<u>EE 232E</u>

<u>Project - I Report</u>

Chandan Dhal - 904588105

Sidharth Gulati - 104588717

Tushar Sudhakar Jee - 004589213

Introduction

In this project, we used a graph of users' personal networks. In the first part of the project, we analyzed facebook users' friendship network and in the second part of the project, we analyzed Google+ users' circles network . The main focus of this project is to analyze the community structures of personal networks and use them to predict romantic ties and tagging relationships.

Facebook Network

Exercise 1

Network

We generated a network using the edge list from the facebook dataset. As the dataset are from Facebook, we can infer that the network has to be undirected as there is a mutual connection between the two users in this social network. The figure 1 depicts the network and the network parameters are tabulated as below:-

Parameters	Values
Connected	True
Nodes(users)	4039
Diameter	8
Edges(connections)	88234

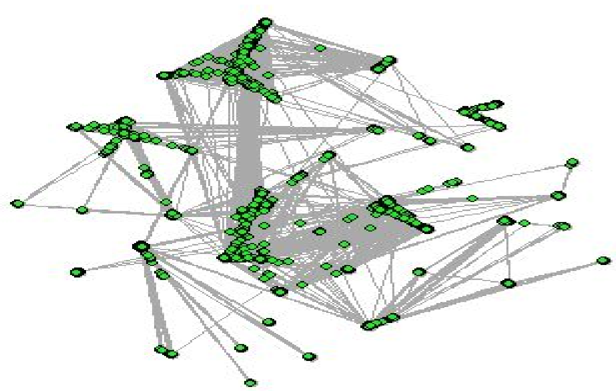


Fig 1. Facebook Network

Degree Distribution

The figure 2 and 3 depicts the degree distribution and histogram of degree distribution of the nodes in the network.

