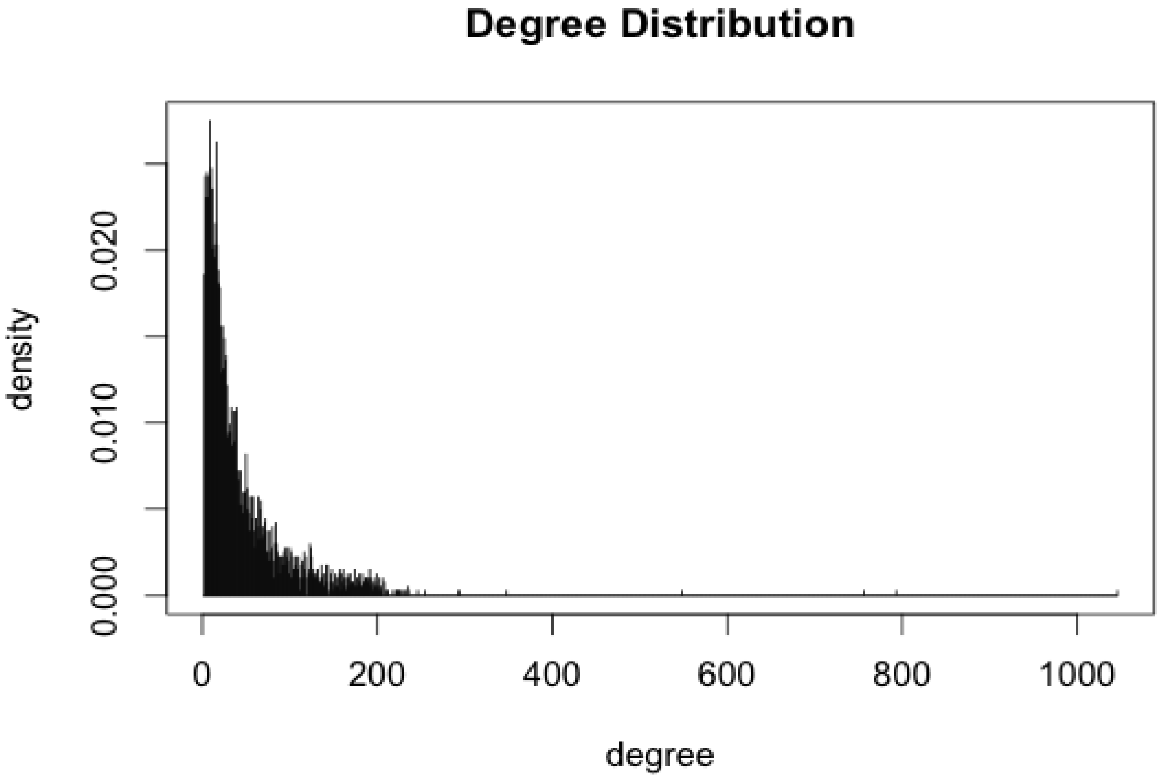
Project 1

EE232E - Graphs and Network Flows

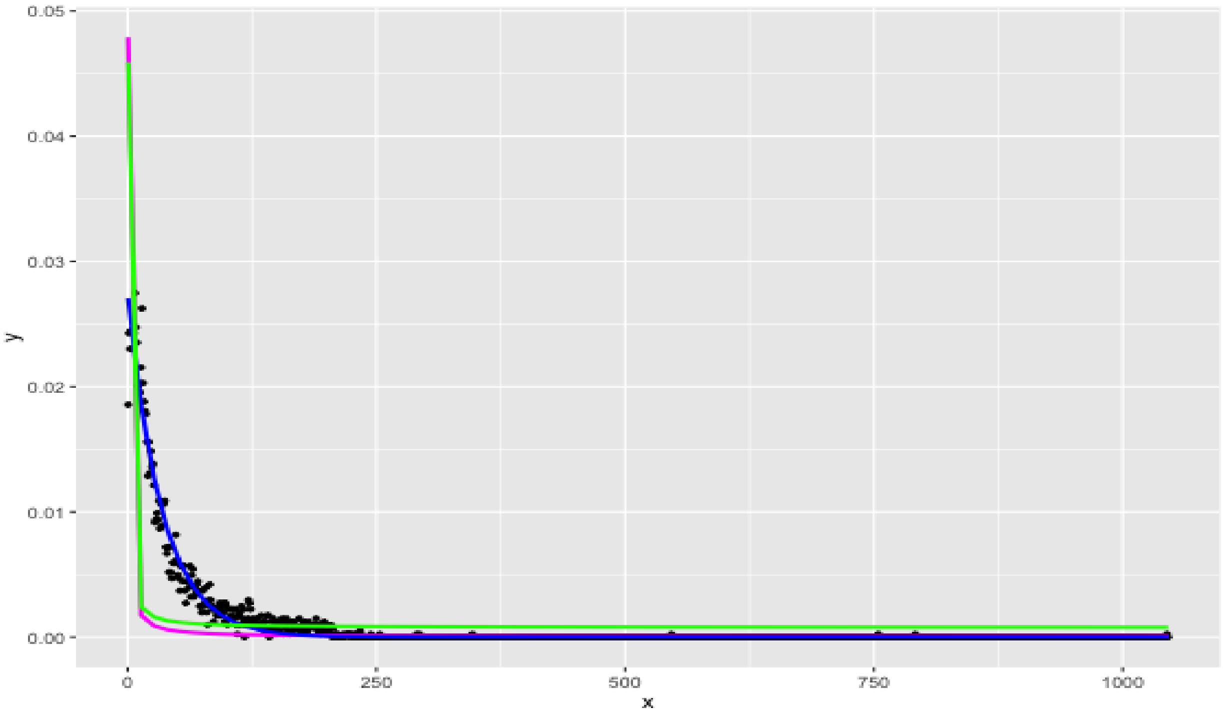
Team:
Rutuja Ubale (UID: 404558257)
Usha Amrutha N (UID: 204590772)
Pallabi Chakraborty (UID: 404519609)

Question 1:

For this section, we have used the function read.graph() to construct a graph from the edge list file facebook_combined.txt. We note that the network is connected. The **diameter** of the network was found to be 8. The degree distribution is as shown below.



