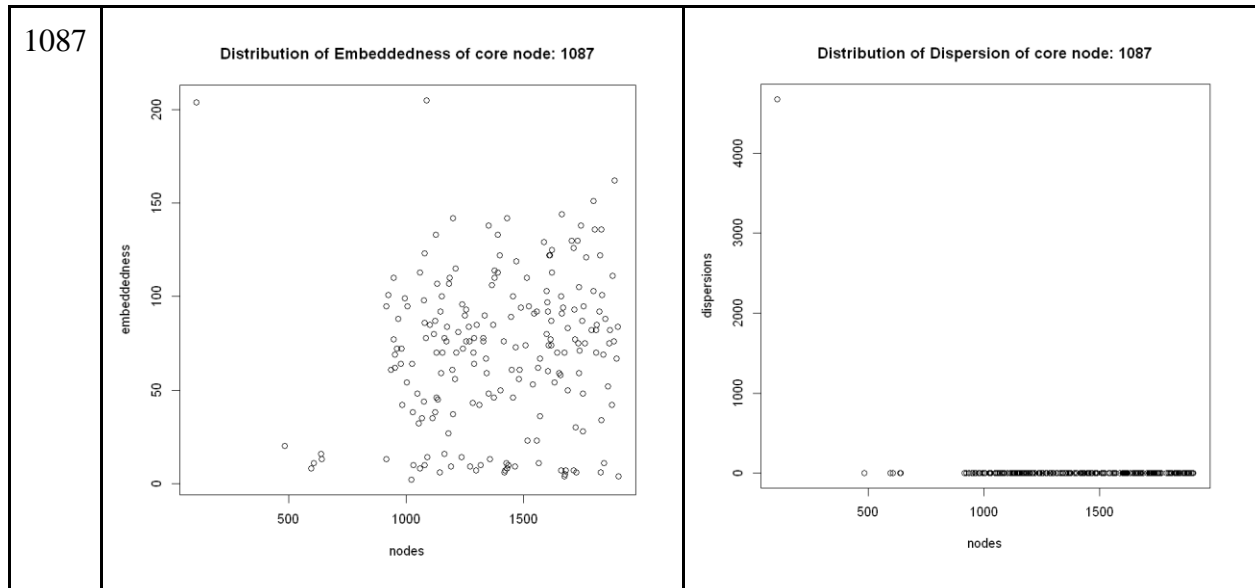
484

Distribution of Embeddedness of core node: 484



Distribution of Dispersion of core node: 484





The dispersion appears to be mostly uniformly low, with a few outliers in the graphs. Embeddedness looks more scattered across the plot, with no real direction positively or negatively. As expected, using dispersion in conjunction produces significantly higher accuracy for estimating tie strength, roughly twice as much as embeddedness.

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. To detect the community structure, use Fast-Greedy algorithm. In this question, you will have 5 plots.

We plotted the community structure for each personalized network of the five node IDs, seen below in Table 6. The node with the maximum dispersion is enlarged and colored hotpink along with the edges incident to this node.

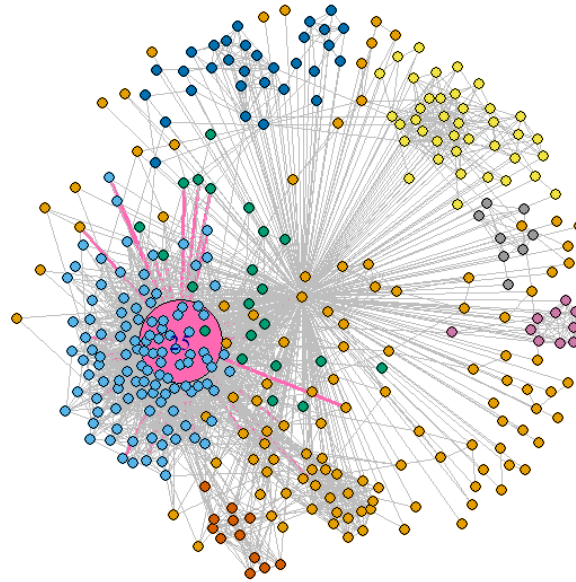
Table 6: Community structure of the personalized networks with maximum dispersion node highlighted