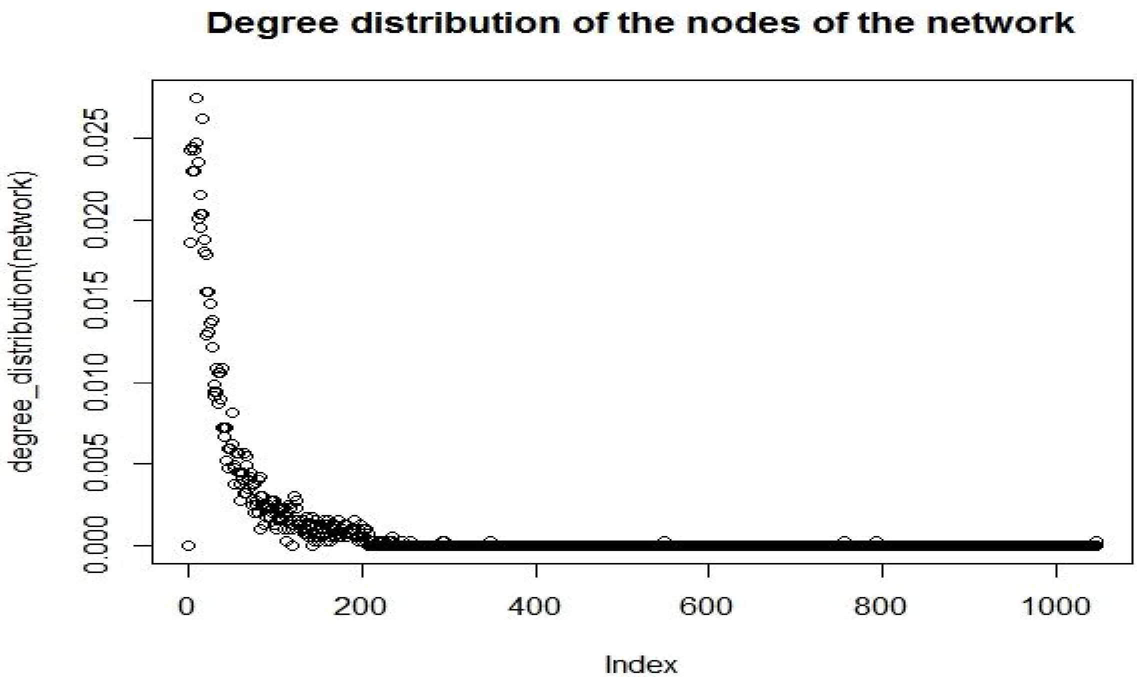


Fig 2. Degree distribution of the nodes of the network

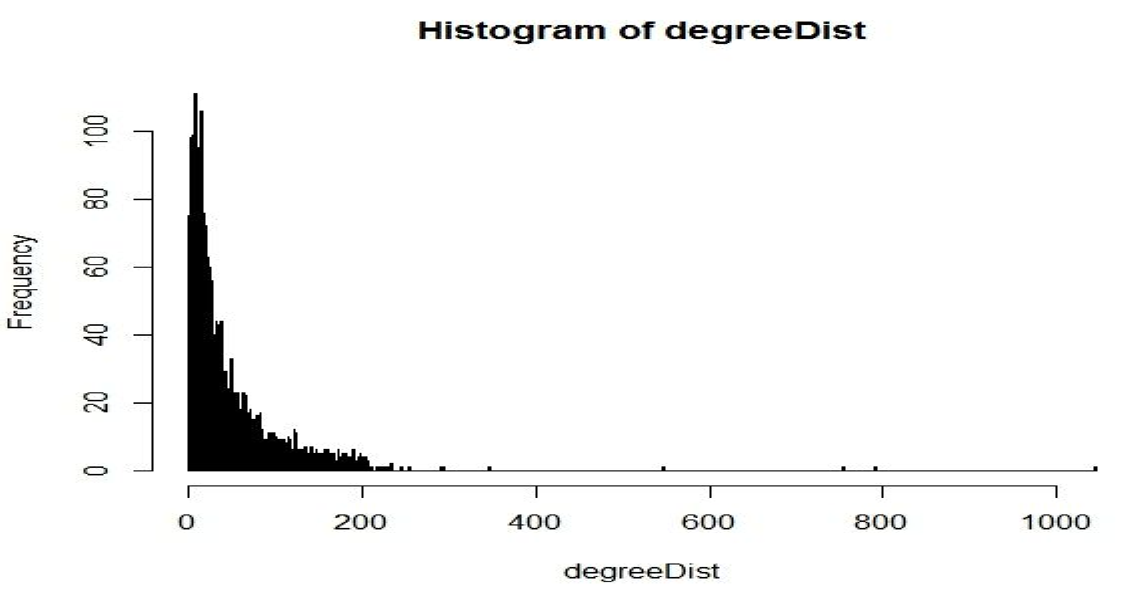


Fig 3. Histogram of degree distribution of the nodes of the network

The average degree of the nodes in the network is 43.691, i.e. each user (node) has 44 friends(connections) in their network.

Degree distribution curve fitting models

From figure 2 and 3 we can infer that the degree distribution follows a model similar to decaying model. We experimented with three functions namely, inverse model, logarithmic model and decaying exponential model and calculated the total mean squared error.
The below plot depicts the inverse model matched with actual degree distribution curve.