In order to generate a model to fit a curve on the degree distribution we have used stat_smooth() function from ggplot2 package. We have tried three models y ~ I(1/x*a) + b*x, y ~ I(exp(1)^(a + b * x) and y ~ I(1/x*a) + b to fit the distribution and it was found that the model **y ~ I(exp(1)^(a + b * x)** fits the distribution better. By analyzing the shape of the curve in the plot below we see that the blue curve fits the distribution better than the red and green curves where the **blue** curve is for the second model and the magenta and green curves indicate the first and third models respectively.



The parameters of the best statistical model as indicated by the summary() method is as follows:

Formula: y ~ I(exp(1)^(a + b * x))

Parameters:

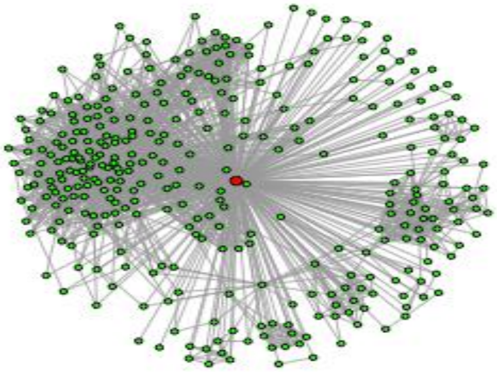
	Estimate	Std. Error	t value	Pr(>|t|)
a	-3.5940045	0.0078770	-456.26	<2e-16
b	-0.0291488	0.0003247	-89.77	<2e-16

Residual standard error: 0.0006339 on 1044 degrees of freedom

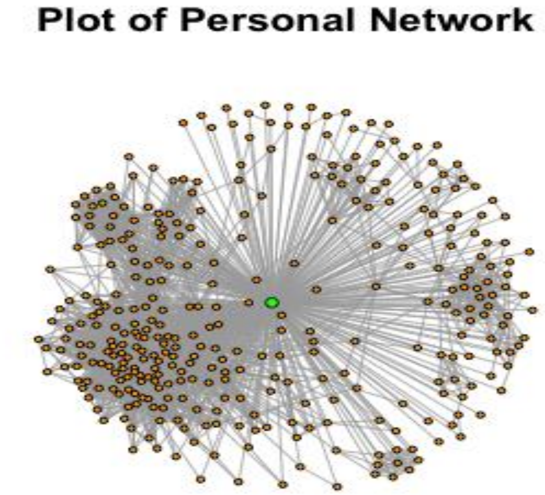
Number of iterations to convergence: 15
Achieved convergence tolerance: 7.841e-07

The **total mean squared error** of the curve is 4.01487e-07. The **average degree** is 43.69.

Question 2:
For this question we take the first node in the graph and generate a subgraph consists of node 1 and its neighbors and the edges that have both ends within this set of nodes. From the figure below we can see that among all the nodes in the personal network of node 1 except for node 1, all other nodes are all friends of node 1.
The **number of nodes** in this graph are 348 and the **number of edges** are 2866.
The plot of the personal network is as shown below where the red dot indicates node 1.

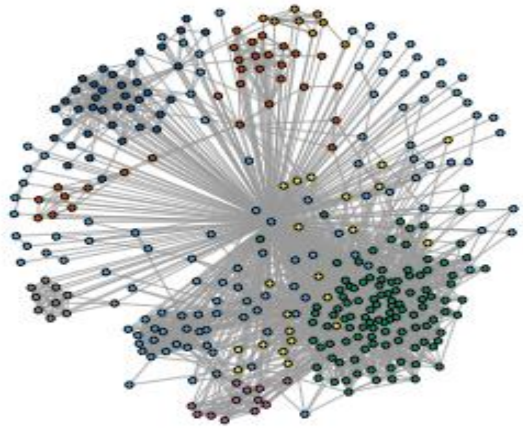


Question 3:
In this question we determine the core nodes i.e. the nodes that have more than 200 neighbors. There are **40 core nodes** in this network and the **average degree** of the core nodes is **279.375**. We have extracted the community structure of node 1, which is a core node. The plot of personal network of node 1 is as shown below.

