Project 1 Report

EE 232E
May 16, 2017

Gourav Khadge
Eun Sun Lee
Yusi Ou

I. Introduction

In this project, Facebook graph edgelist file is analyzed to study a social network and users' personal network. First, personal network is generated with the core nodes with more than 200 neighbors and their neighbor nodes. Different community structure algorithms such as fast-greedy, edge-betweenness, and infomap community detection are used to extract communities. In addition to the study across different algorithms, community detection algorithm's performance with core node and without it are compared and analyzed. Furthermore, the concept of dispersion and embeddedness are introduced. Dispersion is the sum of distances between all mutual friends' pairs. Embeddedness is the number of mutual friends one node shares with core node. These measures can be used to reveal characteristic of nodes in personal network. Across all the users, certain communities such as "family" and "college friends" are expected to be found. It is studied how one user's certain community is equivalent to another user's community. The evidence of equivalence can be backed up with measures such as Modularity Index, Clustering Coefficient, Density, Community Size, or other statistical features of communities. Lastly, Google+ data file is analyzed to show how the relationship tags users put on their friends overlap after the communities are extracted using Walktrap and Infomap. Also, the relationship between the tagging habit of users and the overlap between communities and circles are studied.

II. Problems

1. Network

The network is connected, with a diameter of 8 and an average degree of 43.691.

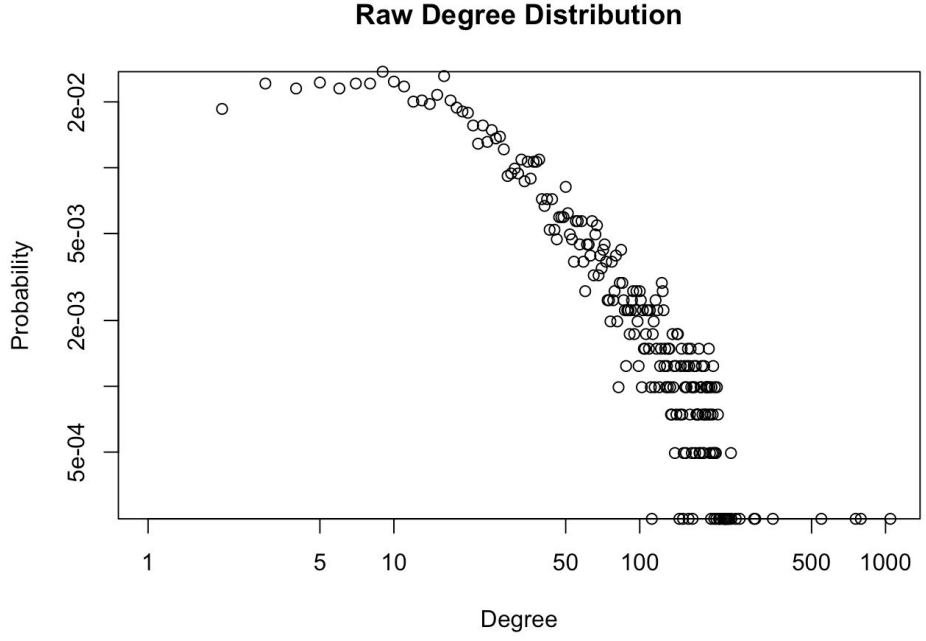

Figure 1.1 The raw degree distribution of the real network on log-xy axes. Note the low and high degree outliers (constant slope sections in the beginning and end of graph).

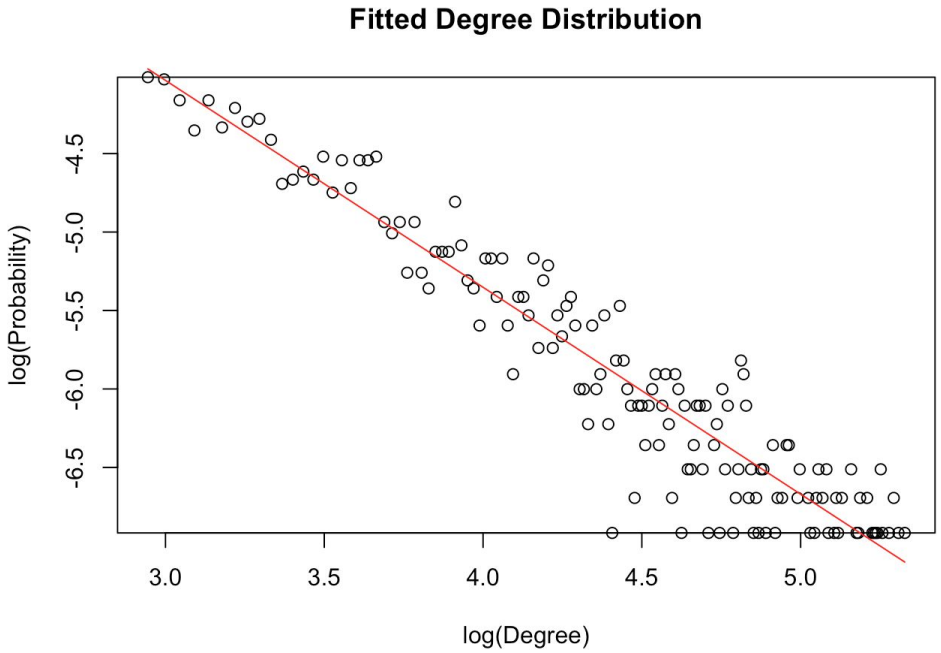

Figure 1.2 Log-xy linear fit of the degree distribution, with outliers removed. This portion of the data fits well with a power law distribution.

The degree distribution of the network does not exactly reflect a power law distribution due to low and high degree outliers. This is due to the fact that the data is sourced from an actual network, and is not expected to perfectly match theory. In real life, there can be extremely well connected nodes with degree greater than 500, which occur with very low probability, as well as very poorly connected nodes with degree close to 1 in a person's network. In some ways, this can be viewed as a form of quantization noise, since the number of nodes with any degree must be an integer. In order to obtain a good fit of the non-outlier data, the low and high degree outliers were removed prior to the fit. In Figure , the non-outlier data is shown with a power law fit. The total mean squared error is 0.06824, denoting a good fit showing that non-outlier real world data does follow a power law distribution in this network.

2. Node 1 Network

The network of Node 1 in the graph was analyzed. This is the personal network of the node, and all other nodes in the network are connected to node 1. Node 1 is therefore the core of its personal network. This network has 348 nodes and 2866 edges in total. A visualization of this network can be seen in Figure .

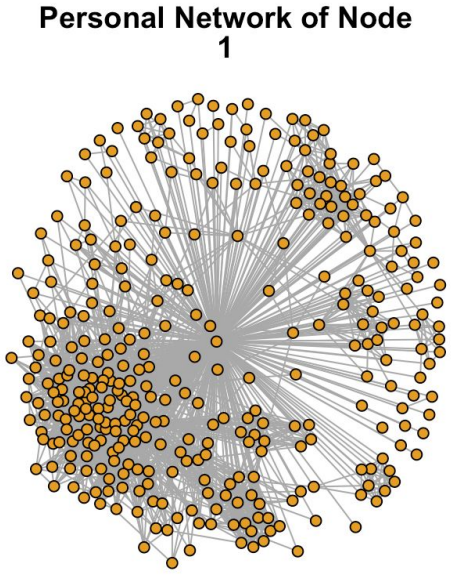

Figure 2.1 Personal network structure of Node 1. Note that all nodes are connected to a central node, which is Node 1.

