Graphs and Network Flows
Project 1

Mansee Jadhav - 204567818
Ronak Sumbaly - 604591897

May 15, 2016

Introduction

In this project we study social networking and graphs of user's personal friendship network. We have tried to interpret community structures in the friendship network and their various graphical applications. Real social networks of Facebook and Google+ have been used through the project for analysis. The results for each of the questions are provided below.

Question 1.

The Facebook graph edgelist was considered which consists of 'circles' (or 'friends lists') from Facebook. The dataset comprises of edges from all egonets combined. An undirected network was created for this dataset using the `read.graph` function. Further analysis of the network is provided below,

1.1 Network Connectivity & Diameter

The created network was checked for connectivity using `is.connected()` function and using the `diameter()` function the diameter of the network was calculated.

Connected = TRUE
Diameter = 8

1.2 Degree Distribution

The degree distribution (frequency & density) of the network were plotted. The average diameter of the network was calculated at the same time using the `mean(degree(network))` function.

Average Diameter= 43.69101

1.3 Fit a Curve to Degree Distribution

In order to fit a curve to this distribution we considered multiple models according to the shape of the distribution (exponential in shape). To plot these models `ggplot2` was used and to calculate the non-linear least square estimates of the model `nls()` function was used. *Fig 2.* below shows models vs. the distribution fit.

The degree distribution of the network is shown below,

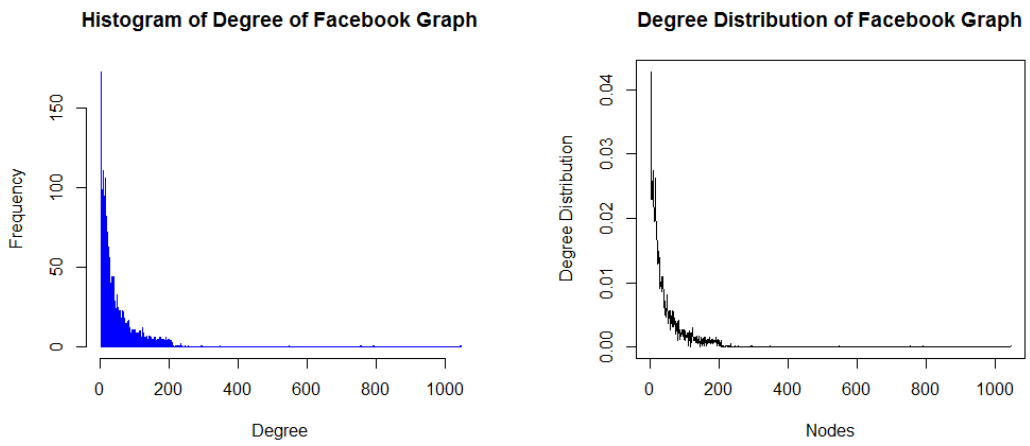

Figure 1: Histogram and Degree Distribution of Facebook Network

The models considered for fitting were,

$$y \approx \frac{1}{x * a} + b * x \qquad \texttt{blue} \tag{1}$$

$$y \approx a + b * log(x) \qquad \texttt{yellow} \tag{2}$$

$$y \approx e^{a+b*x} \qquad \texttt{red} \tag{3}$$

$$y \approx \frac{1}{x * a} + b \qquad \texttt{purple} \tag{4}$$

$$y \approx e^{1^{a+b*x}} \qquad \texttt{orange} \tag{5}$$

The model curve with the degree distribution is plotted below,

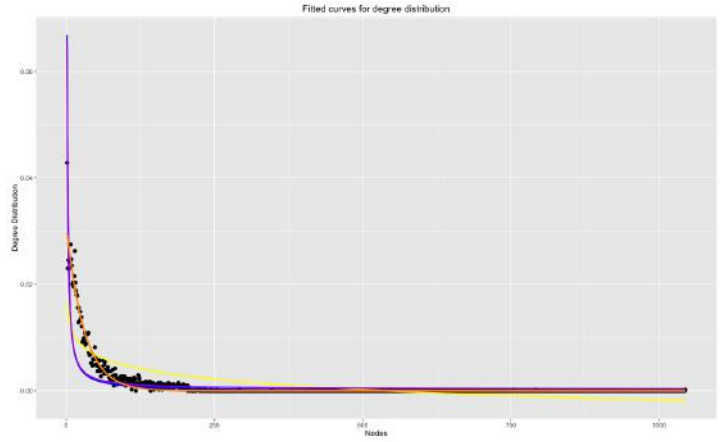

Figure 2: Fitting curves for degree distribution

The graph clearly shows that *Eqn. 5* fits the best for the degree distribution. The error estimates of the model were as follows **Residual standard error** = 0.0007093 & **Mean Squared Error** = 0.0000005031

Question 2.

The **personal network** of a node is the graph comprising of the nodes and its neighbors, having edges that have both end within this set of nodes. We employ this definition to the following question.

2.1 Neighbors of Node `ID-1`

The first node of the Facebook network with `ID-1` was chosen and all its neighbors were found using `neighbors(network, node)` function, where `network = Facebook network` that was initially constructed and `node = 1`.

2.2 Personal Network of Node `ID-1`

After obtaining the neighbors of Node 1 we created itś personal network by connecting the set of nodes obtained above with edges that have both ends within this set of nodes, using `induced.subgraph(network, node_set)` function, where the `network = original Facebook network` and `node_set = (1, Neighbors of 1)`.

`induced.subgraph()` function renames the vertices from 1 to n, where n is # of nodes present. The numbering is done in increasing of the original names of nodes. Hence in-order to preserve names of original node, we sorted the list of neighbors of Node 1 by it's names and stored them as an attribute of the personal network : `personal_network$names`.

Details of personal network of Node 1 are as follows,

Number of Nodes = 348
Number of Edges = 2866

Personal network of Node `ID-1` **(Node 1 = Red Node, Neighbors = Light Blue)**

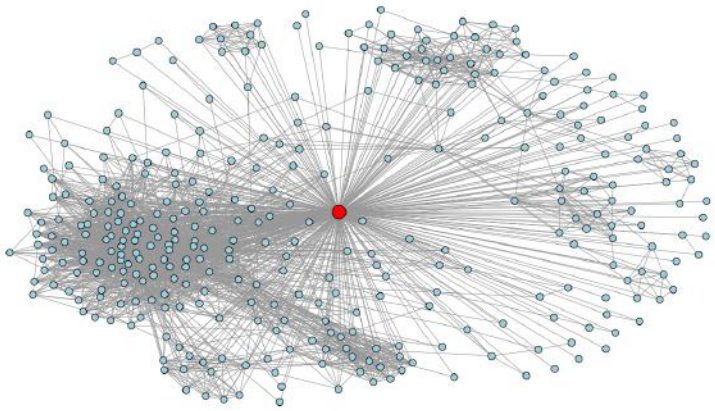

Figure 3: Personal Network of Node ID-1

Question 3.

The **core nodes** of a network is defined as the **nodes which have more than 200 neighbors**. This concept of personal network and core node is applied in the following question.

3.1 Core Nodes

The core nodes of the Facebook network was found by simply looping through each node in the network and checking the number of neighbors using the `neighbors()` function. Degrees of each of the core nodes is found simultaneously using the `degree()` function and the mean is simply calculated using `mean(degree[core_nodes])`. Results obtained were as follows,

Number of Core Nodes = 40
Average Degree of Core Nodes = 279.375

It is evident that the **average degree is greater than 200** as we are only considering those nodes that have more than 200 neighbors.