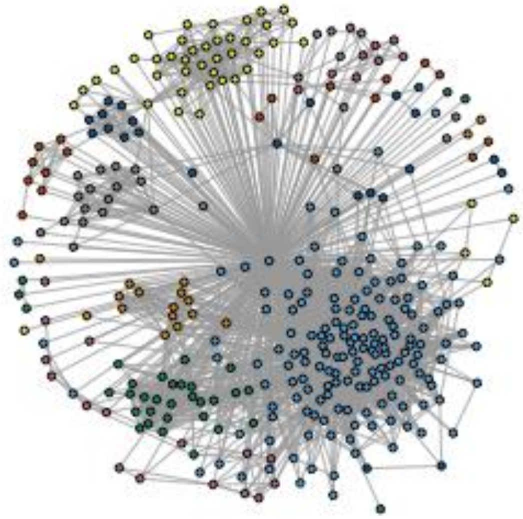


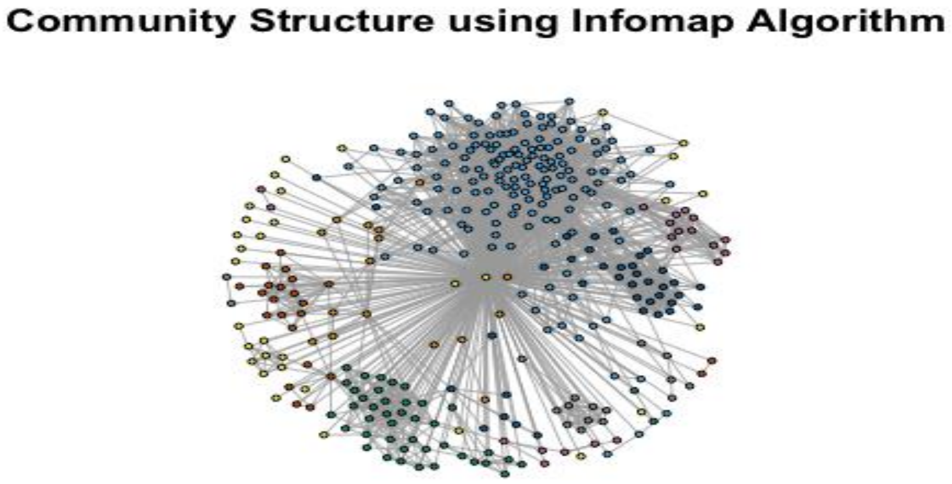
The figures below show the community structure of personal network of node 1 using Fast-Greedy algorithm, Edge-Betweenness algorithm and Infomap algorithms.

Community Structure using Fast-Greedy Algorithm



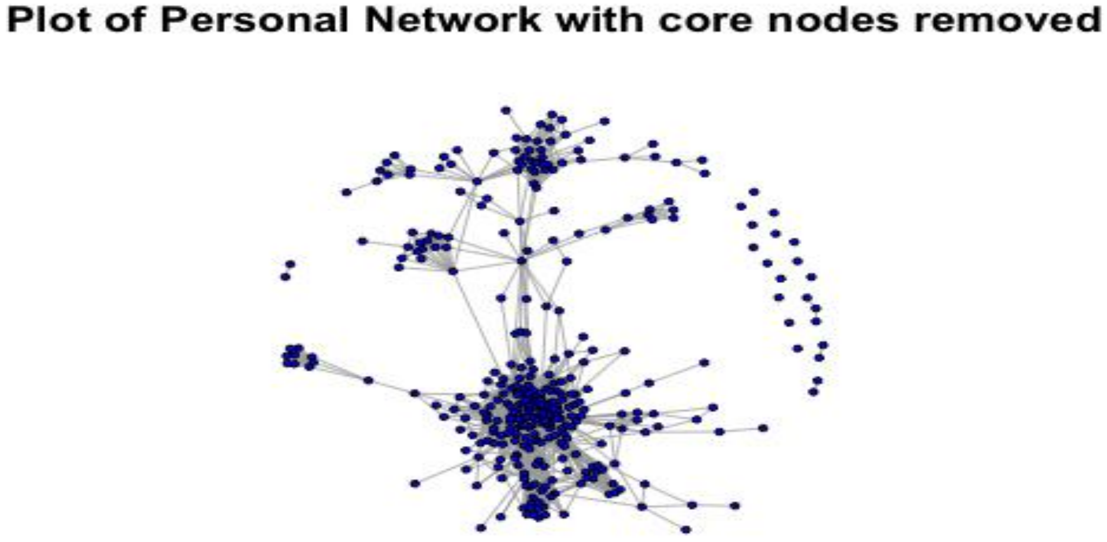
Community Structure using Edge-Betweenness Algorithm



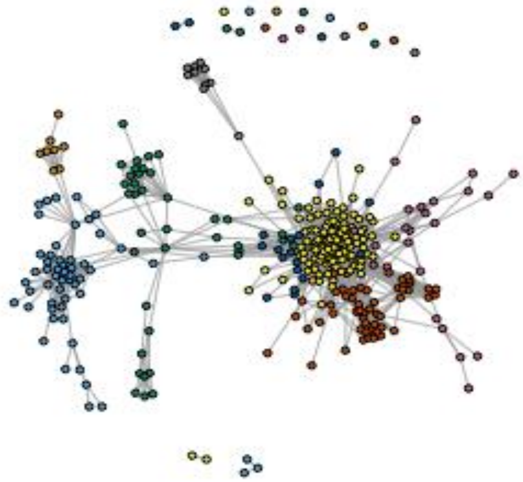

Community Structure using Infomap Algorithm

The communities in the figures above are distinguished with colors and it can be seen that in the communities plotted there is some overlap. Also it is observed that Edge-Betweenness algorithm tends to break the graph into more partitions than the other two algorithms. The modularity of Fast-Greedy algorithm, Edge-Betweenness algorithm and Infomap algorithms is 0.4131014, 0.3533022 and 0.3891185 respectively.