3. Core Node Identification

The number of core nodes, defined as nodes with greater than 200 neighbors, is found to be 40 for the entire data set. The average degree of these core nodes is 279.375, compared to the average degree of the raw network which was previously found to be 43.691. The community structures identified using the fast greedy (FG), edge betweenness (EB) and infomap (IM) methods for the first core node (4039) are shown below. Fastgreedy uses a method that optimizes a modularity and finds communities. The communities are merged so that it would yield the largest increase in the value of modularity and stops when it cannot increase modularity anymore. Edge betweenness algorithm uses that edges connecting different communities tend to be contained in multiple shortest paths as they are the only place to go from one community to another. Edge betweenness is accurate but computes slowly as the calculation is complex. Lastly, Infomap is the method that finds community by minimizing a random walker's expected trajectory. The first eight communities identified in each method are color labeled in the histogram and correspondingly in the network graph for visualization. The nodes that fall outside of these first eight communities are colored grey.

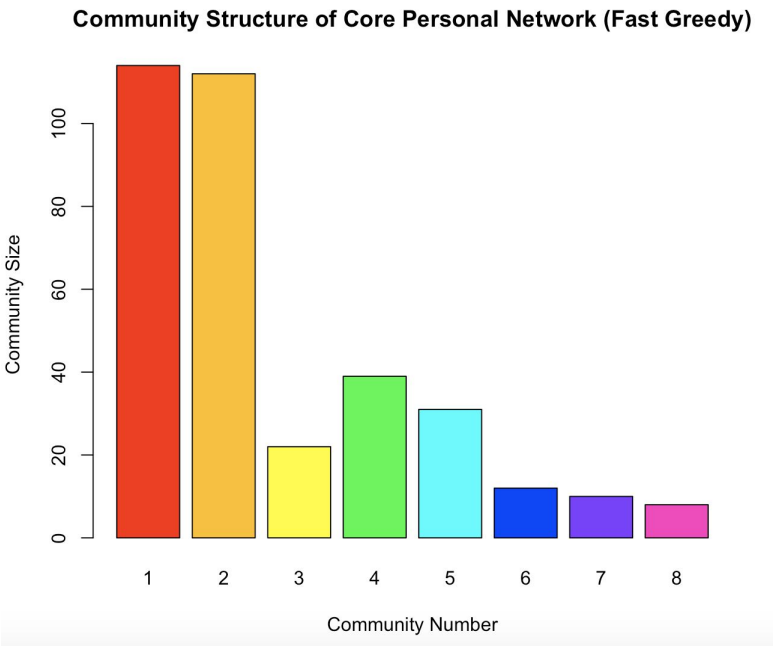

Figure 3.1 Community structure identified using FG method. This method identifies exactly eight communities across all nodes.

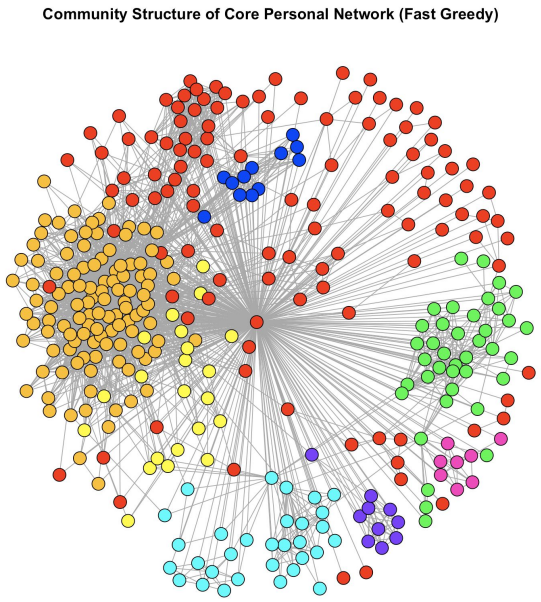

Figure 3.2 Community network structure identified using FG method. This method identifies exactly eight communities across all nodes, and each color corresponds to the histogram.

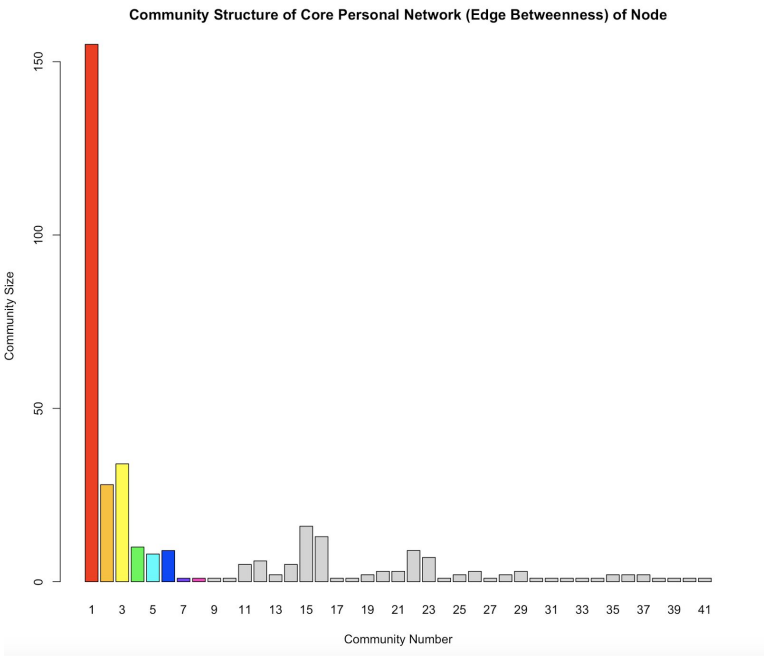

Figure 3.3 Community structure identified using EB method. This method identifies a few large communities and numerous small ones. The first eight communities are colored for visualization.

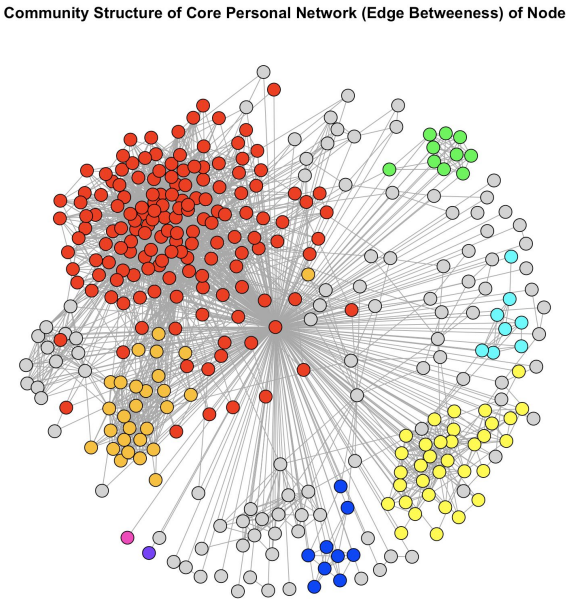

Figure 3.4 Community network structure identified using EB method. This method identifies exactly eight communities across all nodes, and each color corresponds to the histogram.The first eight communities are colored for visualization.