ID	Max dispersion	Personalized network
----	----------------	----------------------

1

25

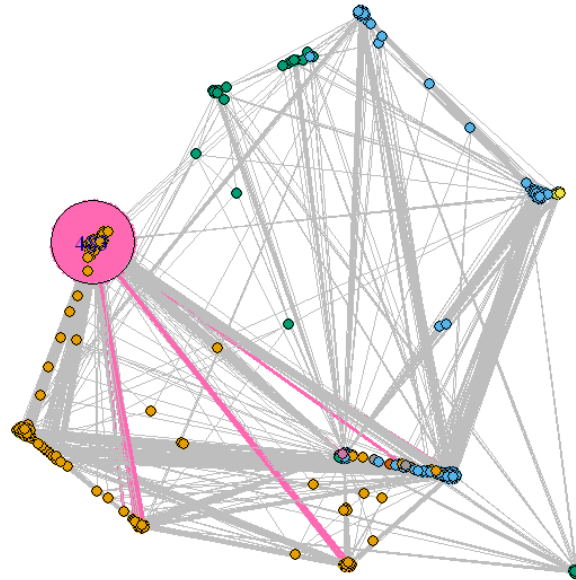
Community Structure (Max Dispersion Node, core node 1)



108

483

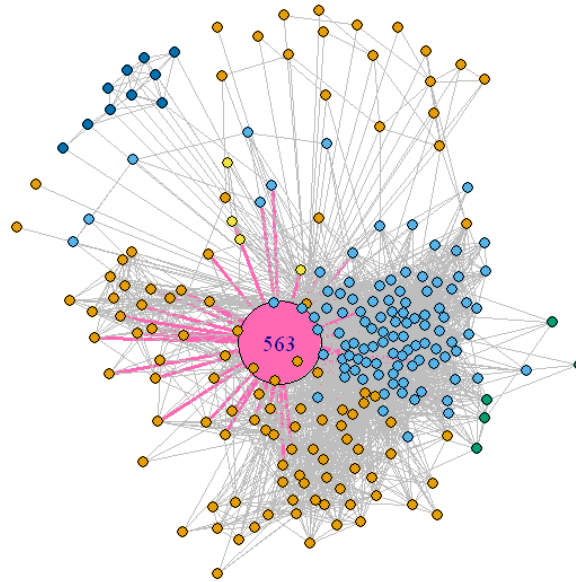
Community Structure (Max Dispersion Node, core node 108)