3.2 Personal Network Node `ID-2048`

For further analysis we chose one of the core nodes. As computer scientist we chose **2048** as our reference node. Using the methodology of *Ques. 2* we construct the **personal network of Node 2048**. The network and its details are provided below,

Personal Network `ID-2048` :
Number of Nodes = 206
Number of Edges = 6611

Personal network of Node `ID-2048` **(Node 2048 = Red Node, Neighbors = Light Blue)**

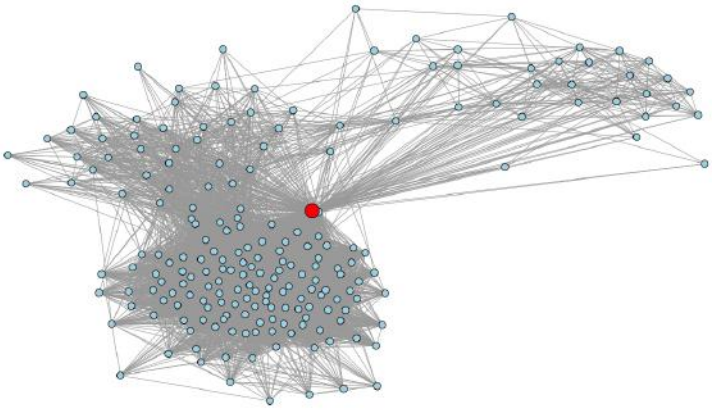

Figure 4: Personal Network of Node ID-2048

3.3 Community Detection Algorithms

The personal network for **Core Node 2048** constructed above was used to find community structures using different community detection algorithms. The results for the algorithms are presented below.

3.3.1 Fast-Greedy Community Detection

The **fast-greedy community detection algorithm** was applied on the personal network. **3 communities** were obtained for the **personal network (PN) of Core Node 2048** as shown in below graphs.

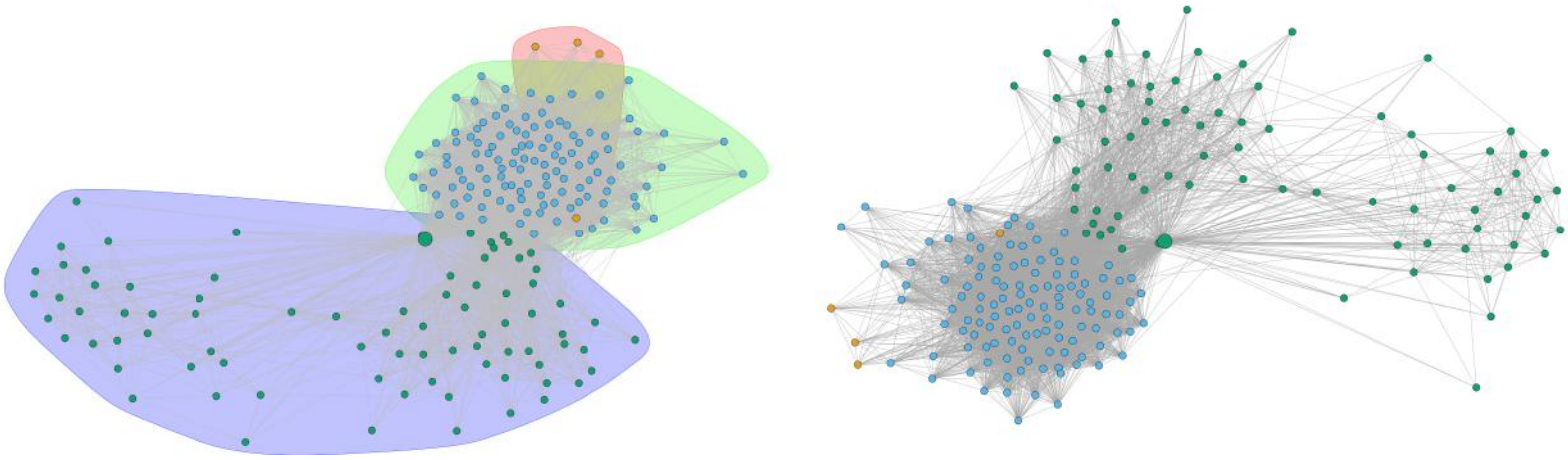

Figure 5: Community Structure of the Core-Node 2048 PN - Fast Greedy Algorithm

The 3 communities which were found are distinguished by blue, green and orange colors. The **Core Node 2048** is shown with a larger size. Community structure is provided below

Community Number	Community Size
1	4
2	120
3	82

Table 1: Community Structure - Fast Greedy Algorithm

3.3.2 Edge-Betweenness Community Detection

The **edge-betweenness community detection algorithm** was applied on the personal network. **57 communities** were obtained for the **personal network (PN) of Core Node 2048** as shown in below graphs.

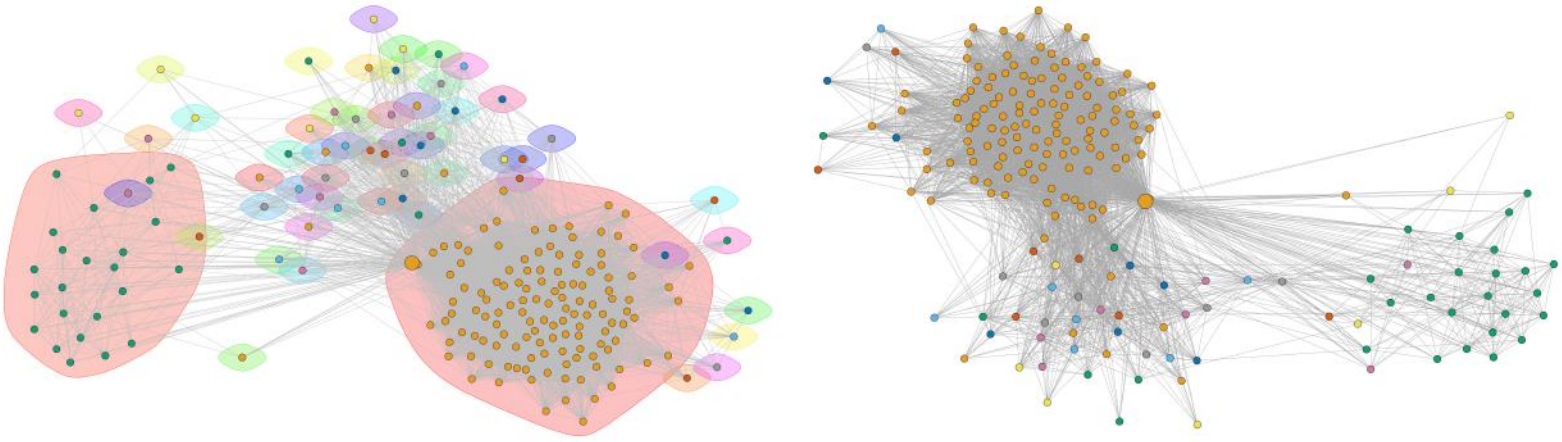

Figure 6: Community Structure of the Core-Node 2048 PN - Edge-Betweenness Algorithm.

Community Structure is provided below,

#	Size	#	Size	#	Size	#	Size
1	128	8	1	15	1	22	1
2	1	9	1	16	1	23	1
3	23	10	1	17	1	24	1
4	1	11	1	18	1	25	1
5	1	12	1	19	1	26	1
6	1	13	1	20	1	27	1
7	1	14	1	21	1	28	1
29	1	36	1	43	1	50	1
30	1	37	1	44	1	51	1
31	1	38	1	45	1	52	1
32	1	39	1	46	1	53	1
33	1	40	1	47	1	54	1
34	1	41	1	48	1	55	1
35	1	42	1	49	1	56	1
57	1	-	-	-	-	-	-

Table 2: Community Structure - Edge-Betweenness Algorithm

3.3.3 Infomap Community Detection

The **infomap community detection algorithm** was applied on the personal network. **3 communities** were obtained for the **personal network (PN) of Core Node 2048** as shown in below graphs.