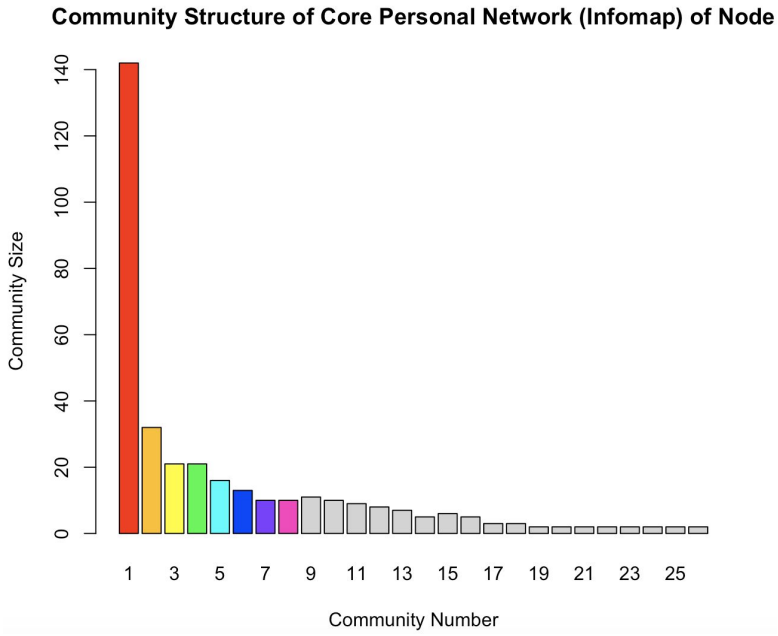

Figure 3.5 Community structure identified using IM method. This method identifies a few large communities and numerous small ones, but less than the EB method. The first eight communities are colored for visualization.

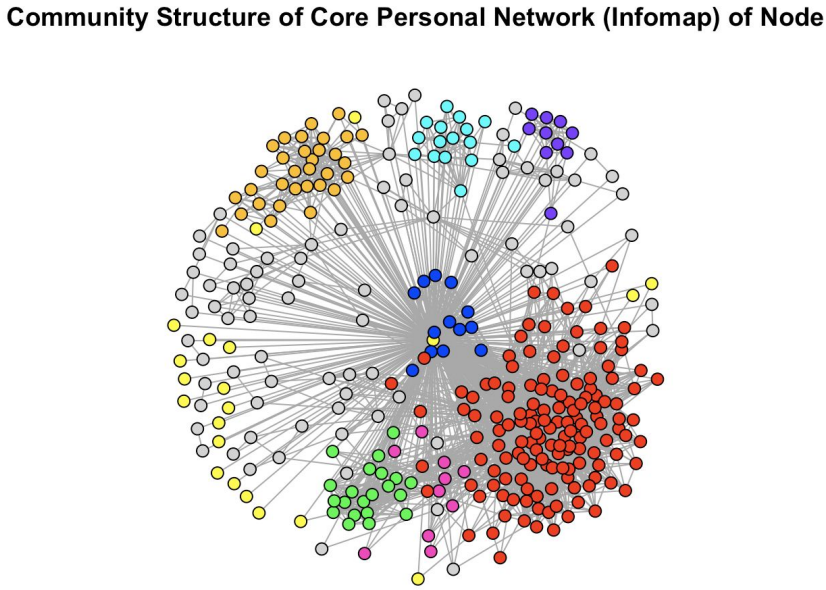

Figure 3.4 Community network structure identified using IM method. This method identifies exactly eight communities across all nodes, and each color corresponds to the histogram.The first eight communities are colored for visualization.

4. Community Detection on Core Neighbor Network

The core node from the network (the Core Personal Network of Node 4039) above was removed so that only the neighbors of the core node remained. This network is called the Core Neighbor network. While the Core Personal Network is connected by definition, as every node is connected to the core node, the Core Neighbor Network is not necessary connected. It is disconnected in this case. This changes the results of the community finding algorithms, as the core node's connections, which are useful for community finding, are no longer present. The results of the community finding algorithms on the Core Neighbor Network can be seen below.

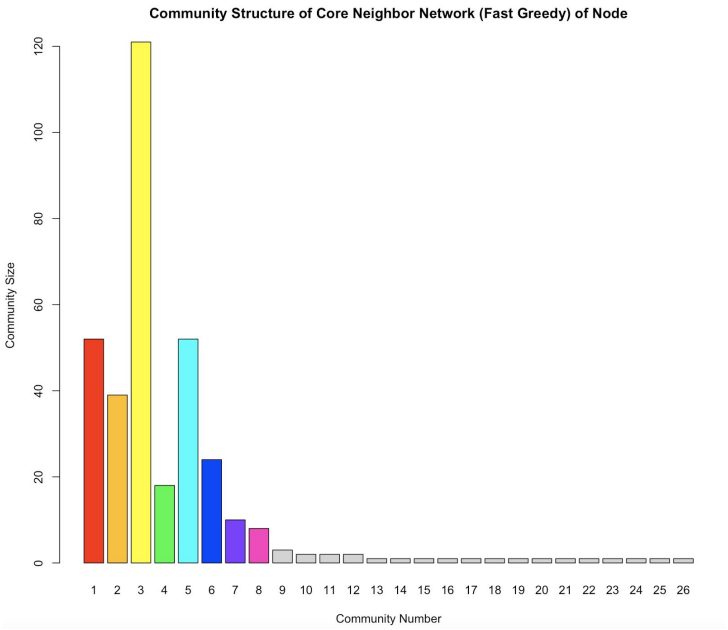

Figure 4.1 Community structure of core neighbor network (core node removed) using FG.

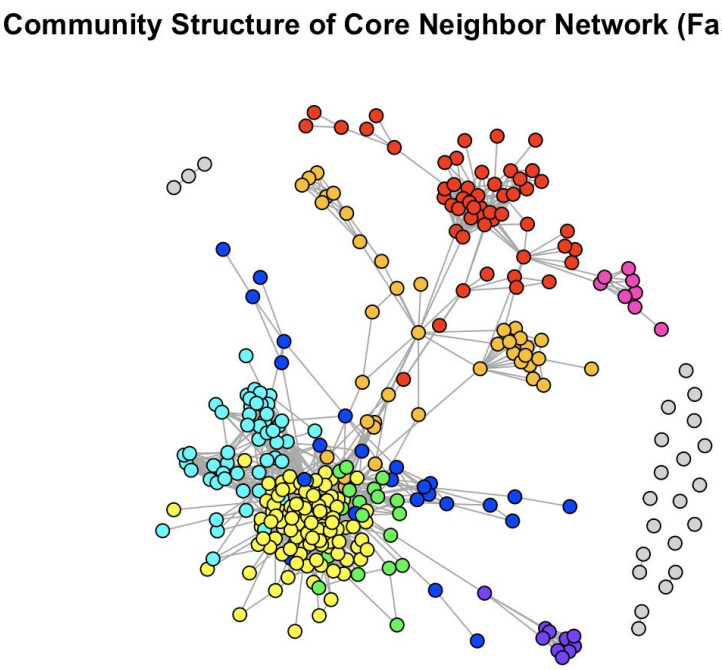

Figure 4.2 Community network structure identified using FG method on the Core Neighbor Network. This method identifies exactly eight communities across all nodes, and each color corresponds to the histogram. The first eight communities are colored for visualization.