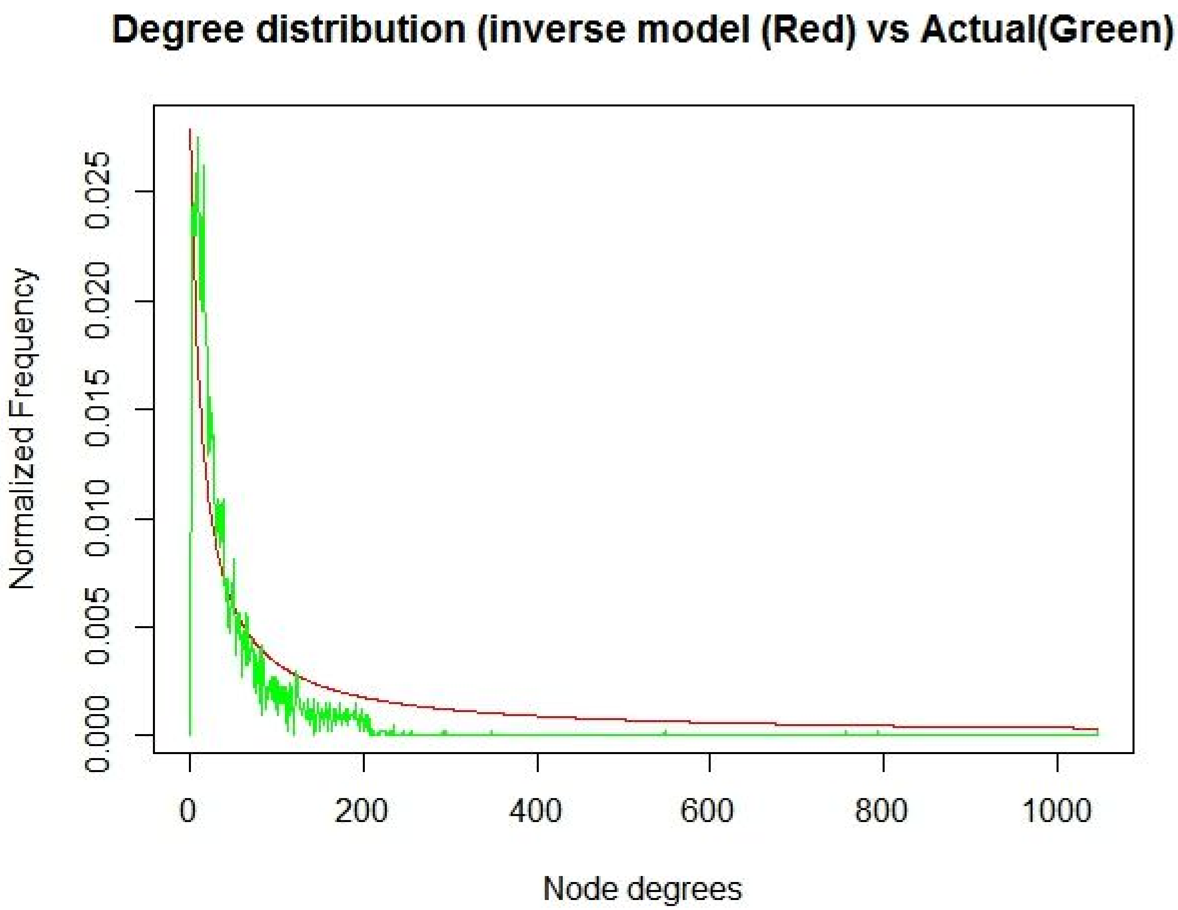


Fig 4. Inverse Model

The below plot depicts the logarithmic model matched with actual degree distribution curve.