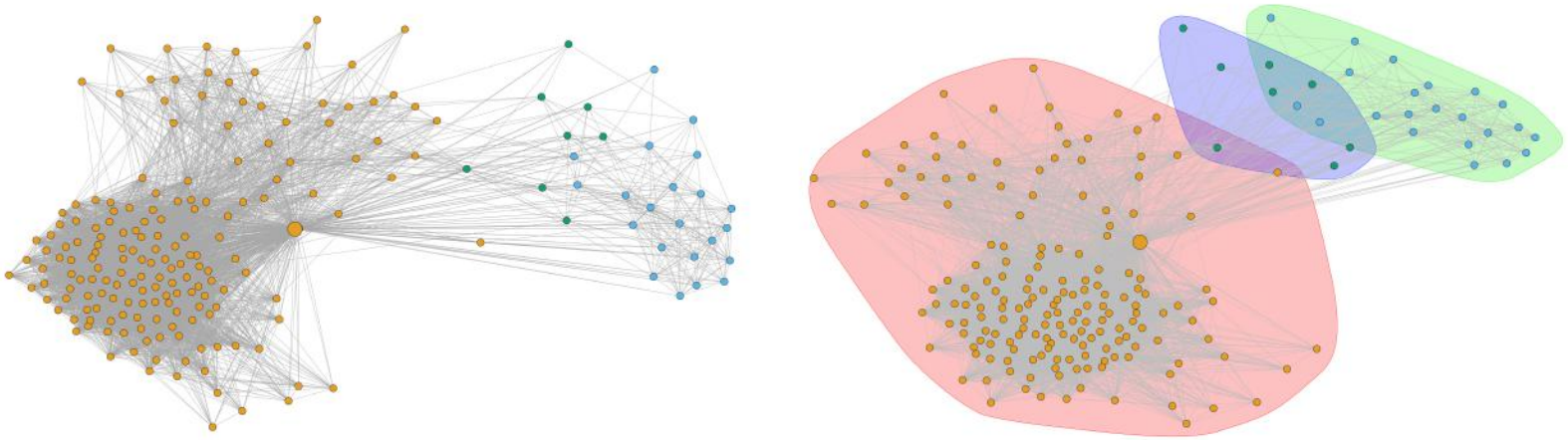

Figure 7: Community Structure of the Core-Node 2048 PN - InfoMap Algorithm

The community structure is provided below,

Community Number	Community Size
1	176
2	22
3	8

Table 3: Community Structure - Infomap Algorithm

3.4 Analysis of Results

A clear observation from the graphs is that all 3 community detection algorithms operate in different manners. Further analysis that can be made are as follows,

1. The **edge-betweenness algorithm** tends to break the graph into **more communities** as compared to the other two algorithms.
2. An apparent overlap in features can be seen in the community structure wherein the **core node 2048** appears in the community with the maximum number of nodes.
3. Since edge-betweenness is a divisive algorithm the high number of communities can be attributed to the **high modularity of the network** and the functioning of the algorithm.

Question 4.

In order to observe the importance of the core node we remove the core node from the personal network and perform the above community detection once again.

4.1 Personal Network without Core Node 2048

The personal network was again constructed without the Core Node 2048. The network and its details are provided below,

Personal Network `ID-2048` :
Number of Nodes = 205
Number of Edges = 6406

The difference in edges is same as the number of neighbors that the core node 2048 had. The new personal network can be visualized below. Personal network **without** Node `ID-2048`

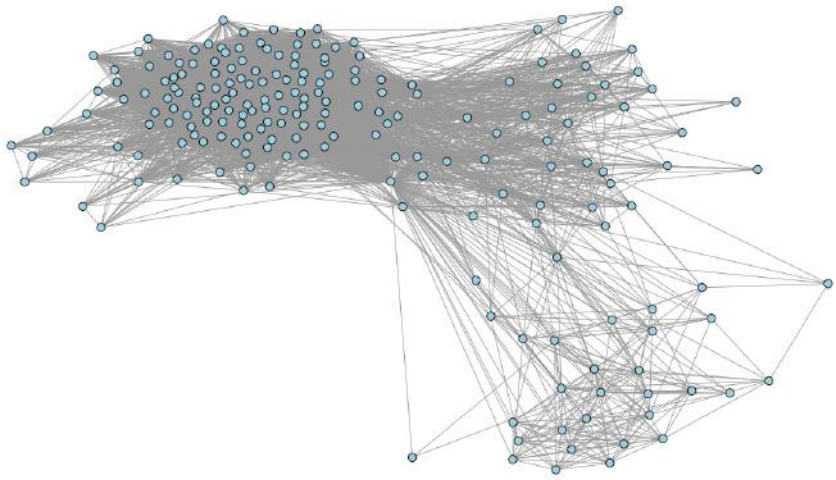

Figure 8: Personal Network without Node ID-2048

4.2 Community Detection Algorithms

The modified personal network for **Core Node 2048** constructed above was used to find community structures using different community detection algorithms. The results for the algorithms are presented below.

4.2.1 Fast-Greedy Community Detection

The **fast-greedy community detection algorithm** was applied on the modified personal network. **2 communities** were obtained for the **personal network (PN) without Core Node 2048** as shown in below graphs.

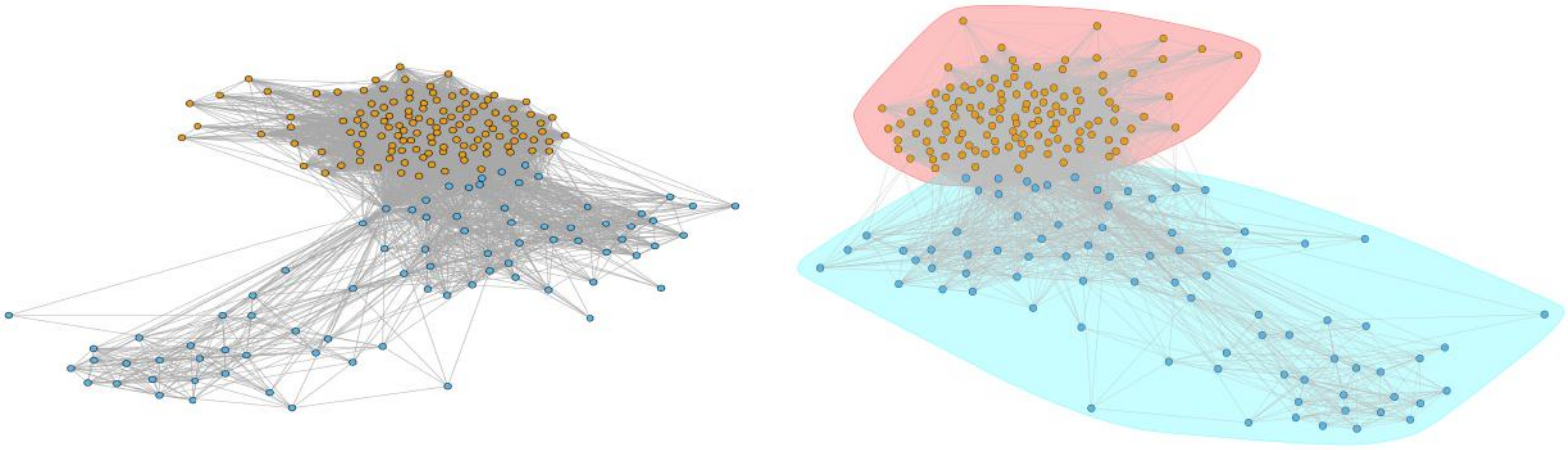

Figure 9: Community Structure of the Modified PN - Fast Greedy Algorithm

The 2 communities which were found are distinguished by blue and orange colors. Community structure is provided below

Community Number	Community Size
1	124
2	81

Table 4: Community Structure - Fast Greedy Algorithm

4.2.2 Edge-Betweenness Community Detection

The **edge-betweenness community detection algorithm** was applied on the modified personal network. **54 communities** were obtained for the **personal network (PN) without the Core Node 2048** as shown in below graphs.