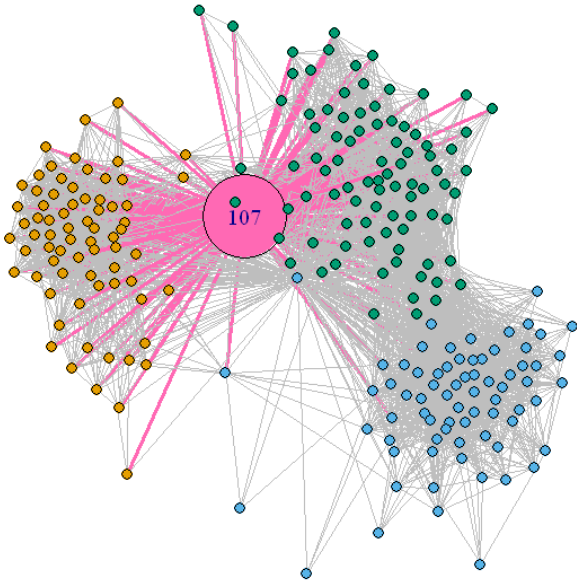


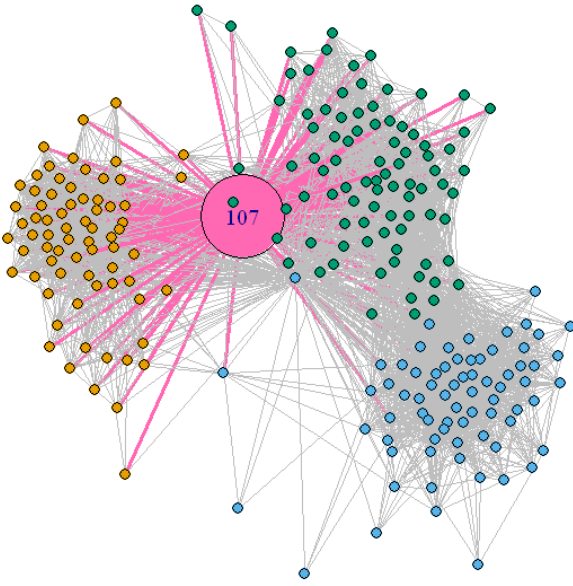
349

563

Community Structure (Max Dispersion Node, core node 349)



484	107	<p data-bbox="581 247 1370 281">Community Structure (Max Dispersion Node, core node 484)</p> 
-----	-----	--

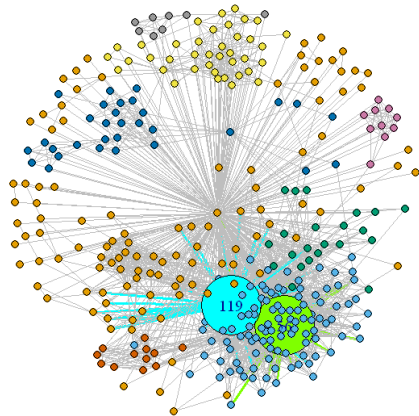
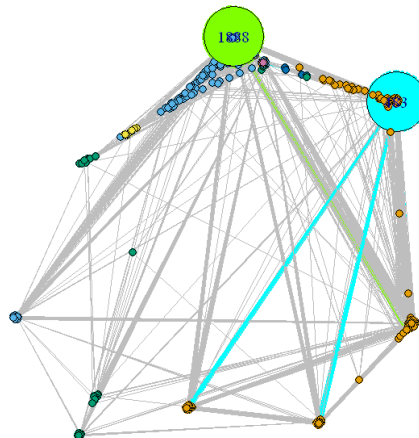
1087	107	<p>Community Structure (Max Dispersion Node, core node 484)</p> 
------	-----	---

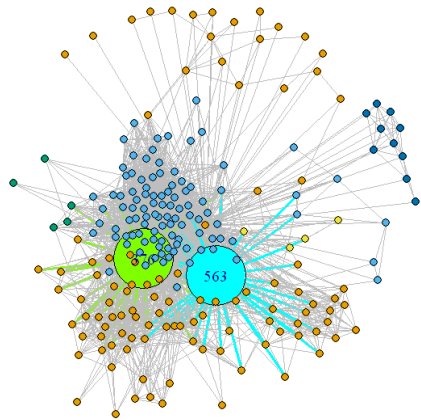
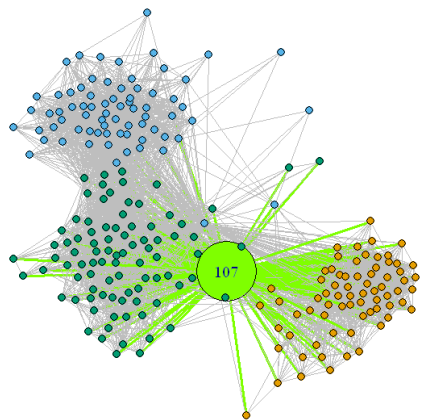
Some of the maximum dispersion nodes are difficult to see because the other nodes are covering it, but often the edges stemming from this node can be seen with the hotpink color. As expected, these nodes tend to centralized in the graph.

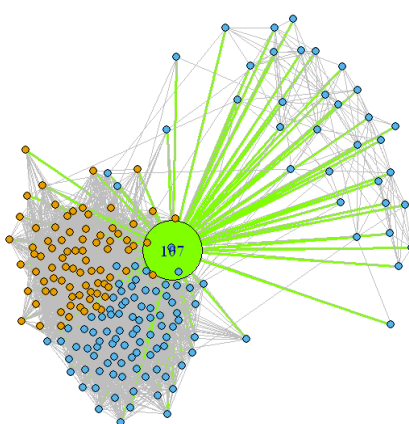
QUESTION 14: Repeat Question 13, but now highlight the node with maximum embeddedness and the node with maximum dispersion/embeddedness. Also, highlight the edges incident to these nodes

The community structure of the personalized networks with the maximum dispersion node and the maximum embeddedness node are highlighted in Table 7.