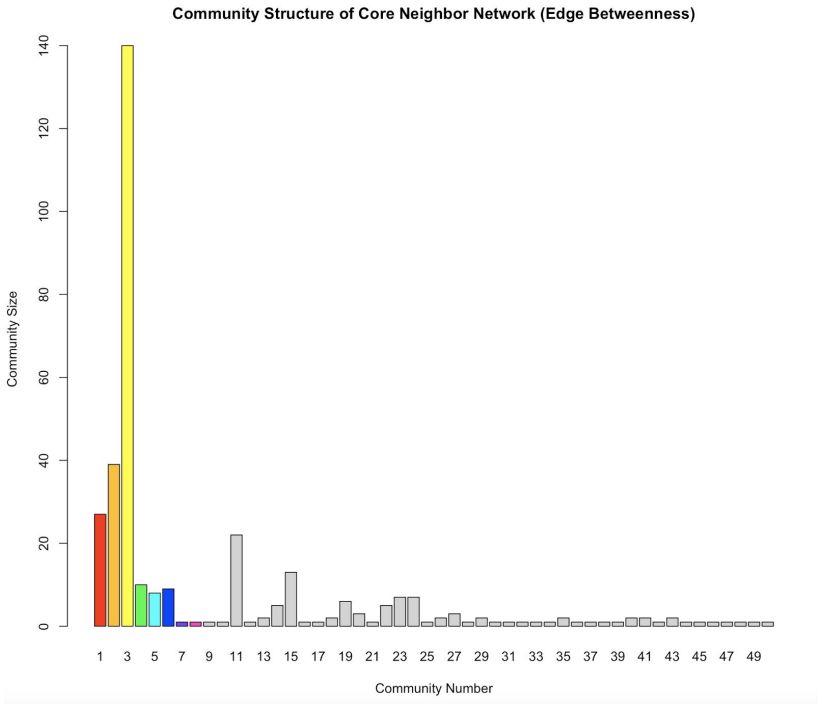

Figure 4.3 Community structure of core neighbor network (core node removed) using EB.

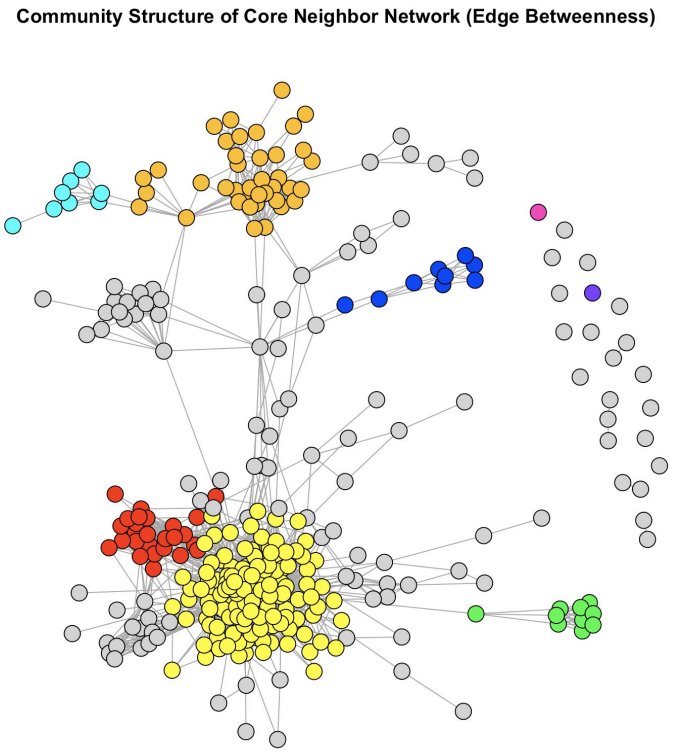

Figure 4.4 Community network structure identified using EB method on the Core Neighbor Network. This method identifies exactly eight communities across all nodes, and each color corresponds to the histogram. The first eight communities are colored for visualization.

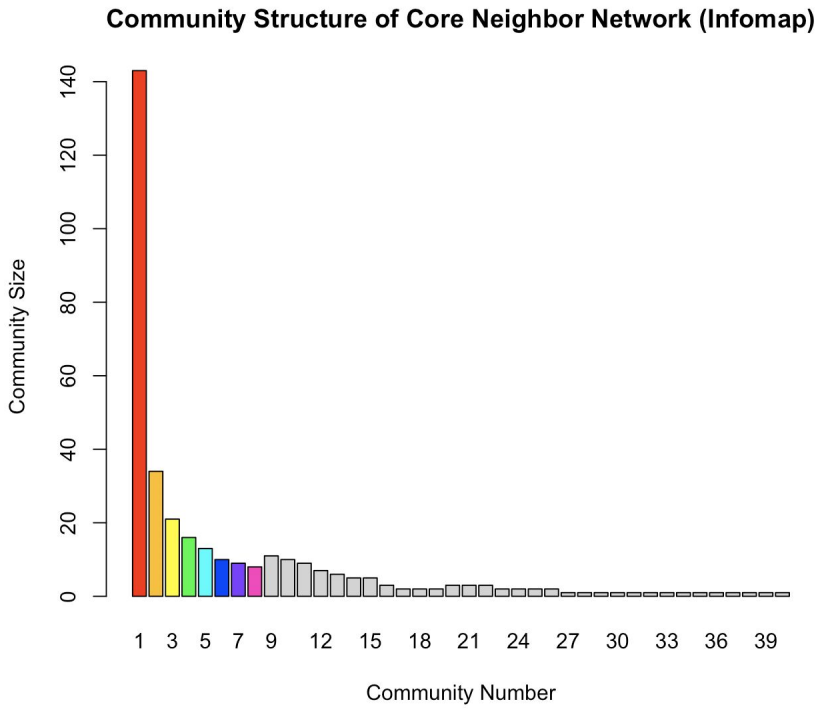

Figure 4.5 Community structure of core neighbor network (core node removed) using IM.