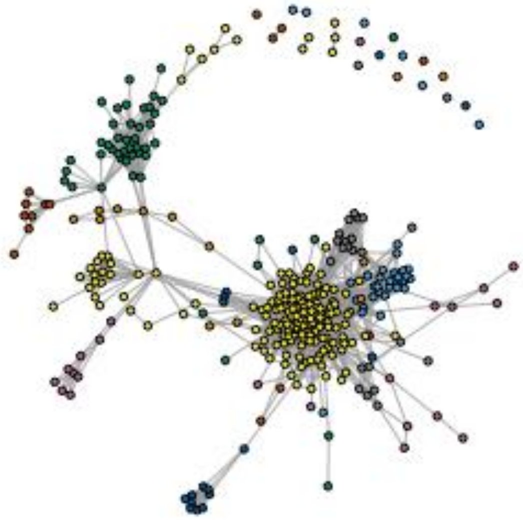
Question 4:
For this problem, core nodes were removed from the personal network and then the community structure was determined again using the three community detection algorithms as in part 3. It was observed that the partitions are similar to those in part 3 even though they are structured without the core node. Also there is a difference of approximately 10% in the modularity with respect to part 3. The plots are shown below.


Plot of Personal Network with core nodes removed

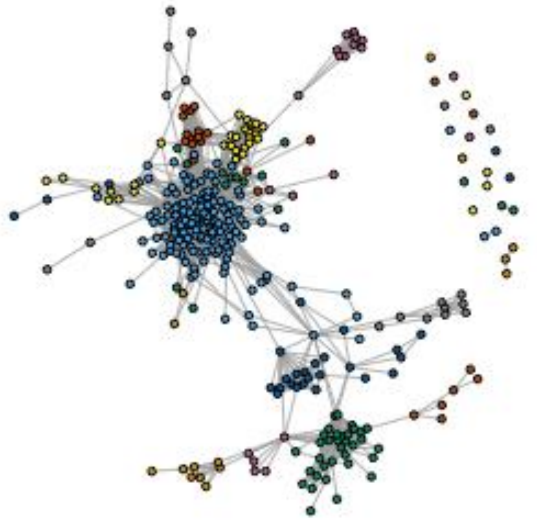
Community Structure using Fast-Greedy Algorithm



Community Structure using Edge-Betweenness Algorithm



Community Structure using Infomap Algorithm

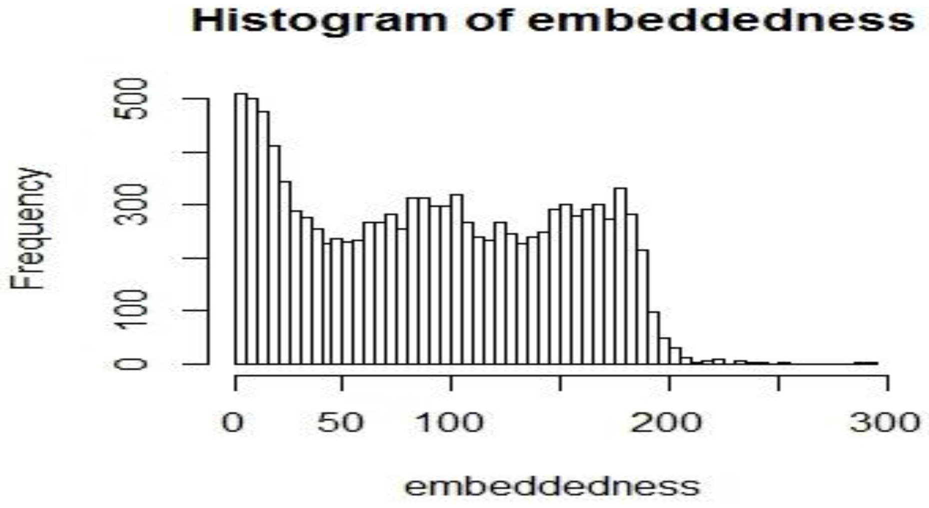


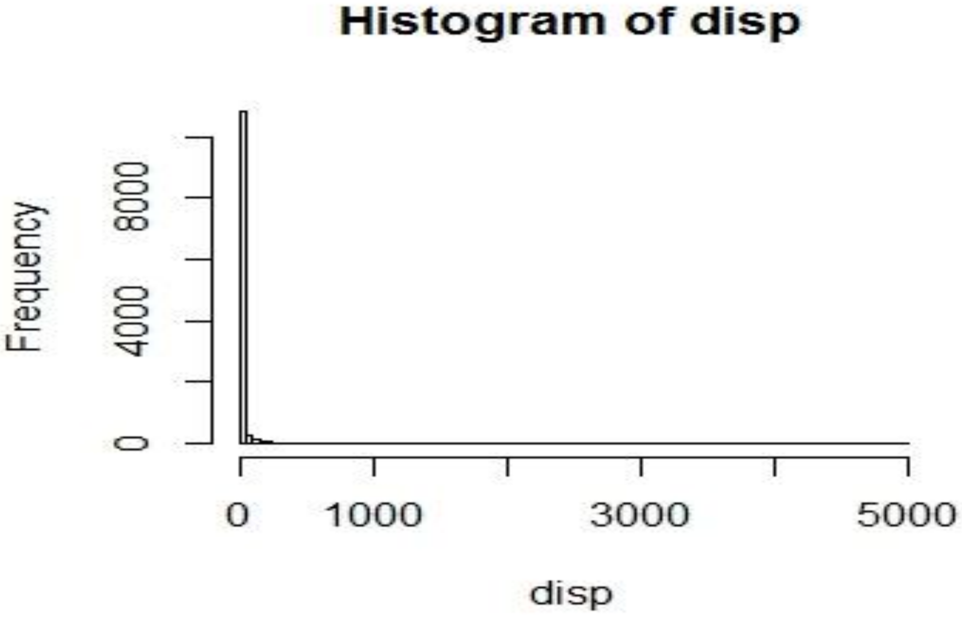
Question 5:

The question asks for dispersion and embeddedness for all nodes in a personal network. "Romantic Partnerships and the Dispersion of Social Ties: A Network Analysis of Relationship Status on Facebook", Lars Backstrom, Jon Kleinberg, mathematically, defines dispersion and embeddedness. Embeddedness of two given nodes is defined as the number of common nodes shared by them. It reflects the strength of ties between these nodes. On the other hand, Dispersion checks if two given nodes have mutual nodes which are connected only through these two nodes. The paper cited above states dispersion as:

$$disp(u, v) = \sum_{s,t \in C_{uv}} d_v(s, t),$$

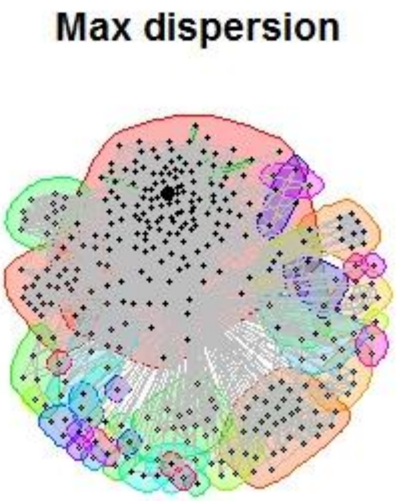
S and t are nodes that belong to common set C*uv* of nodes u and v, where u is the core node of the personal network and v is the node in personal network of a which could be U’s potential partner. *dv(s,t)* is the distance measure calculated as follows: If nodes s and t are disconnected, after removing u and v from the set, then we assign 1 to *dv(s,t)*, and have no common neighbours in G*u* (subgraph induced on node u). Otherwise, we set *dv(s,t) to 0.* The dispersion and embeddedness values have been calculated for 40 personal networks, and the distributions are as follows.

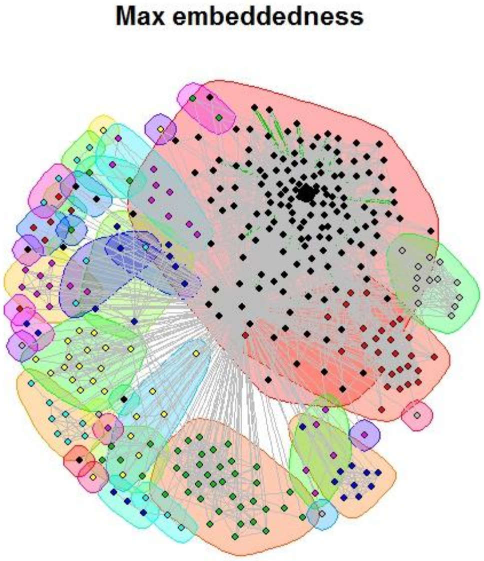




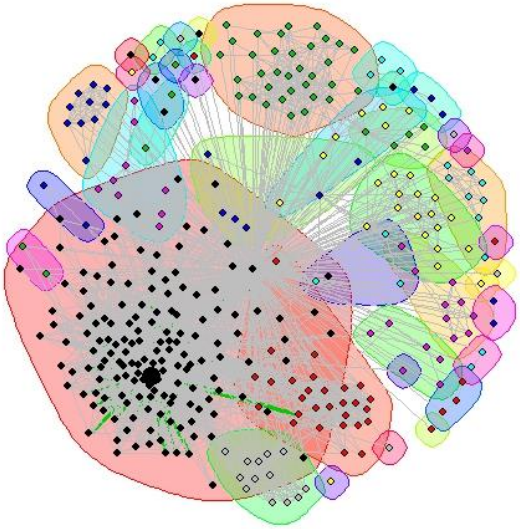
The personal networks 1, 17, 22 have been plotted in the below graphs. The maximum embedness, dispersion and ration nodes have been shown in black color and the edges have been highlighted in green color. We note that, the maximum dispersion for the node (denoted in black) is connected through core node, and the connecting edges are not as dense as embeddedness. This reveals that this node share friends of core node from different circles (say, college, family etc.) hence the dispersion is large. According to the paper, the nodes in blue can be potential partners to the core node. On the other hand, the graphs for embeddedness show a node which is has maximum embeddedness with core node. This node could be a node with maximum social tie with the core node. The graphs below illustrate the same.