Table 7: Community structure of the personalized networks with maximum dispersion node and maximum embeddedness highlighted

ID	Max embeddedness	Max dispersion/ embeddedness	Personalized network
1	56	119	<p>Community Structure (Max dispersion/embeddedness Node, core node</p> 
108	1888	483	<p>Community Structure (Max dispersion/embeddedness Node, core node 1</p> 

349	376	563	<p>Community Structure (Max dispersion/embeddedness Node, core node 3</p>  <p>A network graph visualization showing a central node labeled 563 (cyan) connected to a large cluster of nodes. The nodes are colored in shades of blue, green, and orange. The graph is dense with many edges, indicating a highly interconnected community structure.</p>
484	107	107	<p>Community Structure (Max dispersion/embeddedness Node, core node 4</p>  <p>A network graph visualization showing a central node labeled 107 (green) connected to a large cluster of nodes. The nodes are colored in shades of blue, green, and orange. The graph is dense with many edges, indicating a highly interconnected community structure.</p>

1087	107	107	<p>Community Structure (Max dispersion/embeddedness Node, core node 11</p> 
------	-----	-----	---

Again, the maximum embeddedness node and the maximum embeddedness/dispersion nodes were highlighted and enlarged in the graph. The related edges were also highlighted in the same color. For IDs 484 and 1087, the maximum embeddedness node and the maximum embeddedness/dispersion are the same, 107. Additionally, 107 is the same value for the max dispersion node as well.

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

The maximum dispersion node is the node that, essentially has the greatest sum of distances between mutual friends with the core node (when said node and core node are removed), where a distance amount greater than 2 would be considered a distance value. The node and the core node are not always in the same community, which makes sense because a farther distance apart would mean that the dispersion value is greater.

The maximum embeddedness node is the node that shares the maximum number of friends with the core node. The majority of the plots show the node in the center of the communities. This makes sense because the core node is a neighbor of every node in the personalized network, and is placed in the center for the easiest graph structure. Since the max embeddedness node should have the next biggest amount of neighbors, it would make sense to also be placed somewhere centralized amongst the communities.

The maximum dispersion nodes and the maximum embeddedness nodes can often be the same one, when there is distance between the node and the core node but they share many of the same mutual friends.

The maximum dispersion/embeddedness node represents the nodes that have large dispersion and small embeddedness. This means that there aren't many mutual friends with the core node and the friends they do have aren't strongly connected. In relation to Facebook, this could be two people that became Facebook friends away from their typical social hangouts where there would be a lot of mutual friends. The node IDs for 484 and 1087 have the same node for dispersion/embeddedness and embeddedness and dispersion, which means that the dispersion values must be very high to counteract the comparatively high embeddedness values.

4. Friend recommendation in personalized networks

Sometimes it is desirable to predict future links between pairs of nodes in social networks. In the example of this Facebook network, it is equivalent to recommending friends to users. In this part of the analysis, we explored some neighborhood-based measures for friend recommendation.

4.1. Neighborhood based measure

There are three different neighborhood-based measurements between two nodes: common neighbor measure, Jaccard measure, and Adamic-Adar. Below is the equation for each method.

- Common neighbor measure between node i and node j is defined as

$$\text{Common Neighbors}(i, j) = |S_i \cap S_j|$$

- Jaccard measure between node i and node j is defined as

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

- Adamic-Adar measure between node i and node j is defined as

$$\text{Adamic Adar}(i, j) = \sum_{k \in S_i \cap S_j} \frac{1}{\log(|S_k|)}$$

4.2. Friend recommendation using neighborhood based measures

We utilized the neighborhood based measurements defined above to ‘recommend’ friends to users in the network. In order to recommend t new friends to some user i in the network, we followed the following algorithm.