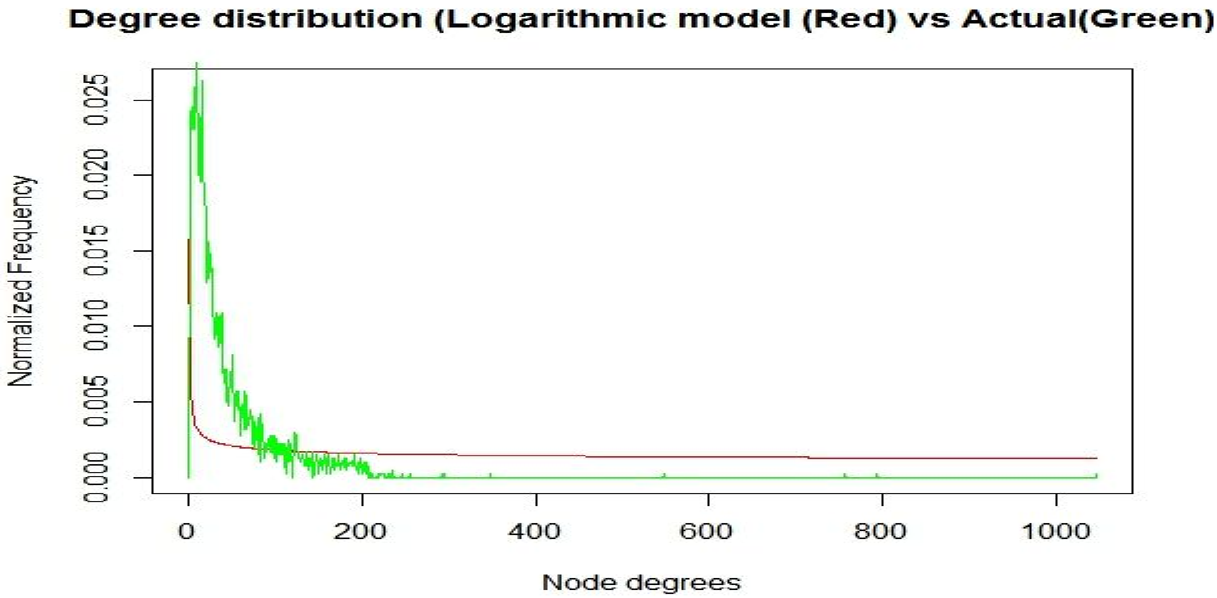


Fig 5. Logarithmic Model

The below plot depicts the decaying exponential model matched with actual degree distribution curve.

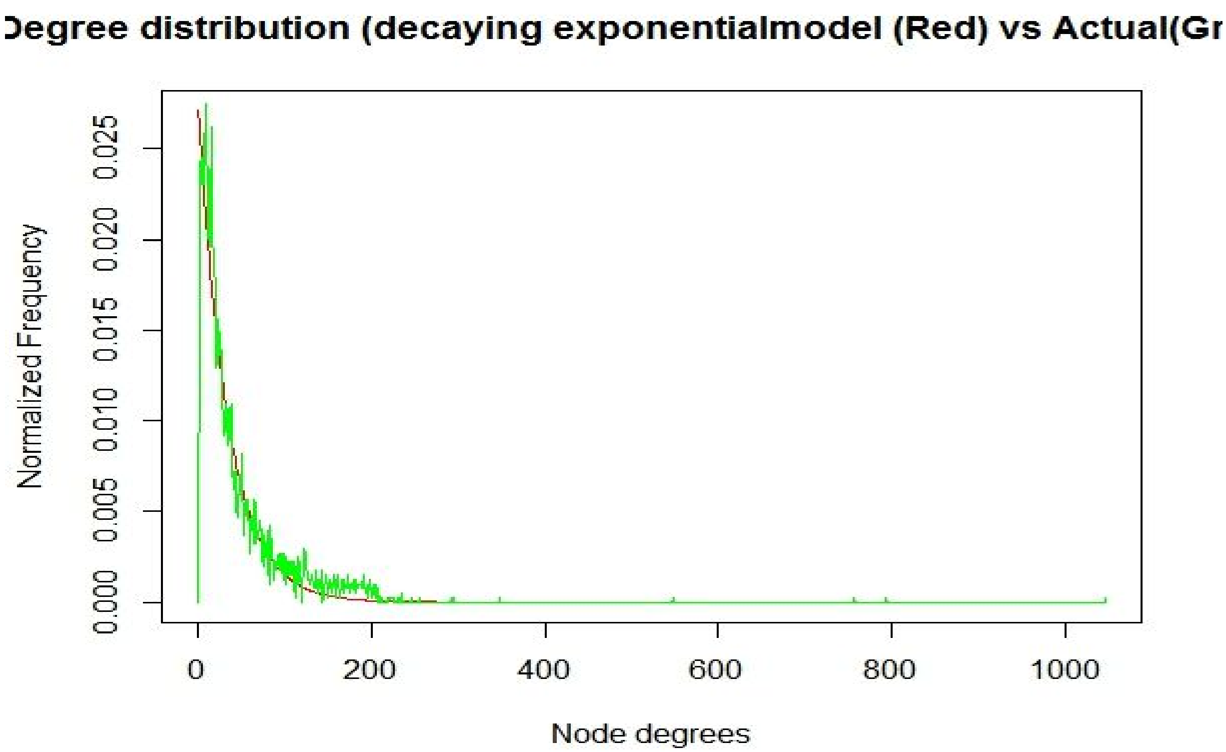


Fig 6. Decaying exponential Model

We can observe that the decaying exponential model and the inverse model is almost fitting in the degree distribution curve. The mean squared error for the models are tabulated as below.