Core node 1



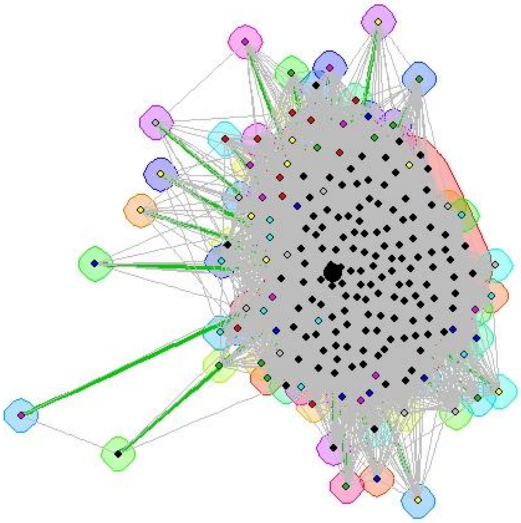


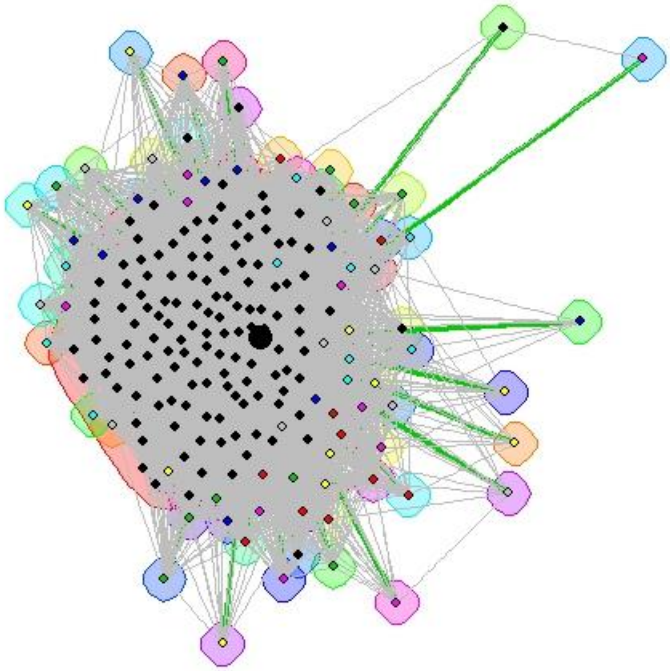
Max dispersion/embeddedness



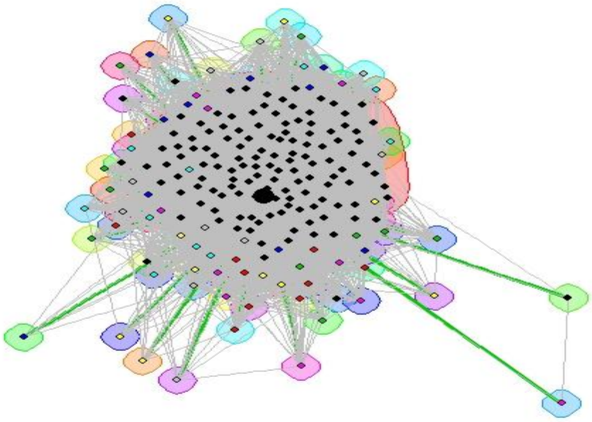
Core node 18

Max dispersion



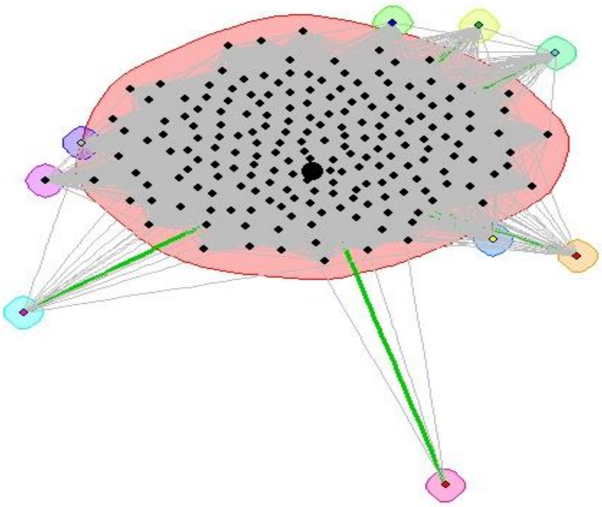


Max embeddedness

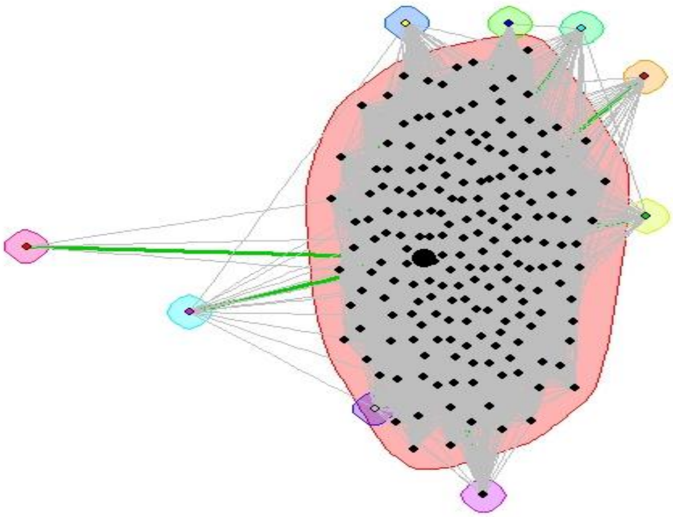


Max dispersion/embeddedness

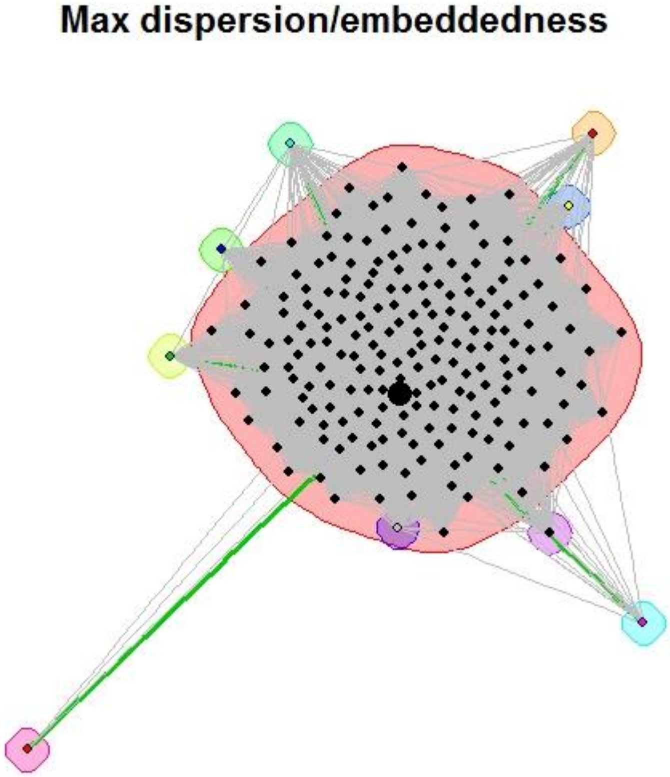
Core Node 22



Max dispersion



Max embeddedness



Question 6:

In this section, we worked on identifying features that help determine if the communities belong to certain categories like “college friends”, “classmates”, “work colleagues”, “family”, etc. The table below shows the communities with maximum and minimum average degree, clustering coefficient and density respectively over all core personal networks. We have used 3 features - average degree, clustering coefficient and density to represent the closeness of the community for communities with size larger than 10 nodes. Using these features we attempt to find out which community’s members have highest closeness and lowest closeness. For e.g., the community “close friends” could probably have the higher closeness, whereas the one lower closeness could probably be “acquaintances” community.

Core nodes	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Index	16	14	4	2	8	2	3	2	6	5	5	13	3	3	2	3	1	1	6	2
max_avg_degree	0.80	0.72	0.64	0.49	0.61	0.68	0.71	0.60	0.69	0.85	0.80	0.75	0.70	0.75	0.55	0.64	0.57	0.53	0.71	0.83
Index	16	12	4	2	8	2	6	3	6	5	5	13	3	3	2	3	1	1	6	2
max_cluster_coef	0.89	0.89	0.80	0.71	0.73	0.81	0.85	0.73	0.77	0.90	0.87	0.87	0.78	0.81	0.69	0.74	0.74	0.68	0.83	0.89
Index	16	12	4	2	8	2	6	3	6	5	5	13	3	3	2	3	1	1	6	2
max_density	0.87	0.78	0.67	0.50	0.61	0.69	0.75	0.61	0.71	0.89	0.83	0.80	0.71	0.76	0.55	0.65	0.59	0.53	0.75	0.84
Index	3	1	1	1	9	3	4	3	7	3	1	2	2	2	1	1	2	2	1	1

min_avg_degree	0.19	0.08	0.22	0.35	0.60	0.56	0.53	0.59	0.65	0.63	0.46	0.1	0.54	0.58	0.46	0.45	0.57	0.49	0.16	0.60
Index	3	1	1	1	9	3	4	2	7	3	1	2	2	2	1	1	2	2	1	1
min_cluster_coef	0.37	0.40	0.49	0.54	0.71	0.71	0.65	0.71	0.75	0.71	0.63	0.31	0.70	0.72	0.65	0.62	0.70	0.64	0.40	0.75
Index	3	1	1	1	9	3	4	3	7	3	1	2	2	2	1	1	2	2	1	1
min_density	0.20	0.08	0.22	0.36	0.61	0.58	0.55	0.59	0.67	0.65	0.47	0.10	0.55	0.60	0.47	0.45	0.57	0.49	0.16	0.61
Core nodes	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**
comm_index	1	1	5	1	1	6	1	3	1	5	1	5	2	3	1	3	2	4	3	23
max_avg_degree	0.85	0.79	0.89	0.88	0.82	0.79	0.80	0.87	0.86	0.72	0.78	0.75	0.76	0.77	0.88	0.85	0.78	0.79	0.78	0.86
index	1	1	5	1	1	6	1	3	1	5	1	5	2	3	1	3	2	4	5	23
max_cluster_coef	0.91	0.84	0.97	0.91	0.86	0.84	0.86	0.94	0.89	0.80	0.84	0.82	0.83	0.83	0.90	0.89	0.85	0.84	0.88	0.93
index	1	1	5	1	1	6	1	3	1	5	1	5	2	3	1	3	2	4	5	23
max_density	0.90	0.80	0.97	0.89	0.83	0.80	0.81	0.93	0.86	0.72	0.78	0.76	0.77	0.77	0.88	0.86	0.79	0.79	0.80	0.92
Index	3	2	3	4	3	2	2	4	3	5	3	4	1	1	2	1	1	3	5	4
min_avg_degree	0.40	0.70	0.41	0.61	0.56	0.68	0.70	0.75	0.56	0.72	0.67	0.41	0.66	0.67	0.54	0.74	0.55	0.71	0.74	0.18
Index	3	2	3	4	3	2	2	4	3	5	3	4	1	1	2	4	1	3	3	2
min_cluster_coef	0.64	0.80	0.62	0.66	0.71	078	0.78	0.80	0.68	0.80	0.82	0.64	0.77	0.77	0.64	0.80	0.72	0.83	0.83	0.37
Index	3	2	3	4	3	2	2	4	3	5	3	4	1	1	2	4	1	3	6	4
min_density	0.43	0.71	0.43	0.66	0.58	0.69	0.72	0.75	0.60	0.72	0.73	0.45	0.67	0.70	0.54	0.77	0.56	0.74	0.78	0.18

A combination of the three measures mentioned would be good enough for estimating the closeness of communities across different people’s personal network. However, they may not be sufficient in determining which communities have higher closeness. For this question, we have explored only two types of communities. However, in addition to the three measures mentioned above other parameters like community size, modularity, embeddedness and dispersion could be used to identify other types of communities. For example, the community “family” could have high closeness but a small community size whereas the community “acquaintances” could have a large community size and lower closeness.