1. For each node in the network that is not a neighbor of i , compute the measurement between node i and the node that is not a neighbor of i .
2. Pick t nodes that have the highest measurement with node i and ‘recommend’ these nodes as friends to node i .

4.3. Creating the list of users

Before we can apply the algorithm to the personalized network of node ID 415, we need to define a list of users who we want to recommend. Therefore, we created a list of all nodes with degree 24.

QUESTION 16: What is $|Nr|$?

In our Facebook network, we created a personalized graph for node ID 415 and got the degree list of the graph, then counted the nodes that had a degree of 24. We found that $|Nr|$ is 11.

4.4. Average accuracy of friend recommendation algorithm

We then applied the three different types of friend recommendation algorithms to the users in list Nr . We defined an average accuracy measurement to compare the performances of the friend recommendation algorithms. The average accuracy measure can be completed in two steps:

1. Compute the average accuracy for each user in the list Nr .
2. Compute the average accuracy of the algorithm by averaging across the accuracies of the user in the list Nr .

In the first step above, we can further expand by iterating over the following steps 10 times and then taking the average:

1. Remove each edge of node i at random with probability 0.25, In this context, it is equivalent to deleting some friends of node i .
2. Use one of the three neighborhood based measures to recommend $|R_i|$ new friends to the user i .

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

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?

Table 8 below summaries our average measurements. From our observation, it seems that all three methods can perform really well. However, given with all the calculated values below, common neighbors measure is the best among the three.

Table 8: Average accuracy values for Common Neighbors, Jaccard, Adamic Adar

Common Neighbors	Jaccard	Adamic Adar
0.8520	0.8396	0.8346

2. Google+ network

Next, we looked at the structure and properties of the Google+ network. To accomplish this, we used the data found at <http://snap.stanford.edu/data/egonets-Gplus.html>, namely gplus.tar.gz. To access this data, we unzipped the data. We created personal networks for those who had made more than 2 circles of friends.

QUESTION 18: How many personal networks are there?

In this dataset, there are a total of 132 personalized networks. There are 57 users who have more than two circles.

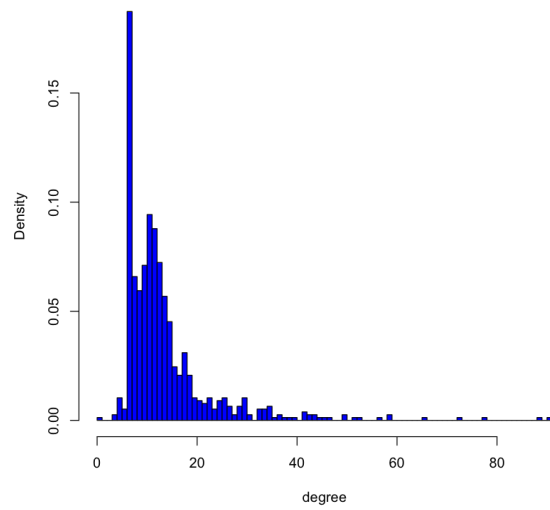
QUESTION 19: For the 3 personal networks (node ID given below), plot the in-degree and outdegree distribution of these personal networks. Do the personal networks have a similar in and out degree distribution. In this question, you should have 6 plots.

- 109327480479767108490
- 115625564993990145546
- 101373961279443806744

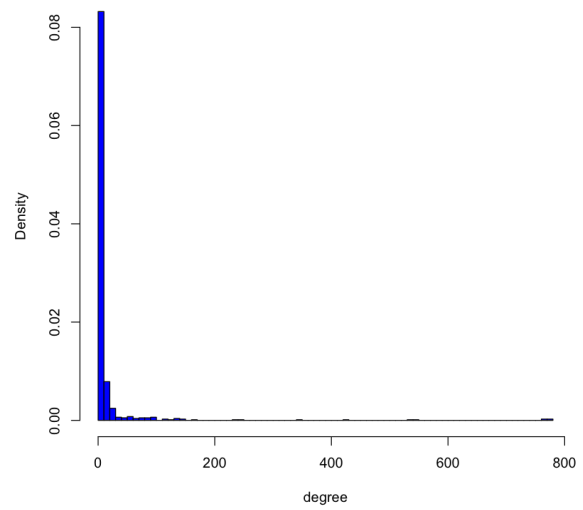
The in- and out-degree distributions of the personal networks for the given IDs are seen below in Table 9.