Model	Total Mean square Error
Logarithmic	1.019291e-05
Inverse	2.472539e-06
Decaying exponential	4.011031e-07

The decaying model has the least mean squared error, hence the decaying model fits the degree distribution curve.

Exercise 2

Personal network of the nodes

We created a graph which consists of node 1 and its neighbors and the edges that have both ends with in the set of nodes. This network is defined as the personal network of node 1. The figure below depicts the personal network of node 1.

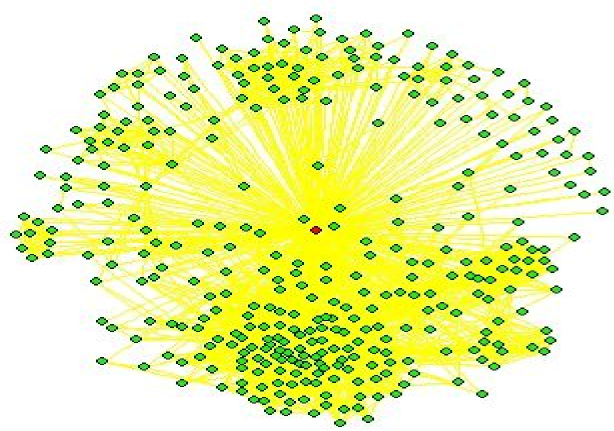


Fig 7. Personal network of node 1(Red) and its neighbors (Green)

The personal network of node 1 has 348 nodes and 2866 edges, i.e these 348 users has user 1 as their mutual friend

Exercise 3

Community structure of the core personal network

In this exercise we selected the core nodes (which have more than 200 neighbors) and analyzed the community structure of each of their personal network. The network has 41 core nodes and the average degree of the core nodes is 277.439, i.e. every core has 278 friends in average. We considered the node 39 for analyzing the community detecting algorithms.

Edge-Betweenness community detection algorithm

This algorithm is based on link-centrality measures which relies on a hierarchical divisive approach. Initially the whole network is seen as a single community, i.e. all nodes are in the same community. The most central links are then repeatedly removed. The underlying assumption is that these particular links are located between the communities. After a few steps, the network is split in several components which can be considered as communities in the initial network. Iterating the process, one can split each discovered community again, resulting in a finer community structure. This eventually leads to a network in which each node is isolated, and therefore constitutes its own community. By considering the communities detected at each step of the process, one obtains a hierarchy of community structures.

The figure below depicts the community structure of personal network of core node 39. Each individual color corresponds to a Community and the nodes are colored according to their community color and the intra-community edges are colored black and the inter-community edges are colored red. The algorithm reported 16 communities and we can observe that most of the nodes in the network below to the orange community. Since edge-betweenness community detection community algorithm finds communities of very uneven sizes (one big community and the rest are very small), so the modularity of the partitioning is very low (0.001).

Figure 9 depicts the intra-community parameters of the personal network of core 39.