In addition to average degree, clustering coefficient and density we have tried other measures like community size and modularity. The **number of personal networks** with communities whose size is greater than 10 nodes is thus **40**. Each statistical feature is rounded off and tabulated. The corresponding frequencies of each feature are displayed below.

Community Size	0	25	50	75	**100**	125	150	175	200	225	325	375	475
No. of communities	3	13	3	16	**37**	22	3	2	1	2	1	1	2

Community Density	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
No. of communities	9	9	2	8	13	17	15	26	7

Modularity	0	**0.1**	0.2	0.3	0.4	0.5
No. of communities	40	**41**	7	9	6	3

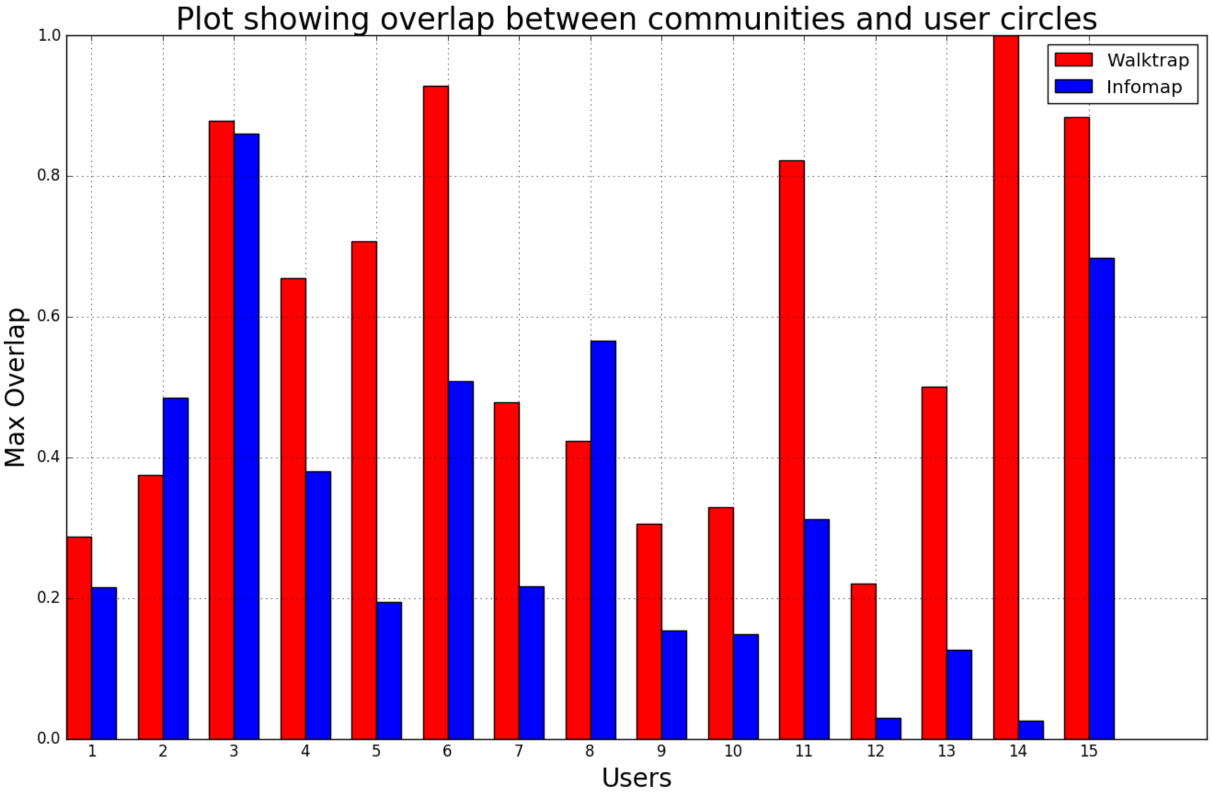
Clustering Coefficient	0.3	0.4	0.5	0.6	0.7	0.8	0.9
No. of communities	1	5	7	9	28	30	26

Average Degree	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
No. of communities	9	9	2	8	13	18	16	29	2

From the tables, we note that there are **37 communities** with community size approximately equal to 100. Similarly, there are **41 communities** with modularity approximately 0.1. Thus, the two recurring types of communities are **100-sized communities** and **0.1-modularity communities**.

Question 7:

We try to run the same kind of analysis another real social network with tagged relationships. We create personal networks for users who have more than two circles (the default number). We also extract the community structure of each personal network using both Walktrap and Infomap algorithms and show how communities overlap with the users' circles. We then check how overlaps vary across users and how this relates to a user's habit on tagging relationships with circles.



From the above plot, we can see that overlap varies across users. The formula used to calculate overlap is as follows:

Overlap(A,B) = Number of elements in Intersect(A,B)/ Number of elements in Union(A,B)

This value is calculated for all possible combinations of communities and circles, and the final value reported is the maximum value of overlap. The aim of the algorithm is to find these communities by using the knowledge of edges between the nodes.

A higher overlap value indicates that the user is very efficient in the tagging process. A lower overlap value indicates that the user has added other users to multiple circles or the wrong circle, perhaps indicated by the other user adding me to a circle that is distinct from the one I added her to. In Google Plus, users can generate new circles. For example, we can create a circle called Close Friends and create a subset of my existing Friends to classify some of the users added there to the new circle. But it is not necessary that the user I add in that circle will also add me in those circles.

Walktrap is a better algorithm than Infomap as indicate by the plot above. Out of the 15 users, Walktrap performed better in the case of 13 out of them.