Table 9: Degree distributions of personal networks

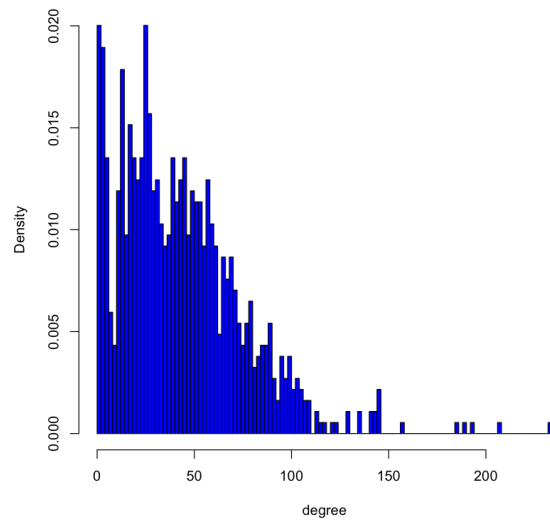
109327480479767108490 In-Degree Distribution



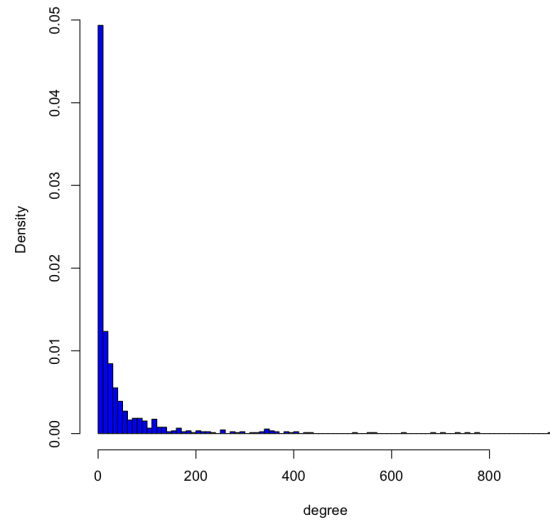
109327480479767108490 Out-Degree Distribution