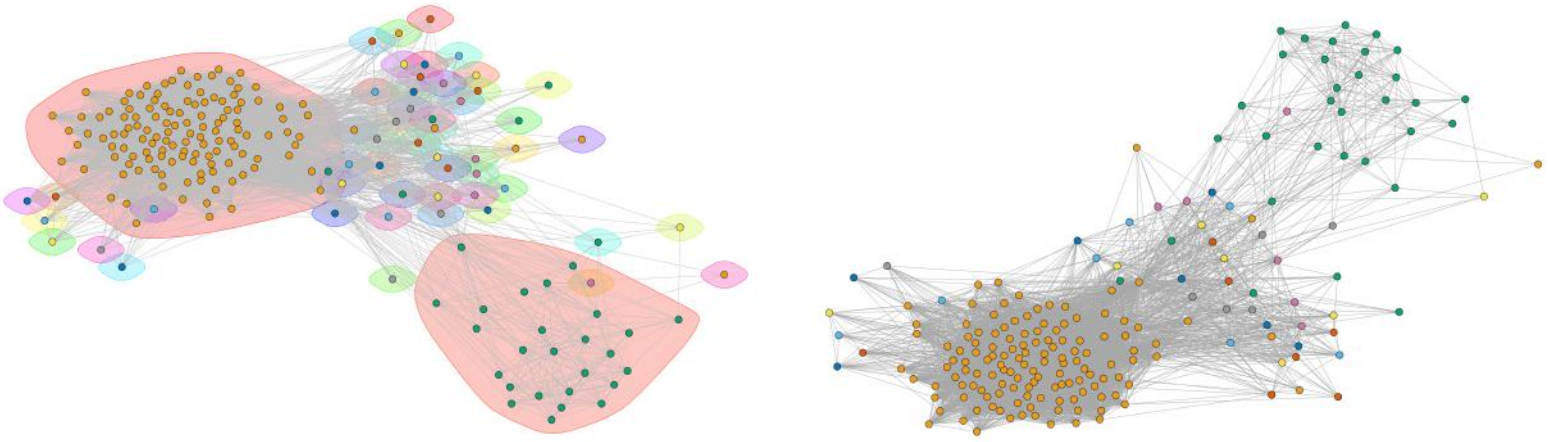

Figure 10: Community Structure of the Modified PN - Edge-Betweenness Algorithm.

The community structure obtained is provided below,

#	Size	#	Size	#	Size	#	Size
1	127	8	1	15	1	22	1
2	1	9	1	16	1	23	1
3	26	10	1	17	1	24	1
4	1	11	1	18	1	25	1
5	1	12	1	19	1	26	1
6	1	13	1	20	1	27	1
7	1	14	1	21	1	28	1
29	1	36	1	43	1	50	1
30	1	37	1	44	1	51	1
31	1	38	1	45	1	52	1
32	1	39	1	46	1	53	1
33	1	40	1	47	1	54	1
34	1	41	1	48	1	-	-
35	1	42	1	49	1	-	-

Table 5: Community Structure - Edge-Betweenness Algorithm

4.2.3 Infomap Community Detection

The **infomap community detection algorithm** was applied on the modified personal network. **3 communities** were obtained for the **personal network (PN) without the Core Node 2048** as shown in below graphs.

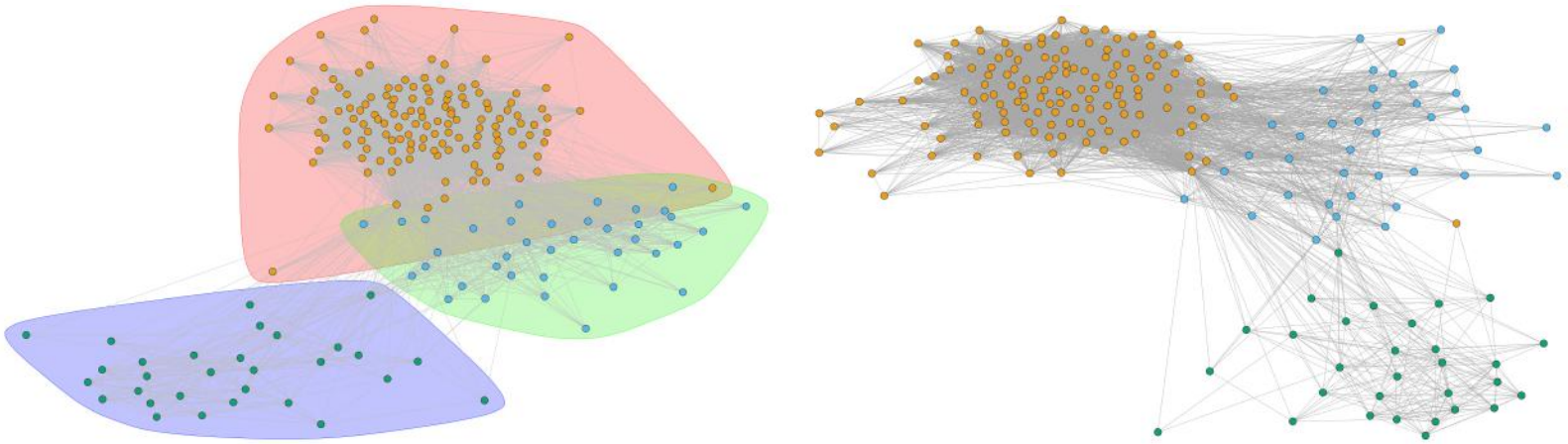

Figure 11: Community Structure of the Modified PN - InfoMap Algorithm

The community structure is provided below,

Community Number	Community Size
1	137
2	38
3	30

Table 6: Community Structure - Infomap Algorithm

4.3 Analysis of Results

The analysis of the results is provided below,

1. Comparing the results with *Ques. 3* it can be seen that removing the core node decreases the number of communities for the fast greedy and edge-betweenness algorithms but not for the infomap algorithm.

2. The nodes in the infomap algorithm are distributed amongst the communities thereby keeping the count same as before.

3. Even though the number of communities have decreased the modularity of each community algorithm remains almost the same. This shows that the individual communities still have the same distribution of nodes.

Question 5.

This question deals with two features of the personal network namely, **Embeddedness and Dispersion**.

- **Embeddedness** is the number of mutual friends a node shares with the core nodes, which means the larger the embeddedness is, the more mutual friends a node will have. It is calculated using the `intersect()` function which takes the common mutual neighbors between a node and its core-node.
- **Dispersion** is the sum of distances between every pair of the mutual friends a node shares with core node. It basically is calculating the relationship among all of a node's mutual friends, which means the larger the dispersion is, the more likely a node's mutual friends aren't acquainted with each other.
- **Dispersion** on the other hand is calculated by making all pairs of the mutual friends between a node and the core-node using `comb()` function and then getting the shortest path between these nodes within the network not containing the present node in question and the core-node. This is done using `shortest.paths()` function. The sum of the distances is dispersion of the node w.r.t the core-node.

5.1 Distribution of Embeddedness & Distribution

Using the above method the distribution of the embeddedness and distribution was plotted as follows,

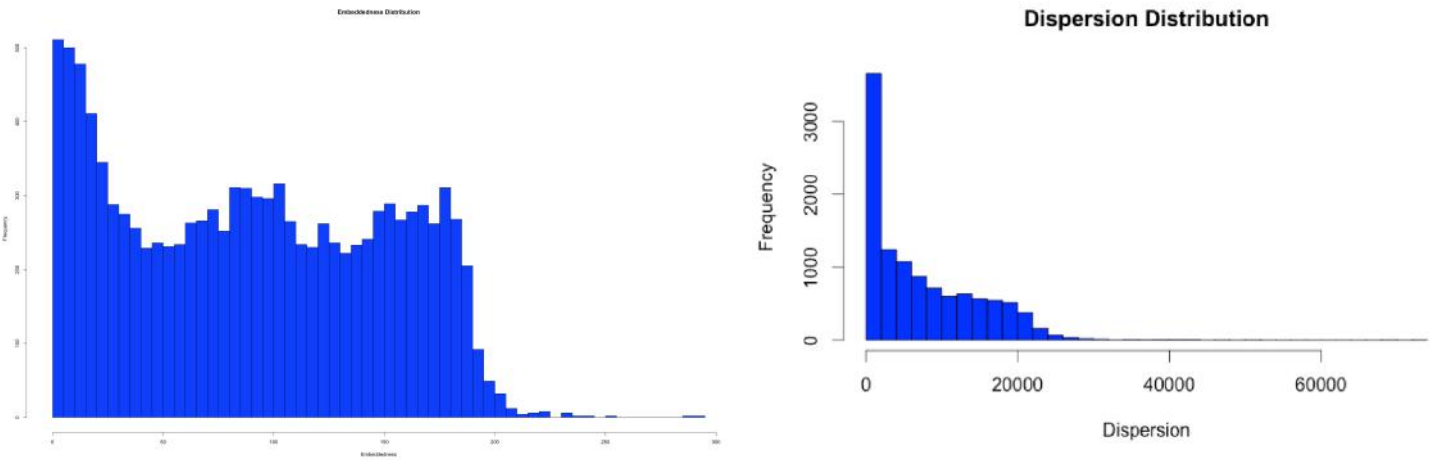

Figure 12: Distribution of Embeddedness (Left) & Dispersion (Right)