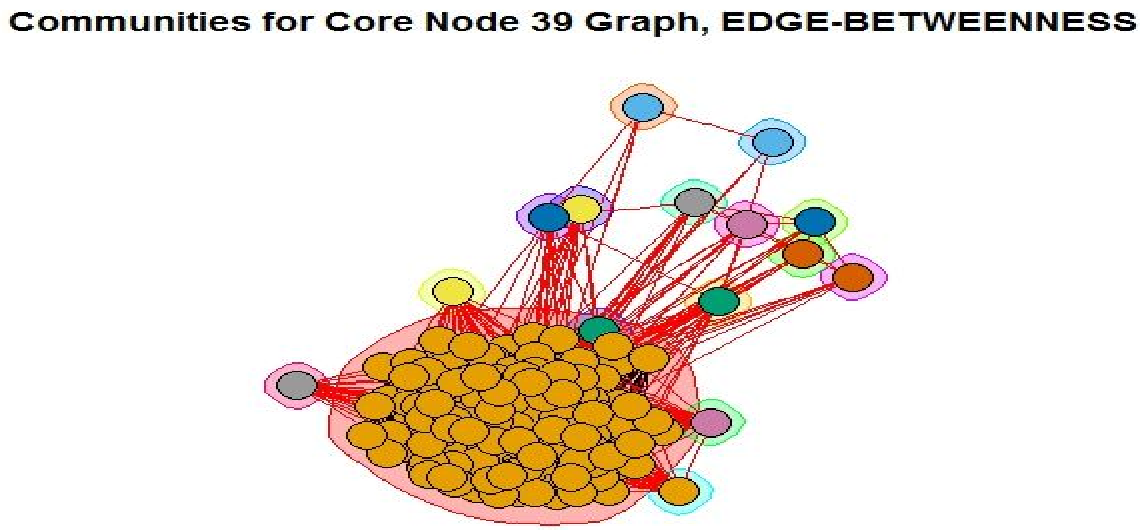


Fig 8. community structure of personal network of core node 39(Edge Betweenness)

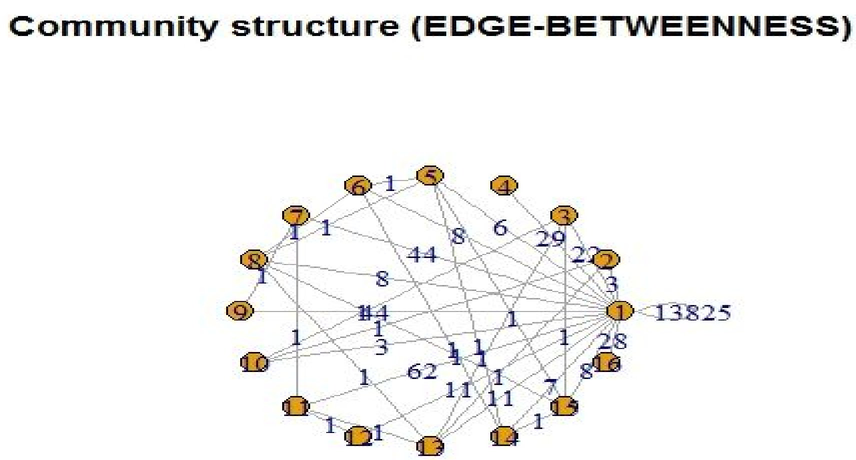


Fig 9. Community structure of core node 39.(Edge-Betweenness). The circles represents the 16 communities. The loops around the circles represents the number of edges between the nodes in that community and the lines between two communities represents the number of edges between two communities. For example, there are 13825 edges between the nodes in community 1 and 3 edges between community 1 and 2.

The various Community parameters of the largest community are tabulated below:

Community ID	Number of nodes
1	187
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1

Fast-Greedy community detection algorithm

The Fast-Greedy(FG) algorithm relies on a greedy optimization method applied to a hierarchical agglomerative approach. The agglomerative approach is symmetrical to the divisive one described in the previous subsection. In the initial state, each node constitutes its own community. The algorithm merges those communities step by step until only one remains, containing all nodes. The greedy principle is applied at each step, by considering the largest increase (or smallest decrease) in modularity as the merging criterion. Because of its hierarchical nature, FG produces a hierarchy of community structures like the divisive approaches. The best one is selected by comparing their modularity values.

The figure below depicts the community structure of personal network of core node 39. We can observe that the nodes are colored according to their community membership and the edges are colored depending on their type. For example, the nodes are colored as blue, green and orange. The intra-community edges are colored black and the inter-community edges are colored red. It can be seen that most of the nodes are colored blue or green, implying that most of the nodes in core node 39 personal network belong to either of the two communities. Also, since there are lots of edges connecting communities (red edges), so the modularity of the partitioning is low (0.03).