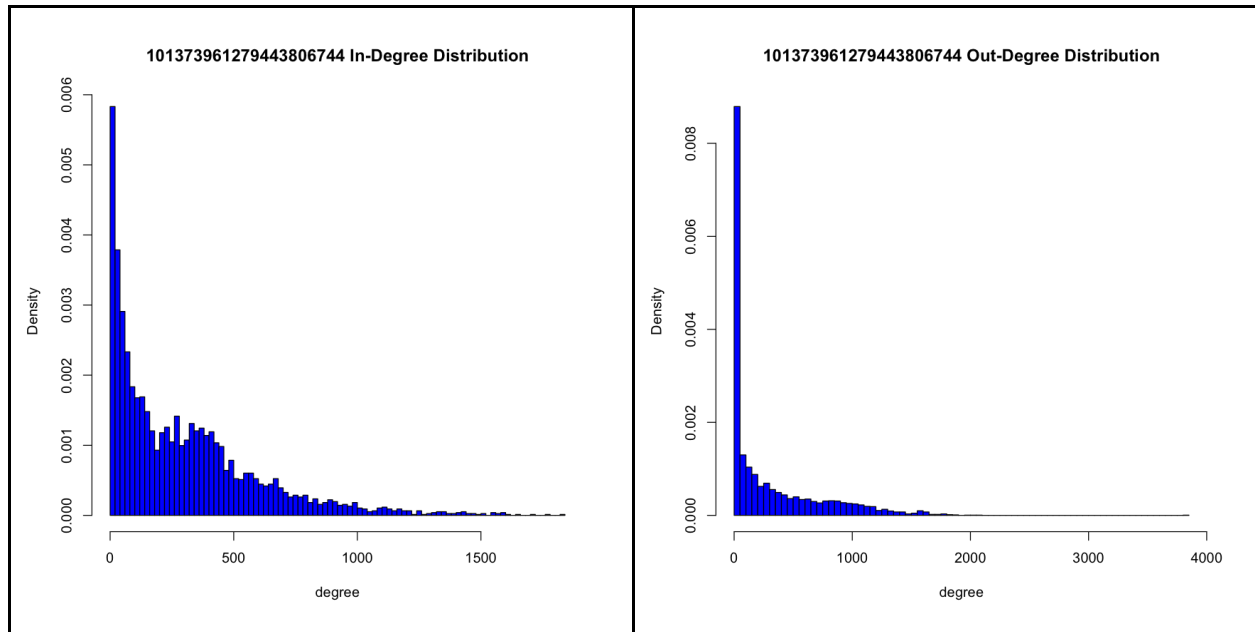


115625564993990145546 In-Degree Distribution



115625564993990145546 Out-Degree Distribution





Between the three Node IDs, the in-degree distributions show some variance. Nodes 109327480479767108490 and 101373961279443806744 have personalized networks that appear to exponentially fall off as the degree increases. In both networks, a very small minority of members have the highest in-degrees. These might be thought of as “hubs,” or very popular people within the network that most people connect to. If these hubs are deleted, it is likely that the network will be heavily affected. In contrast, if one of the low-degree nodes are removed, the network structure may not change much.

Node 115625564993990145546’s personalized network appears to have more of a linear fall-off, so a larger portion of the users have a relatively high in-degree compared to the other two networks. This may suggest that the network does not depend nearly as much on “hubs,” and that the users are closer together in terms of how popular they are. The out-degree distributions appear the same across all three personalized networks.

1. Community structure of personal networks

Again, we also looked at the community structures for the personal networks and the relationship between communities and user circles.

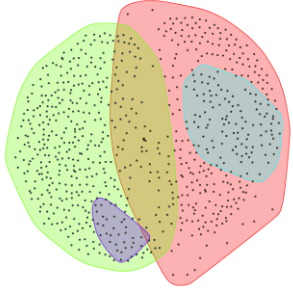
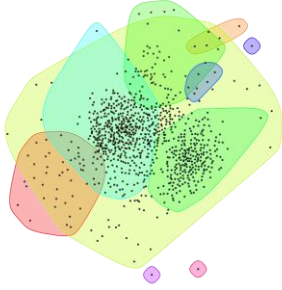
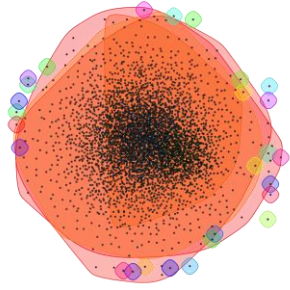
QUESTION 20: For the 3 personal networks picked in question 19, extract the community structure of each personal network using Walktrap community detection algorithm. Report the modularity scores and plot the communities using colors. Are the modularity scores similar? In this question, you should have 3 plots.

Table 10: Modularity scores of personal networks using Walktrap

Ego Node ID	Modularity Score
109327480479767108490	0.25
115625564993990145546	0.32
101373961279443806744	0.19

There is some variation on modularity score between the 3 personalized networks, so we would not consider the modularity scores that similar. From the visualization of the graph below, it indeed seems that the communities do not yield a very high modularity score:

Table 11: Community structure of personal networks using Walktrap

Walktrap Community Prediction Visualization		
109327480479767108490	115625564993990145546	101373961279443806744
<p>109327480479767108490 Walktrap Community Prediction</p> 	<p>115625564993990145546 Walktrap Community Prediction</p> 	<p>101373961279443806744 Walktrap Community Prediction</p> 

We also looked at how the circles and communities themselves were related, namely to measures:

- Homogeneity
- Completeness

In order to understand these concepts, we must first understand the related notation:

- C is the set of circles, $C = \{C_1, C_2, C_3, \dots\}$
- K is the set of communities, $K = \{K_1, K_2, K_3, \dots\}$
- a_i is the number of people in circle C_i
- b_i is the number of people in community K_i with circle information
- N is the total number of people with circle information
- C_{ji} is the number of people belonging to community j and circle i