5.2 Analysis of Personal Networks

In this section we select 3 core-nodes for construction of personal networks and analyze their dispersion and embeddedness. We select `core-node ID-1, 349, 2612`. The results for each feature and personal network is provided below,

5.2.1 Maximum Dispersion

The dispersion was computed for the personal networks of the 3 core-nodes. The **maximum dispersion** value was found for each personal network along with its corresponding node (neighbor of core-node). Since sometimes during dispersion calculation two nodes might not be connected giving a dispersion value of `Inf`, in such a cases these values are ignored and only numeric values are considered in this problem.

The edges incident to the **maximum dispersion is highlighted in red** as well as the **node is enlarged and highlighted in pink**. The communities of the network are represented in different colors. Following are the graphs obtained,

Core-Node `ID-1` (Left) & `ID-349` (Right)

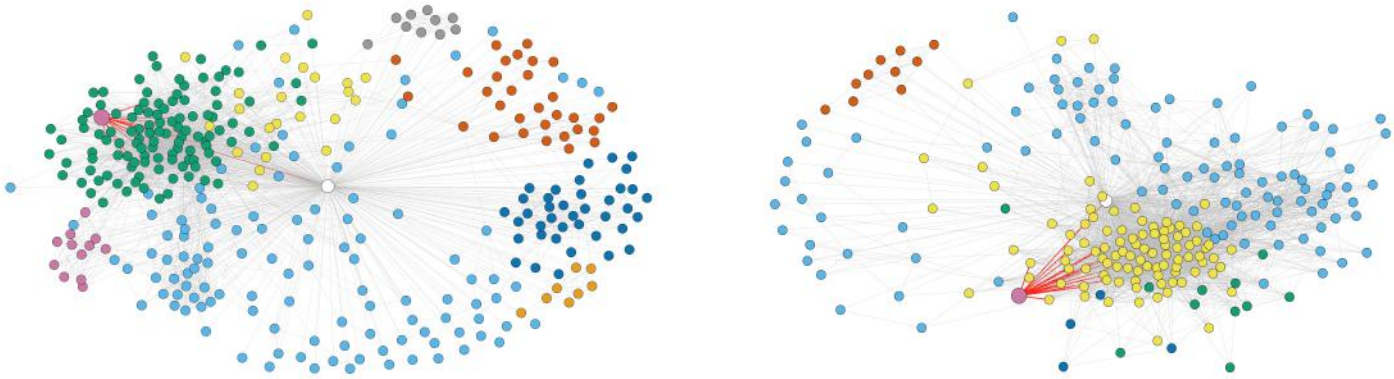

Figure 13: Maximum Dispersion Network: Node ID-1 (Left) & ID-349 (Right)

Core-Node `ID-2612`

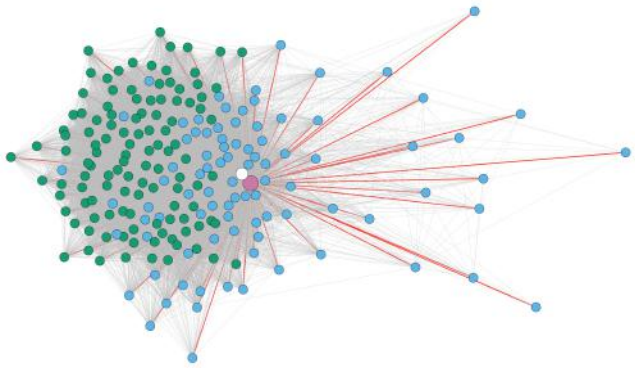

Figure 14: Maximum Dispersion Network: Node ID-2612

5.2.2 Maximum Embeddedness

Similar to the dispersion above, the embeddedness was computed for the personal networks of the 3 core-nodes. The **maximum embeddedness** value was found for each personal network along with its corresponding node (neighbor of core-node). The edges incident to the **maximum embeddedness is highlighted in red** as well as the **node is enlarged and highlighted in pink**. The communities of the network are represented in different colors. Following are the graphs obtained,

Core-Node `ID-1` **(Left) &** `ID-349` **(Right)**

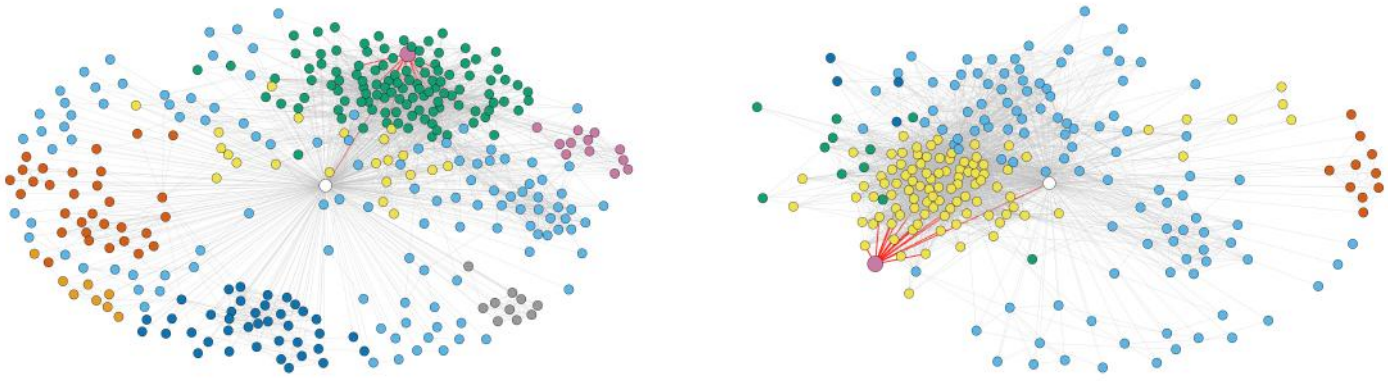

Figure 15: Maximum Embeddedness Network: Node ID-1 (Left) & ID-349 (Right)

Core-Node `ID-2612`

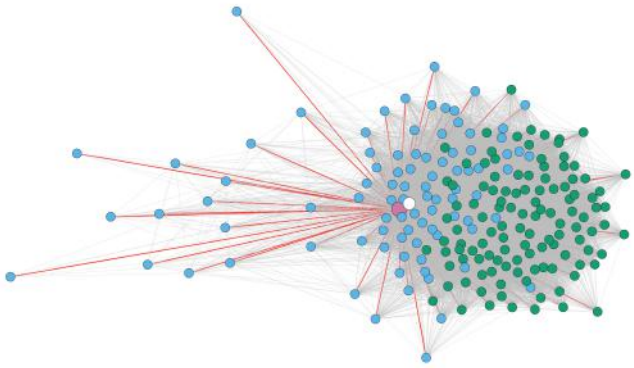

Figure 16: Maximum Embeddedness Network: Node ID-2612

5.2.3 Maximum $\frac{Dispersion}{Embeddedness}$

The fraction of $\frac{Dispersion}{Embeddedness}$ was computed for each personal network and the maximum value in the personal network was found. The results are shown below,