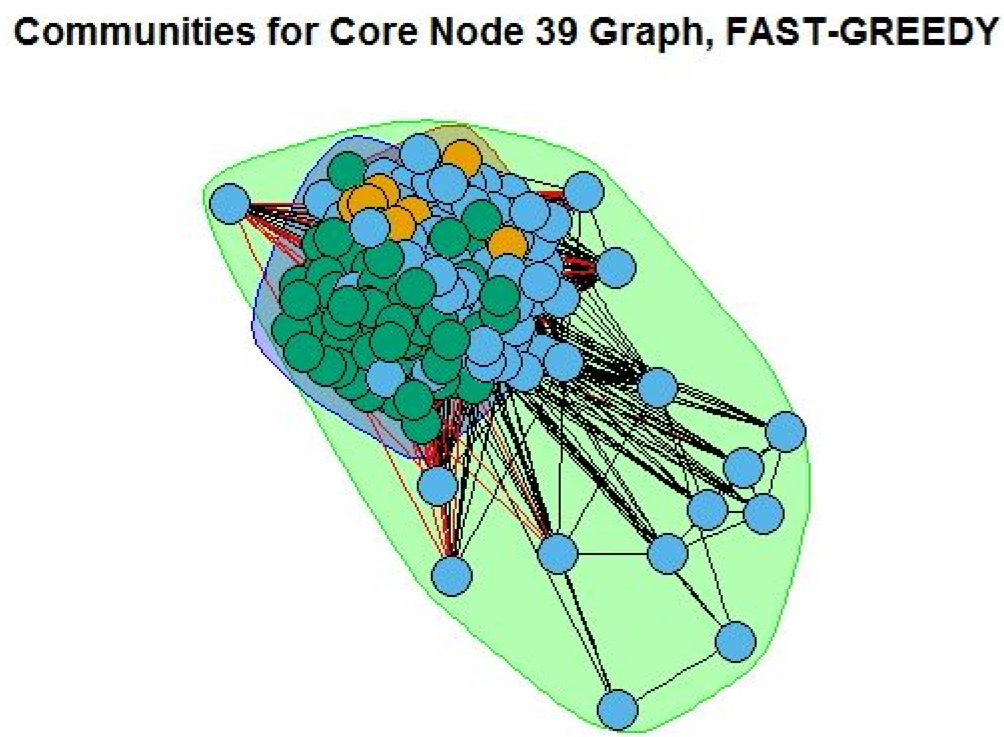


Fig 10. community structure of personal network of core node 39(Fast-Greedy)

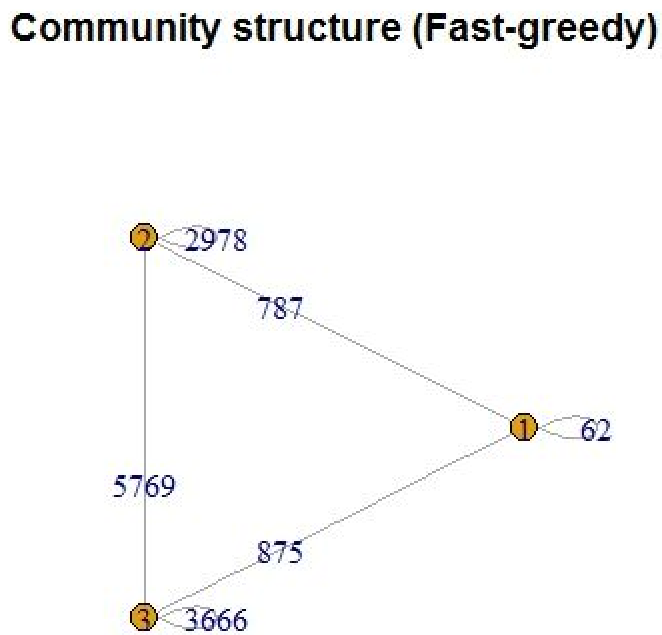


Fig 11. Community structure of core node 39.(Fast-Greedy). The circles represents the 3 communities. The loops around the circles represents the number of edges between the nodes in that community and the lines between two communities represents the number of edges between two communities. For example, there are 62 edges between the nodes in community 1 and 875 edges between community 1 and 3.

The various Community parameters of the largest community are tabulated below:

Community ID	Number of nodes
1	12
2	95
3	95

Infomap community detection algorithm

Information-Based algorithms use tools derived from the information theory to estimate the best partition of the network. The community structure is represented through a two-level nomenclature based on Huffman coding: one to distinguish communities in the network and the other to distinguish nodes in a community. The problem of finding the best partition is expressed as minimizing the quantity of information needed to represent some random walk in the network using this nomenclature. With a partition containing few inter-community links, the walker will probably stay longer inside communities, therefore only the second level will be needed to describe its path, leading to a compact representation.