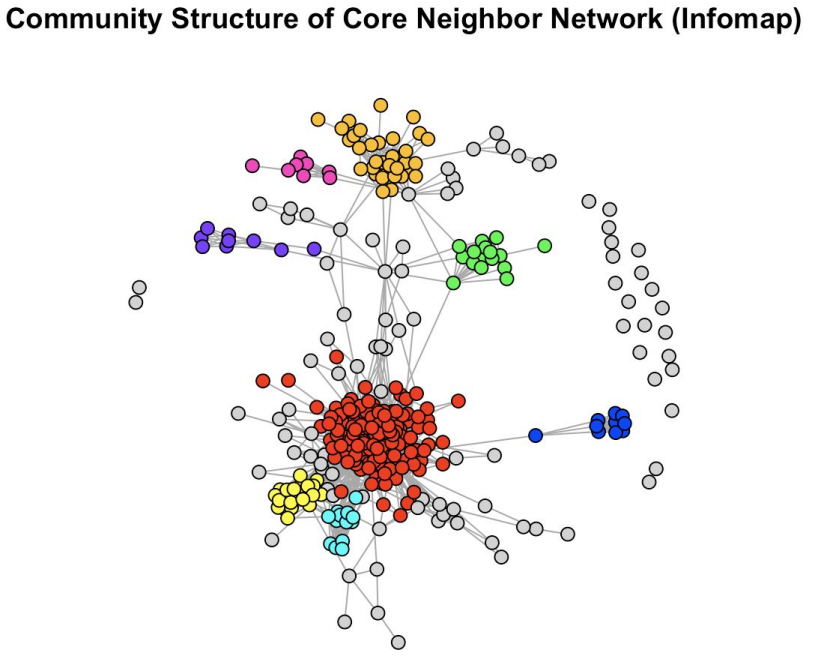

Figure 4.6 Community network structure identified using IM method on the Core Neighbor Network. This method identifies exactly eight communities across all nodes, and each color corresponds to the histogram. The first eight communities are colored for visualization.

5. Embeddedness and Dispersion

Embeddedness and dispersion are useful metrics for defining communities in core node networks. Embeddedness is the number of mutual friends a node shares with the core node. In a core node's network, this is equivalent to the degree of the node minus one (which corresponds to the connection with the core node). The embeddedness can be used to explain the degree of the connection between the core node and the selected node, and how closely the friend group (core node, friend node, and mutual friend nodes) are connected. The fast greedy method was used to find communities in our analysis, as the other methods identified many very small communities which we do not believe to be significant.

The dispersion is the sum of distances between every pair of the mutual friends a node shares with core node. In other words, between every pair of mutual friends, we consider the shortest path that does not pass through the said node and the core node. This gives a measure of how closely the mutual friends of said node and core node are. For married couples for example, dispersion is expected to be high as there are many connections between them that are not connected to each other.

For disconnected nodes, the distance is defined as the total number of nodes in the core personal network plus one. This was used in order to better define the difference between nodes that are weakly connected (for which the distance would be the diameter of the graph) and nodes that are not connected at all. This definition was useful in our community analysis. There are also other options not used such as diameter times two.

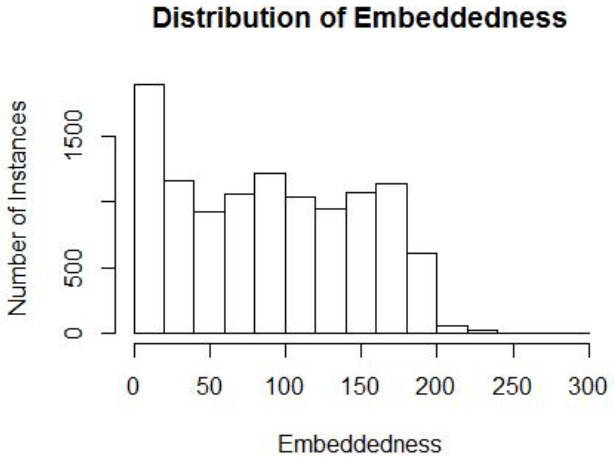

Figure 5.1 The distribution of embeddedness for the personal networks of all core nodes.

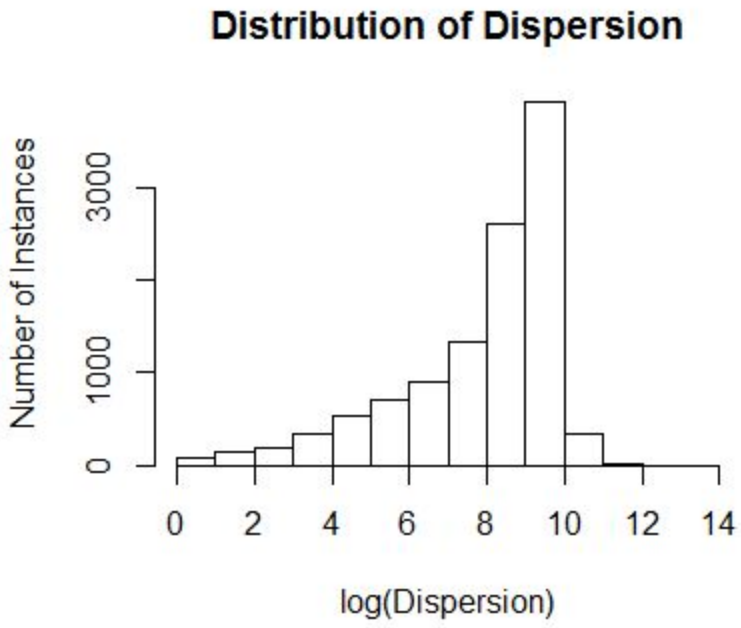

Figure 5.2 The distribution of dispersion for the personal networks of all core nodes.

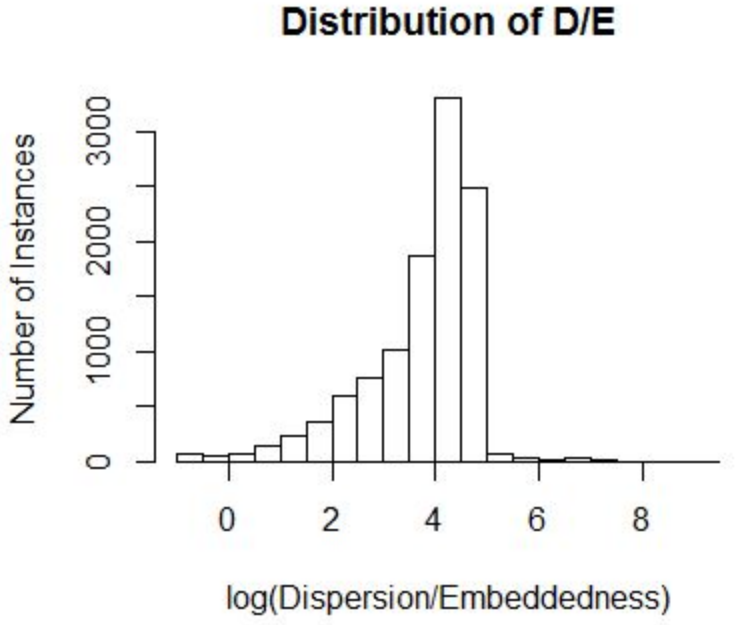

Figure 5.3 The distribution of dispersion over embeddedness for the personal networks of all core nodes.