Core-Node `ID-1` **(Left) &** `ID-349` **(Right)**

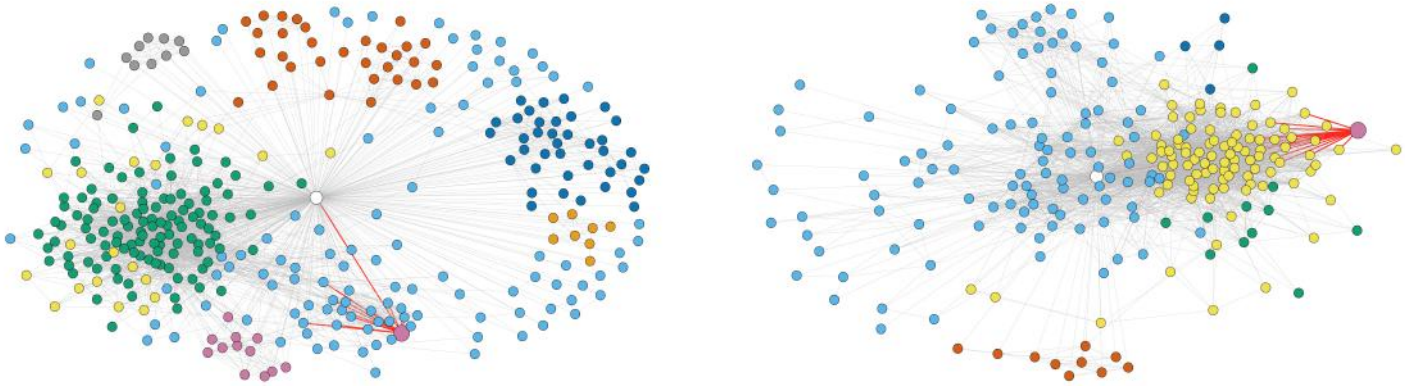

Figure 17: Maximum $\frac{Dispersion}{Embeddedness}$ Network: Node ID-1 (Left) & ID-349 (Right)

Core-Node `ID-2612`

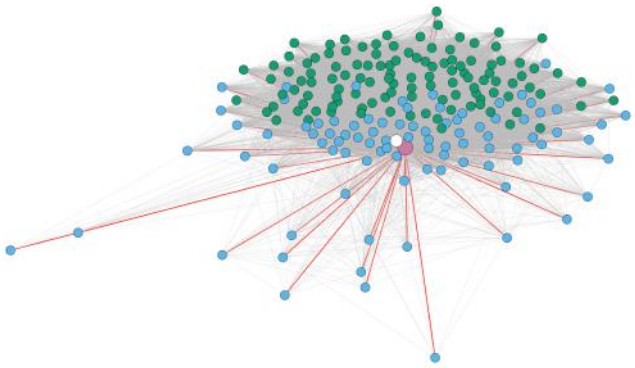

Figure 18: Maximum $\frac{Dispersion}{Embeddedness}$ Network: Node ID-2612

5.3 Analysis of Results

- **Dispersion**
  The highlighted pink node, shares maximum dispersion with core node. It means that the core-node is **very less acquainted** with the node in pink. Mutual neighbors of core-node and the pink node are not very well connected to each other. Also it is seen that dispersion decreases when core-node's network size increases.

- **Embeddedness**
  As seen from figure 15 and 16, the considered white node has maximum embeddedness with the pink node, which belongs to white node's largest community. This reveals that, if the size of a community within personal network of a core-node is maximum, then there exists a pink node within that community which shares maximum number of mutual friends with this core-node. Thus it measures how much of direct neighbors of the core-node belong to the same community.

- $\frac{Dispersion}{Embeddedness}$

  A high dispersion to embeddedness ratio perform well to predict the correct romantic relationship measure. One definition of `dispersion` is

  > The number of pairs `(s,t)` of mutual friends of u and v such that `(s,t)` have no common neighbors except for u and v.

  Thus the two communities are linked to each other only via these common nodes i.e the core node and pink node. Thus the core-node and pink node might be in romantic relationship.

Question 6.

Every person has communities within his/her personal network. This question deals with analyzing structural features, to map similar communities among different personal networks of core nodes. This communities can be classmates, organization based or family members etc. We try to find closeness among members using three features- `Clustering coefficient, Density of node and its Community size`.

6.1 Structural Features of Communities

In order to analyze the similarity between personal networks and communities we compute the following structural networks,

- `Clustering coefficient:` It is a measure of the degree to which nodes in a graph tend to cluster together. It measures the closeness among neighbors of a node.

$$\textbf{Clustering coefficient} = \frac{2 * \texttt{(Number of direct links between neighbors of vertex V)}}{\texttt{(degree of node V) * (degree of node V - 1)}}.$$

- `Density:` It is the ratio of the number of edges and the number of possible edges.
- `Community size:` It represents the sizes of each communities within each personal network. Using this features we measure closeness of communities within personal network. Results and analysis are provided in the following sections.

6.2 Similarity among Communities

For determining features of communities to classify them into types like **"classmates" or "families"** we used the above structural features. In order to differentiate between a **Type 1 and Type 2** we look at the maximum and minimum value of these structural features. Only communities with greater than 10 nodes were considered for analysis.

The **maximum values** for the structural features indicate the communities are **closely connected** and they share same common type. For example, all communities with similar ranging clustering coefficient **share more closeness with each other**. Thus such communities can be of type business, best friends, classmates etc. **High density values** consider present as well as **possible future relationships**. Thus in a community of school mates, two students who are not friends yet, share lot of mutual friends and have large density value and hence belong to same community. The results obtained for each personal network of the core-node is given below,

Core-Node	Clustering Coefficient	Density	Community Size
1	0.0073, **0.023**, 0.009, 0.003, 0.0015, 0.001	0.0671, 0.217, 0.209, 0.237, 0.181, **0.718**	114, 112, 22, 39, 31 12
2	0.0157, **0.0319**, 0.0011	0.0794, 0.148, **0.252**	464, **484**, 70
3	0.0300, 0.0008, **0.06657**	0.135, 0.348, **0.358**	**107**, 11, 98
4	**0.0469**, 0.0429, 0.0539	**0.487**, 0.434, 0.358	71, 72, **89**
5	0.0983, **0.131**	**0.619**, 0.350	81, **125**
6	0.133, **0.154**	0.520, **0.605**	**109**, 109
7	0.0028, 0.116, **0.121**	**0.658**, 0.651, 0.441	15, 98, **122**
8	**0.178**, 0.105	0.483, **0.649**	**133**, 88
9	0.103, **0.164**	**0.657**, 0.433	83, **129**
10	0.0108, 0.102, **0.141**	**0.698**, 0.643, 0.586	25, 81, **100**
11	**0.174**, 0.102	0.493, **0.589**	**139**, 97
12	**0.021**, 0.003, 0.018, 0.00043	0.097, 0.073, **0.218**, 0.715	**371**, 165, 226, 19
13	0.105, **0.138**	**0.531**, 0.499	100, **118**
14	**0.202**, 0.106	**0.607**, 0.564	**116**, 87
15	**0.151**, 0.107	**0.558**, 0.482	**108**, 98
16	0.002, **0.168**, 0.0787	**0.626**, 0.532, 0.505	13, **137**, 96
17	**0.208**, 0.0800	**0.568**, 0.550	**121**, 76
18	0.0562, 0.0493, **0.0976**	**0.651**, 0.620, 0.513	74, 71, **110**
19	0.0568, 0.0370, 0.0035, 0.0050	**0.588**, 0.213, 0.109, 0.549	234, **314**, 136, 72
20	**0.189**, 0.169	**0.826**, 0.596	106, **118**
21	0.0401, 0.0087, **0.295**	0.414, 0.789, **0.827**	69, 23, **133**
22	**0.227**, 0.168	**0.776**, 0.755	**109**, 95
23	**0.216**, 0.0450	**0.623**, 0.276	**120**, 82
24	0.0118, **0.161**, 0.151	0.816, **0.833**, 0.730	24, 89, **92**
25	**0.189**, 0.170	**0.799**, 0.749	**98**, 96
26	0.098, **0.265**	0.504, **0.820**	97, **125**
27	**0.176**, 0.144, 0.008	0.722, **0.827**, 0.800	**103**, 87, 21
28	**0.311**, 0.107	**0.813**, 0.694	126, 80
29	**0.247**, 0.136	**0.818**, 0.637	113, 95
30	0.136, **0.212**	0.577, **0.767**	107, **116**
31	**0.166**, 0.011, 0.145	**0.814**, 0.780, 0.748	**90**, 24, 88
32	0.0836, **0.246**	0.460, **0.814**	99, **128**
33	**0.286**, 0.102	**0.759**, 0.656	**178**, 114
34	0.129, **0.260**	0.660, **0.803**	91, **117**
35	0.004, **0.197**, 0.156	0.790, 0.776, **0.812**	14, **101**, 88
36	0.173, **0.227**	0.732, **0.820**	97, **105**
37	**0.224**, 0.126	**0.778**, 0.565	**157**, 138
38	0.0031, 0.148, **0.182**	0.795, 0.653, **0.804**	12, 95, 95
39	0.116, **0.266**	0.577, **0.836**	92, **116**
40	0.00067, 0.008, **0.016**, 0.0013, 0.0008, 0.0009, 0.004	0.476, 0.135, 0.111, 0.214, **0.673**, 0.177, 0.205	20, 137, **208**, 42, 19, 39, 76