And the expressions for entropy:

$$H(C) = - \sum_{i=1}^{|C|} \frac{a_i}{N} \log\left(\frac{a_i}{N}\right)$$

$$H(K) = - \sum_{i=1}^{|K|} \frac{b_i}{N} \log\left(\frac{b_i}{N}\right)$$

And conditional entropy:

$$H(C|K) = - \sum_{j=1}^{|K|} \sum_{i=1}^{|C|} \frac{C_{ji}}{N} \log\left(\frac{C_{ji}}{b_j}\right)$$

$$H(K|C) = - \sum_{i=1}^{|C|} \sum_{j=1}^{|K|} \frac{C_{ji}}{N} \log\left(\frac{C_{ji}}{a_i}\right)$$

From these formulas, we can derive the expressions for homogeneity:

$$h = 1 - \frac{H(C|K)}{H(C)}$$

And completeness:

$$c = 1 - \frac{H(K|C)}{H(K)}$$

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

Homogeneity measures how much variation there is in each predicted community. If every predicted community contains members that all share the same circle, that results in a higher homogeneity score. If a predicted community contains members that are equally distributed among all the possible circles, that leads to the lowest homogeneity score.

The completeness metric looks at each circle and identifies how much variation there is in each circle member's community assignment. If every member in a circle is predicted to be part of the

same community, then that circle has a high completeness score. If the distribution of community assignments within a circle is uniform, then that circle has a low community score.

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.

Table 12: Homogeneity and completeness values of the community networks

Node Id	109327480479767108490	115625564993990145546	101373961279443806744
Homogeneity	0.893595819134775	0.954893270716987	0.785019383353625
Completeness	0.496742463437966	0.689026673579128	0.0398169439013152

In looking at the community clustering visualization for Node 101373961279443806744, there is no clear community structure that can be observed. There is a significant amount of overlap for the bigger communities. It is thus not surprising to see a very low completeness score -- each of the predicted communities share users, so regardless if there is overlap of users between each circle, the completeness score will be lowered.

A potential reason for why the Walktrap algorithm failed to find groups that correspond well with the circles is that the degree-distribution showed that the graph was skewed towards low-degree users with a small minority of high-degree hubs. During random walks of Walktrap, this increases the probability that the walker will jump to a popular hub and then to another community, rather than to stay within its own circle.

One justification for why the homogeneity score is so comparatively high is that, out of the 31 communities that were found in this network, 28 of them had no members who also had circled ata. Since the homogeneity decreases by the amount of conditional entropies for each community, and these 28 communities with no circle-data users contribute zero conditional entropy, the homogeneity remains high. If all of the community members had circle data, we might expect the homogeneity score to lower.

For Node ID 109327480479767108490's network, the homogeneity score is significantly higher than the completeness score. A likely factor is that there are more predicted communities than there are circles. So each circle has more communities where it can assign its members to, which might lower the completeness. Another factor could be that the degree-distribution is skewed towards low-degree users, which could be detrimental to Walktrap's ability to find communities as a majority of the low-degree users will tend to jump to the same hub with high probability.

Finally, for Node 115625564993990145546's Network, the homogeneity and completeness scores were the highest among the three personalized networks. A potential reason is that the

degree distribution was not nearly as skewed towards low-degree users, and probably has less of a “hub-like” property. Walktrap thus has less of a chance of transferring the walker to a common hub that most users share, and can increase its ability to discover communities. Additionally, the higher homogeneity score is helped by the fact that there are communities that were discovered that have zero members with circle information. This means that those communities will not reduce the homogeneity by any amount.

Conclusion

In this project, we explored and learned various interesting properties of an undirected and directed network. We learned how to extract a personalized network from a larger network; we learned the different measurements such as dispersion vs embeddedness as well as calculating accuracy among the three measures: common neighbors, Jaccard, and Adamic Adar which was used to simulate ‘friend recommendation’ in the Facebook network. Furthermore, we also dived into the directed Google+ network and analyzed the homogeneity and completeness. Overall, we think this project is a good introduction to graph theory and machine learning on real dataset.