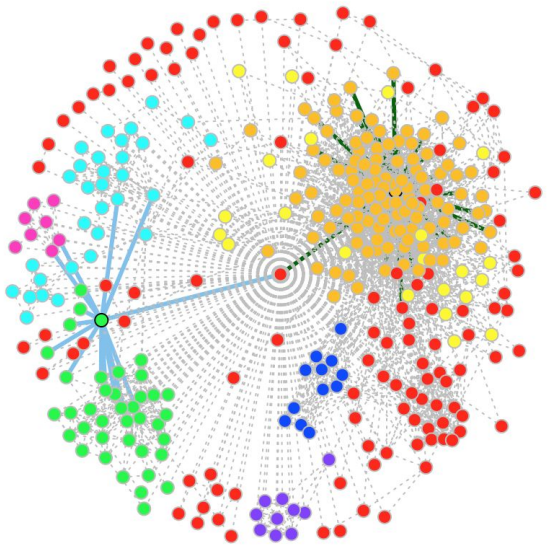

Figure 5.4 Node "19" in the personal network of core node "0", identified to be in community 4 (vertex green) has the highest dispersion and dispersion/embeddedness (incident edges blue). Node "56" is identified to be in community 2 (vertex orange), with the highest embeddedness (incident edges green).

Community	1	2	3	4	5	6	7
Number of Nodes	114	112	22	39	31	12	10
Dispersion	372.5	1246.0	517.9	1207.1	612.4	99.1	133.8
Embeddedness	7.9	27.5	10.4	9.7	6.2	12.6	7.9
D/E	45.4	30.0	22.9	77.2	42.9	7.3	29.6
Edge Density* Number of Nodes	7.8	**24.8**	5.0	9.7	6.0	10.2	8.7
D/E/ Number of Nodes	0.398	0.268	1.04	1.98	1.39	0.608	**2.96**

Community Structure of Core Neighbor Network (Fast Greedy)

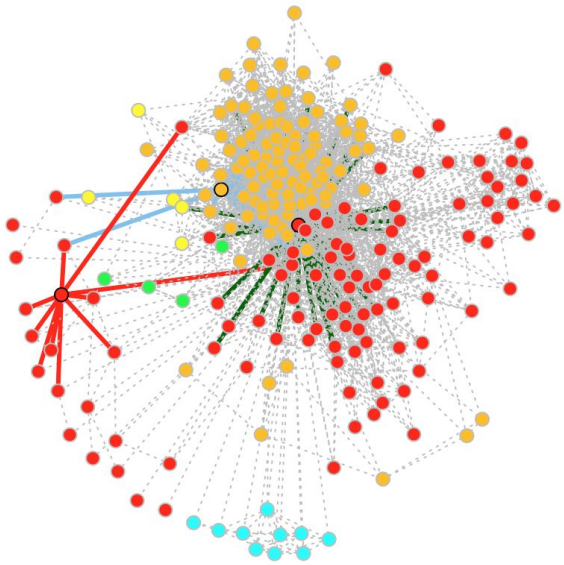

Figure 5.4 Node “387” in the personal network of core node “348”, identified to be in community 2 (vertex orange) has the highest dispersion (incident edges blue). Node “376” is identified to be in community 1 (vertex red), with the highest embeddedness (incident edges green). Node “532” is identified to be in community 1 (vertex red), with the highest dispersion/embeddedness (incident edges red).

Community	1	2	3	4	5
Number of Nodes	108	103	5	4	10
Dispersion	559.8	1703.2	96	64.5	204.7
Embeddedness	18.4	41.9	10.2	8.3	6.3
D/E	22.6	34.3	6.5	5.8	23.4
Edge Density* Number of Nodes	14.7	**36.0**	2.5	4.0	6.9
D/E/ Number of Nodes	0.209	0.333	1.30	1.45	**2.34**

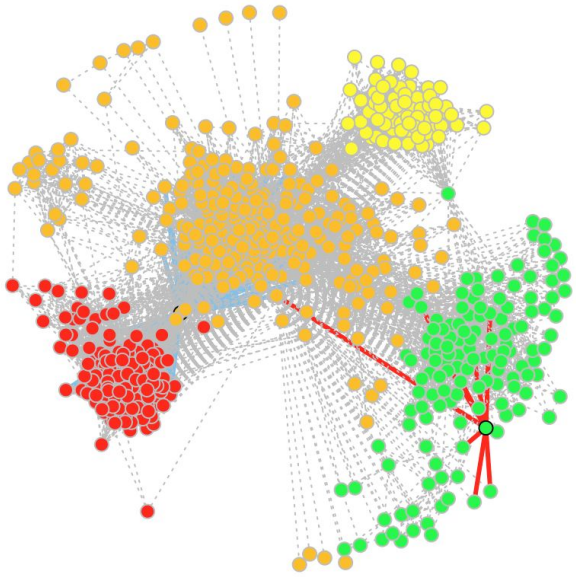

Figure 5.5 Node “2543” in the personal network of core node “1912”, identified to be in community 2 (vertex orange) has the highest dispersion and embeddedness (incident edges blue). Node “2456” is identified to be in community 4 (vertex green), with the highest dispersion/embeddedness (incident edges red).

Community	1	2	3	4
Number of Nodes	234	314	72	136
Dispersion	14043.3	5496.2	1359.8	437.1
Embeddedness	141.7	68.9	42.9	16.5
D/E	86.4	55.4	28.5	21.7
Edge Density* Number of Nodes	**139.0**	67.4	40.7	15.0
D/E/ Number of Nodes	0.369	0.1764	**0.396**	0.160

Taking a closer look at the node with the largest ratio of dispersion to embeddedness. We see the node has 7 mutual friends with the core node, with the distance matrix between its mutual friends

as below

	2041	2313	2317	2341	2342	2382	2541
2041	0	1	2	2	1	755	755
2313	1	0	1	2	1	755	755
2317	2	1	0	2	1	755	755
2341	2	2	2	0	2	755	755
2342	1	1	1	2	0	755	755
2382	755	755	755	755	755	0	755
2541	755	755	755	755	755	755	0

We see that it's dispersion value is dominated by two neighbor nodes which are inaccessible from the other 5 neighbors. On Figure 5.5, we see that these two neighbor nodes are are only friends with this node and the core node and no one else. This calls into question the usefulness of this particular implementation of the dispersion metric, because it seems to heavily be affected in these scenarios with extremely sparsely connected friends. We might also consider reducing the dispersion contribution from a disconnected node. Here it it set to the number of nodes in the graph, 755. It might be better to set the dispersion to be related to the diameter of the graph instead when the pair of nodes is unconnected.

This issue might not be as dominant if we used the dispersion metric used in the paper "Romantic Partners and the Dispersion of Social Ties" by Lars Backstrom.

6. Community Types

Several metrics were calculated in order to extract two community types present in each of the three core personal networks above. The metrics used to identify the two communities are bolded in the tables above. Dispersion and embeddedness are explained previously, and D/E is defined as the ratio between dispersion and embeddedness. The edge density is calculated using the edge_density, and represents the number of edges present between a subset of vertices (in this case, the vertices in each community), and the total number of possible edges.