Question 7.

We now consider another real social network with tagged relationship: **Google+ ego networks**. The analysis of the network is provided in the sections below,

7.1 Dataset Details

Google+ ego network was downloaded and the unique ids from the file list were obtained which indicated the number of ego nodes.

$$\textbf{Total Unique IDs / Ego Nodes} = 132$$

7.2 Ego Nodes with $>$ 2 circles

For each unique id the dataset comprises of a ***.circle** with contains the circles of the ego node. Each line in the file corresponds to a single cirle. We looped through each file and read the number of lines in the file. If the file contained line count ≥ 3 we stored those ego nodes for further consideration.

$$\textbf{Number of Ego Nodes} > \textbf{2 circles} = 57$$

7.3 Community Structure

For each of the ego nodes found in *Section 7.2* a personal network was created wherein the edges from the ego node to every other node in the ***.edges** file was made. We created communities of this personal network using the `walktrap.community` and `infomap.community` algorithm. To illustrate the community structure a random node was chosen from among the **57 nodes**.

Community Structure for Ego Node ID = 100535338638690515335

7.3.1 Walktrap Community

The community obtained using the walktrap community detection algorithm is presented below,

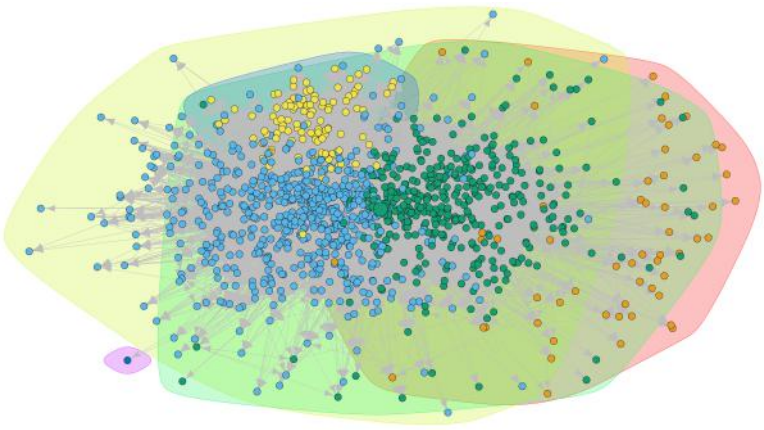

Figure 19: Walktrap Community

Community Structure is as follows,

Community #	Freq
1	56
2	628
3	389
4	114
5	1

Table 7: Walktrap Community Structure

7.3.2 Infomap Community

The community obtained using the infomap community detection algorithm is presented below,

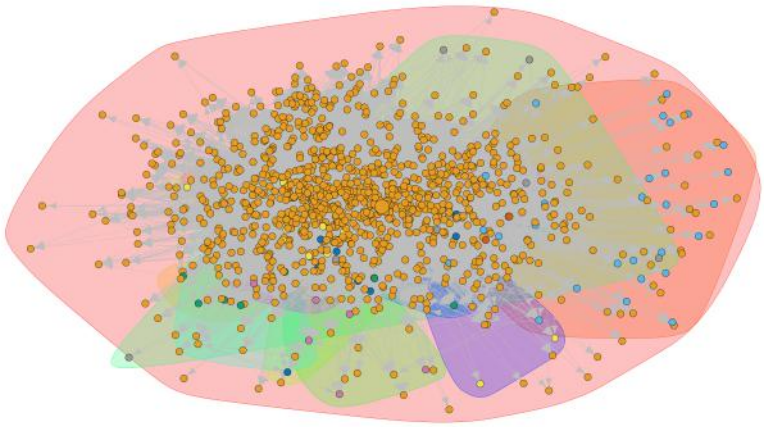

Figure 20: Infomap Community

Community Structure is as follows,

Community #	Freq	Community #	Freq
1	1,109	9	3
2	26	10	3
3	5	11	2
4	9	12	3
5	6	13	2
6	2	14	2
7	5	15	2
8	5	16	2
17	2	-	-

Table 8: Infomap Community Structure

7.4 Overlap between User's Circle & Communities

In order to check for the overlap between a user's circle and communities we just check for the percentage of intersection of nodes between a circle and a community with respect to the circle.

$$\textbf{\% of Nodes B/t Community \& Circle} = \frac{\#\ \texttt{of common nodes}}{\#\ \texttt{of nodes in the circle}}$$

We loop through each user and check for the comparison between the overlap. For analysis purpose we choose three ego nodes *(start, middle and finish)*. The plots below show the degree of overlap. **Darker the color more** is the overall **overlap** between the community and circle

7.4.1 Walktrap Community

Overlap for Walktrap community is presented in the below plots,

`Node 1 & 2`

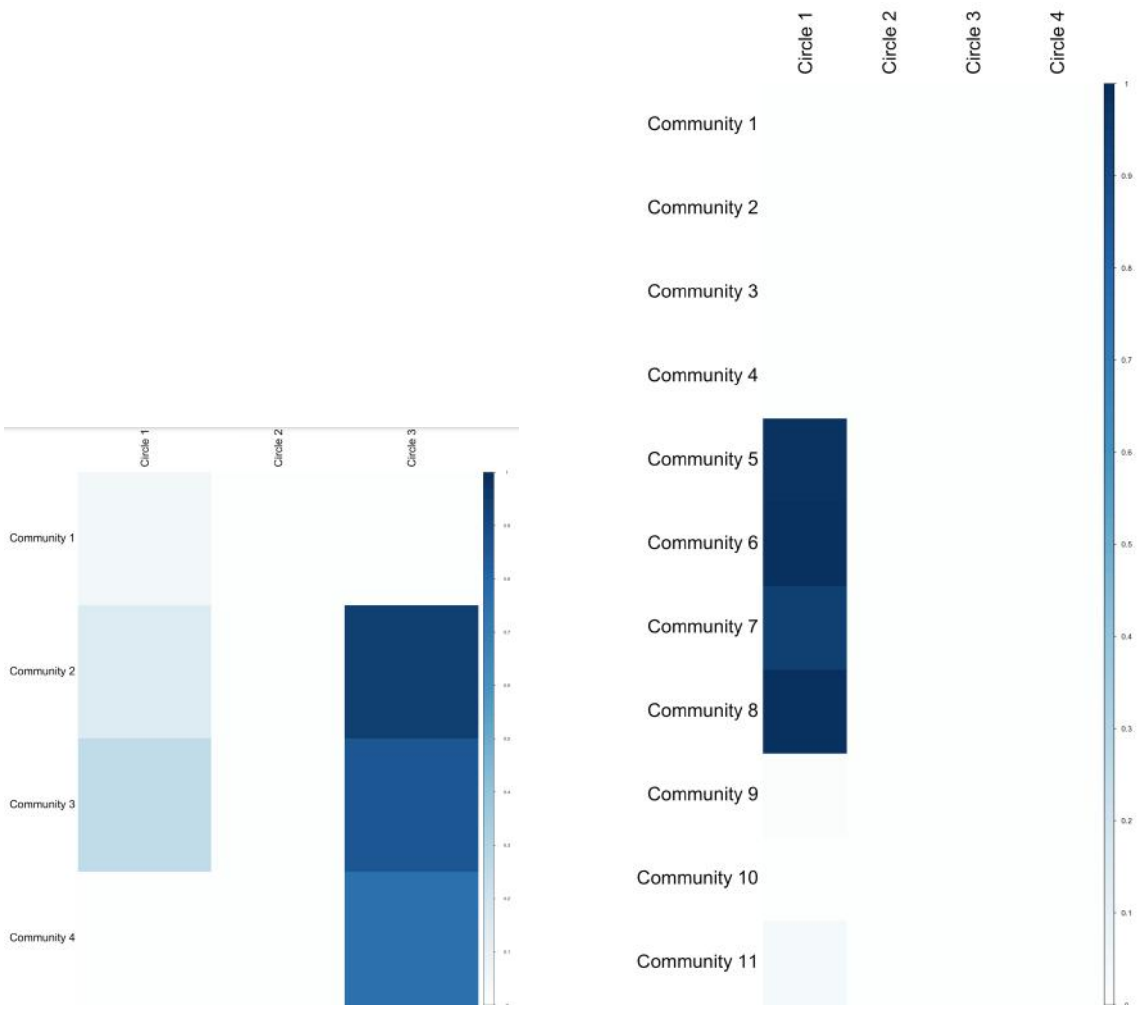

Figure 21: Walktrap Community Overlap Node - 1 & 2

`Node 3`

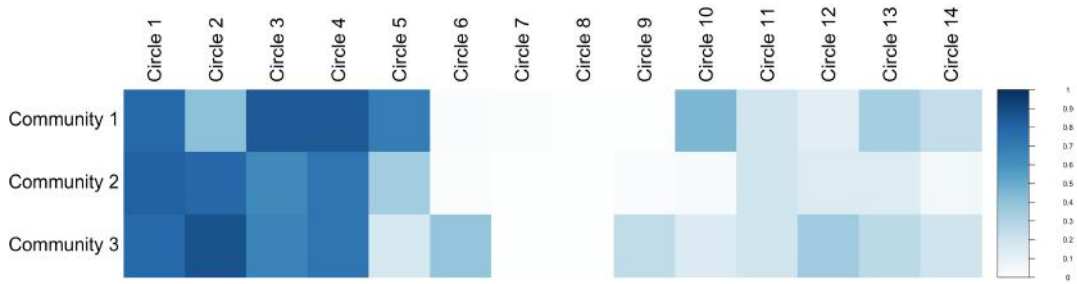

Figure 22: Walktrap Community Overlap Node - 3

7.4.2 Infomap Community

Overlap for Infomap community is presented in the below plots,

`Node 1 & 2`

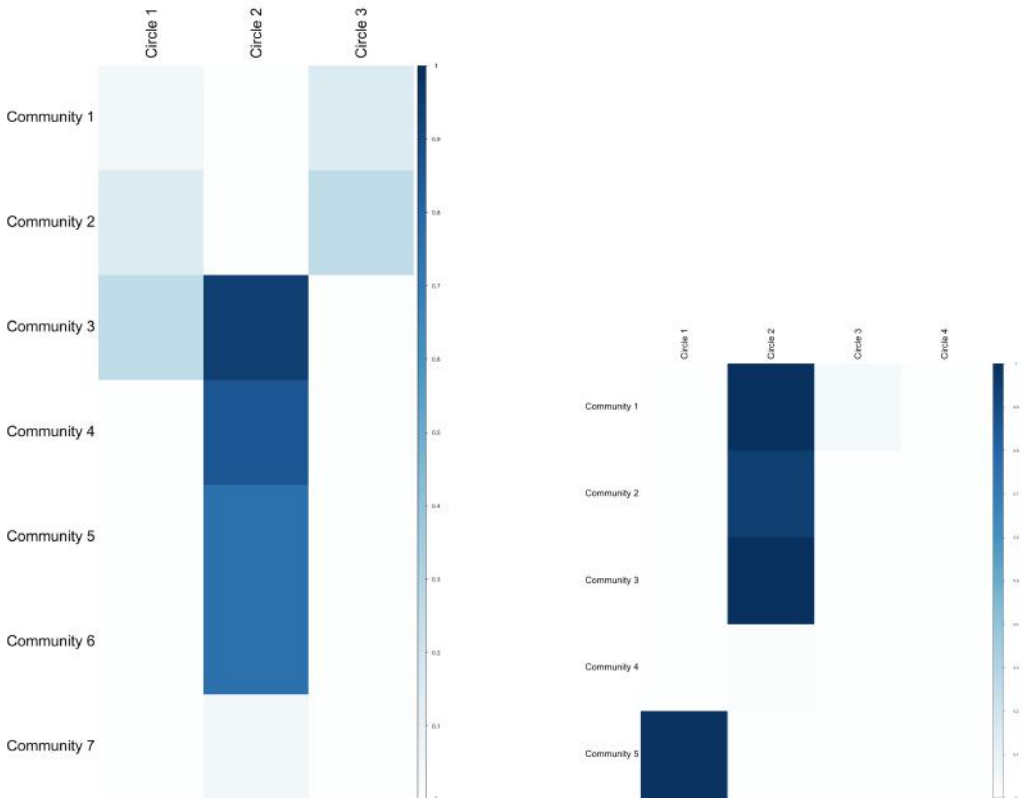

Figure 23: Infomap Community Overlap Node - 1 & 2

`Node 3`

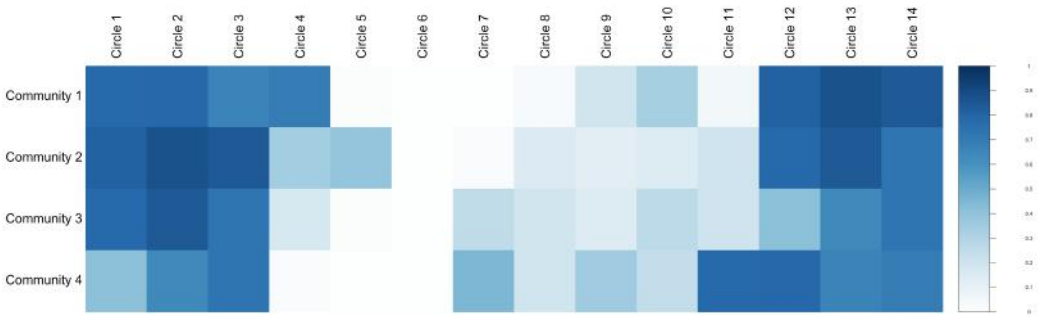

Figure 24: Infomap Community Overlap Node - 3

7.5 Analysis of Results

1. The `walktrap.community` algorithm is shown to give more number of communities as compared to the `infomap.community` detection algorithm for the Google+ network.

2. The following observations can be made from the overlaps between the communities and the circles,

   - **Google+ social network** has 3 inbuilt circles for new users - `Friends, Family & Acquaintances`.
   - **Variation in Overlap:** As a result, we can observe that communities of ego-nodes which have fewer number of circles, seem to be concentrated within these in default circles. The relationships are assigned by default by Google+. This is depicted in the plots of `Nodes 1 & 2`. For user's with larger number of circles, community nodes are shown to be distributed among the circles and not a defined structure can be seen.
   - **Relationship with Circle:** As the number of circles increases users tend to choose their relationship with people and choose their own circles for them (e.g business, colleagues etc.). In situations like these people related to the ego-node might belong to multiple circles. For example a person can be in the Family as well as Friends circle.