We ended up choosing two metrics to distinguish two unique types of communities observed fairly universally: Edge Density*Number of Nodes and D/E/Number of Nodes.

The edge density multiplied by the number of nodes gives weight to communities which are large and very well connected. We observe at least one such community in each of the core node's personal networks. We believe that this community might correspond to a college network or some other large well connected organization that serves the purpose of creating large unit communities. The nodes in these communities are all well connected with each other, which makes sense for a tightly clustered college community.

D/E divided by the number of nodes was chosen due to its ability to distinguish what we believe are family groups. The family groups have very high D/E/Number of Nodes because nodes in the family do not share many mutual friends. This makes sense, as family members of a core node do not interact significantly with the core node's friend group. This is the opposite of the college group identified above. These low node count groups are defined by the highest D/E between all communities in the core node personal network.

7. Community Detection of Google+

Unlike Facebook, Google+ allows users to tag friends with circles. Circles are relationship tags such as "friends" or "family". One can tag friends in his or her circle regardless of whether they tag the user in their circles or not. The network, thus, is a directed structure. First, to create personal network for each ego node, files with extension .edges are read as graphs. From the filename, ego node's id can be extracted. However, ego nodes are not included in their edgelist files. To the graph created from .edge files, ego node is added with r function vertex(). Edges are missing between the ego node and other nodes in the network. With r function, add_edges(), all the edges are added. As mentioned, Google+ is a directed network so as.directed() function adds direction to the newly added edges. Finally, the graph with ego node and its edges to other nodes is created and it is the personal network of each ego node. Cluster_walktrap() and cluster_infomap() functions are used to extract the community structure of each personal network. Walktrap algorithm finds communities via random walks as short random walks are more likely to stay in the same community. On the other hand, infomap algorithm finds communities that minimizes random walker's path length. Circles from .circles extension files for each node are used to compare how the circle tags overlap with communities extracted from these algorithms. Generally, infomap algorithm gives communities that better match with circles. Only the personal networks with more than 2 circles are created and viewed. Figure 7.1 and figure 7.2 are the personal networks of 7th node extracted from Walktrap and Infomap algorithms. The plot does not include ego node. Nodes are colored by their community membership.

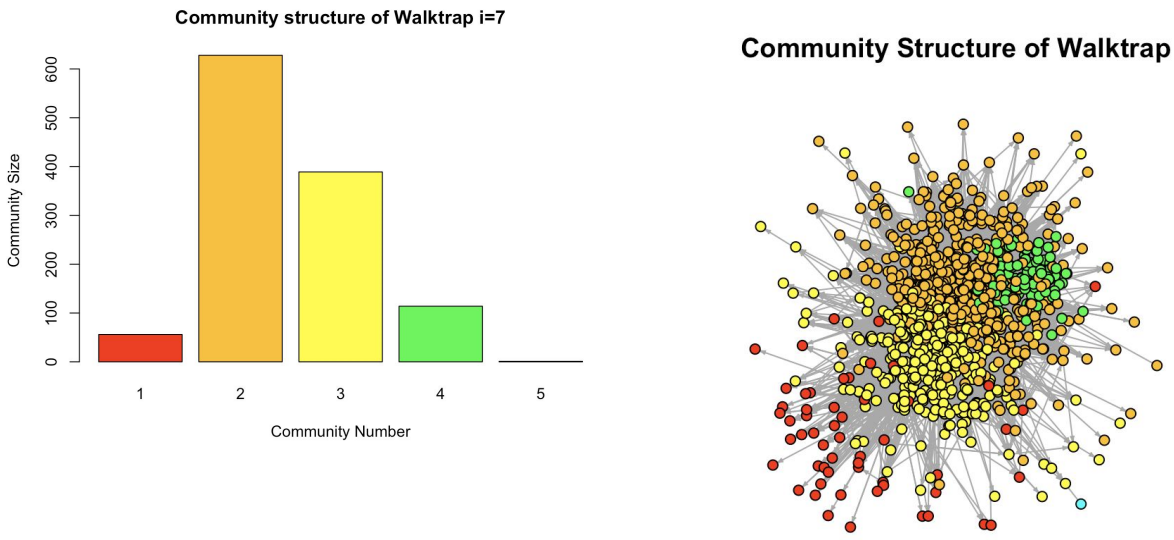

Figure 7.1 Community structures of 7th ego node, "100535338638690515335", extracted from Walktrap algorithm.

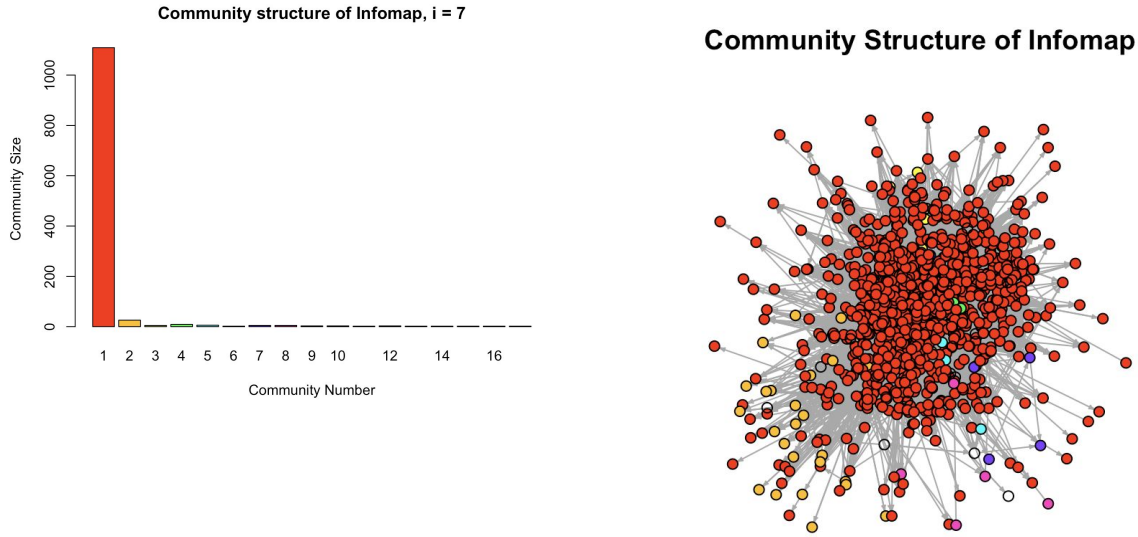

Figure 7.2 Community structures of 7th ego node, "100535338638690515335", extracted from Infomap algorithm.

To analyze how communities overlap with circles, subgraph is used to extract only the nodes in community 1 (red colored nodes in Community structure of infomap). These nodes are searched in different circles of the particular ego node. It was shown that most nodes in community 1 are found in circle 3. However, there are variations as the nodes are tagged with multiple

relationship circles. In Figure 7.3, red colored nodes indicate the nodes that are included in community 1 and circle 1. Yellow nodes include the ones that do not have circle tags and that are included in other circles. Moreover, it can be seen that the nodes that are both in community 1 and circle 3 tend to cluster in the core side of the network. This trend can also be seen in Figure 7.4, Figure 7.5 and Figure 7.6 where the ego node is the 12th node.

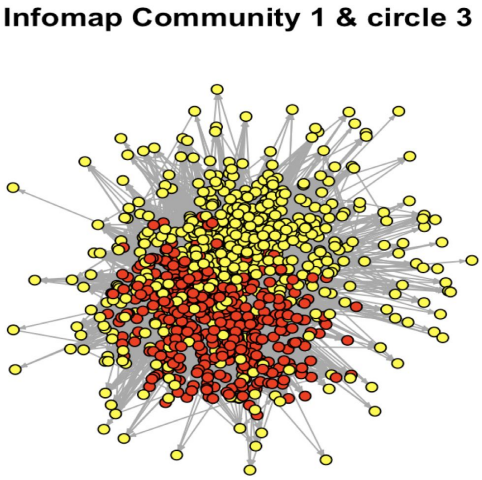

Figure 7.3 Network of nodes that are in both community 1 of Infomap and circle 3

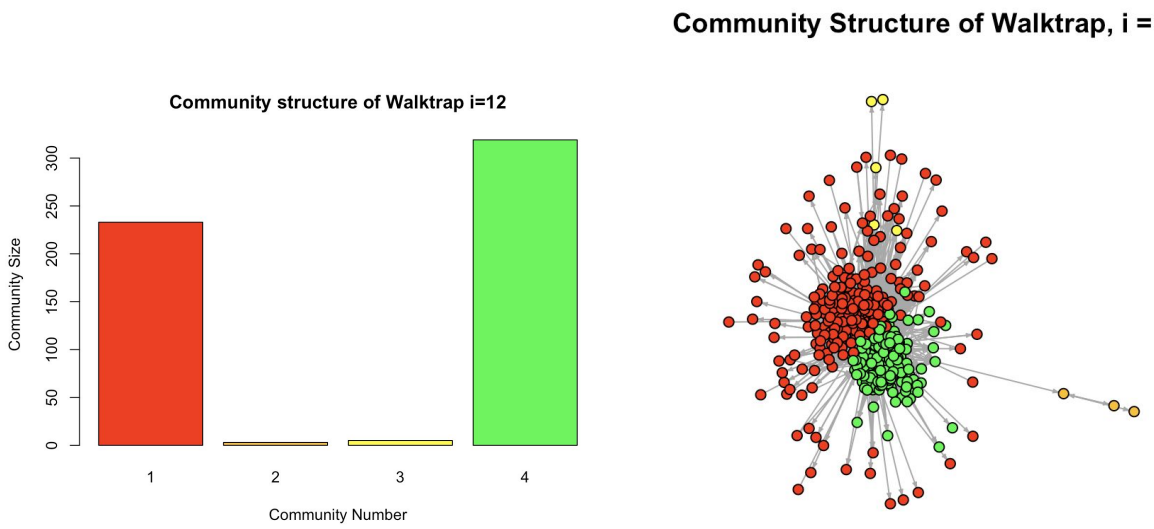

Figure 7.4 Community structures of 12th ego node, "100962871525684315897", extracted from Walktrap algorithm

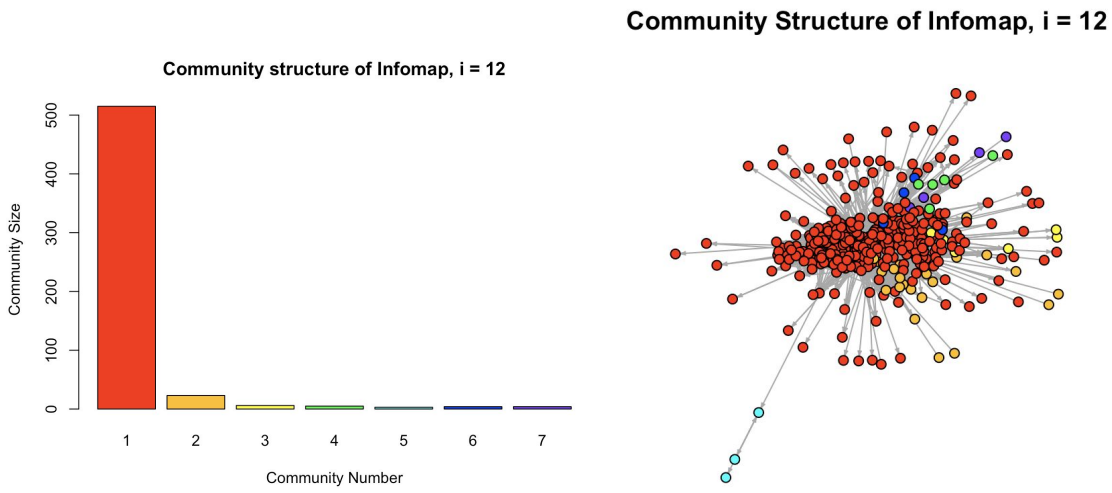

Figure 7.5 Community structures of 12th ego node, "100962871525684315897", extracted from Walktrap algorithm (12th ego node)

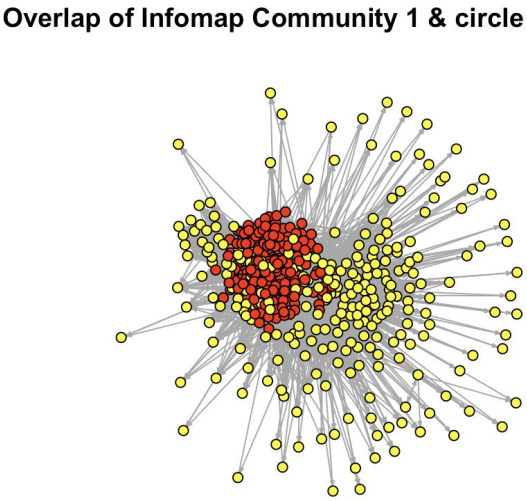

Figure 7.6 Network of nodes that are in both community 1 of Infomap and circle 1 (12th ego node)

To conclude, the membership of node matches well with the circle group it belongs to. However, as discussed in the previous homework, community structure extraction algorithms assign only one membership per node. When the nodes are tagged with multiple circle groups, it is hard to see the distinction between communities. In this case, community extraction algorithm becomes less accurate. Moreover, not all the nodes are tagged in circles by users. If a user tends to skip tagging relationships, the percentage the community structure and circles overlap would be low even though the algorithms perform accurately. Thus, across different users, communities and circles overlap well when users have the habit of tagging most of their friends and with one relationship tag.

III. Conclusion

In this project, we have analyzed communities and friendship structures on social networks and demonstrated their applications for mining the data for high levels of information. We also explored metrics for differentiating these communities into identifiable, real world communities such as college friend groups and family groups. Dispersion and embeddedness provide insight into the connections that can aid in the identification of romantic partners, school networks, and family members. Furthermore, Google + data is given to analyze how well the community structure algorithms can detect circles. We can see how users' tagging habit affect how the circles overlap with communities. A user can tag friends in multiple circles or none, while the community algorithms strictly assign one membership per each node.