The figure below depicts the community structure of personal network of core node 39. Each individual color corresponds to a Community and the nodes are colored according to their community color. The algorithm failed to detect any community and reported the entire personal as one large community and as a result the modularity of the partitioning is 0.

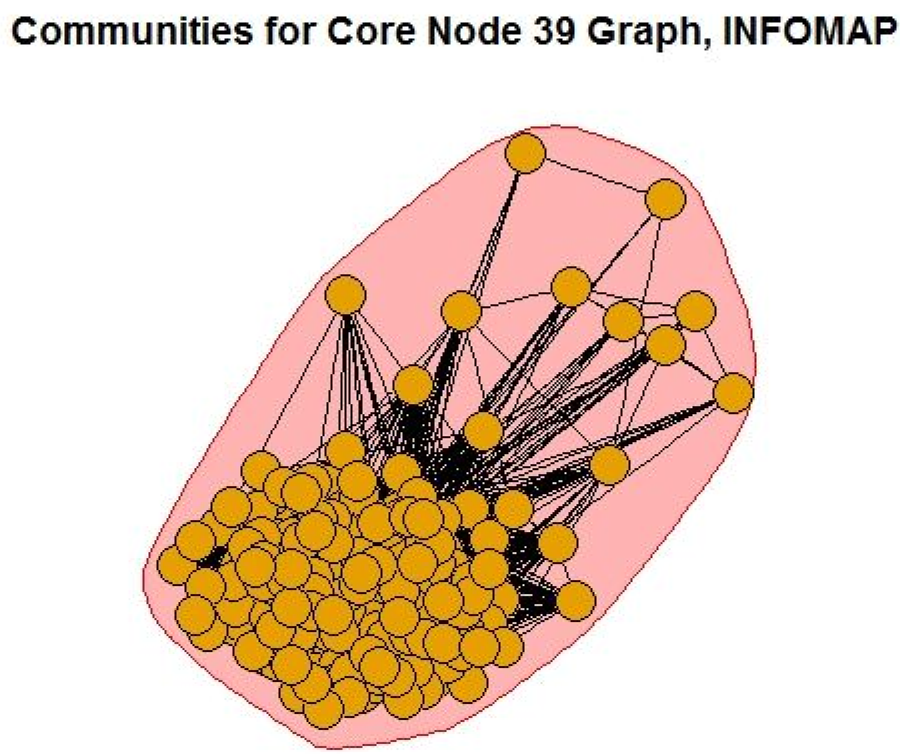


Fig 12. community structure of personal network of core node 39(Infomap)

Results

Fast-greedy community detection is a fast algorithm and finds communities of even sizes, whereas edge-betweenness community detection is a very slow algorithm and finds communities of uneven sizes. The infomap algorithm failed to detect any Community of the core 39 personal network.

The modularity score for each algorithm is tabulated as below.

Algorithm	Modularity Score
Edge-Betweenness	0.001122465
Fast-Greedy	0.3
Infomap	0

Exercise 4

Community structure with core node removed

In this exercise we will be analyzing the sub-community structure of the core personal network where the core node is removed. We chose the node 39 to compare the community structures of personal network with core node and without the core node. We also analyzed the community structure using the detection algorithms discussed in the above exercise, namely, Edge-Betweenness, Fast-Greedy and Infomap.

Community detection algorithm for Core node 39 removed

We observed that the personal network of the core node 39 had no effect when the core node was removed. The figure 13, below depicts the results of Fast-Greedy community detection algorithm on the personal network of core node 39 with core node and with core node removed. Same results were also obtained for other community detection algorithms.

The personal network of core node 39 has an edge density of 0.7 and is hence dense. Therefore most of the nodes are connected strongly to each other and most of their connection is direct and not via the core node. As a result, removing the core node does not affect the community structure much. Hence, we received almost similar results for all the community detection algorithms.

Hence for this exercise we selected a node with low edge density. We analyzed the effect of core node on community detection algorithm for core node 1, which has an edge density of 0.04.

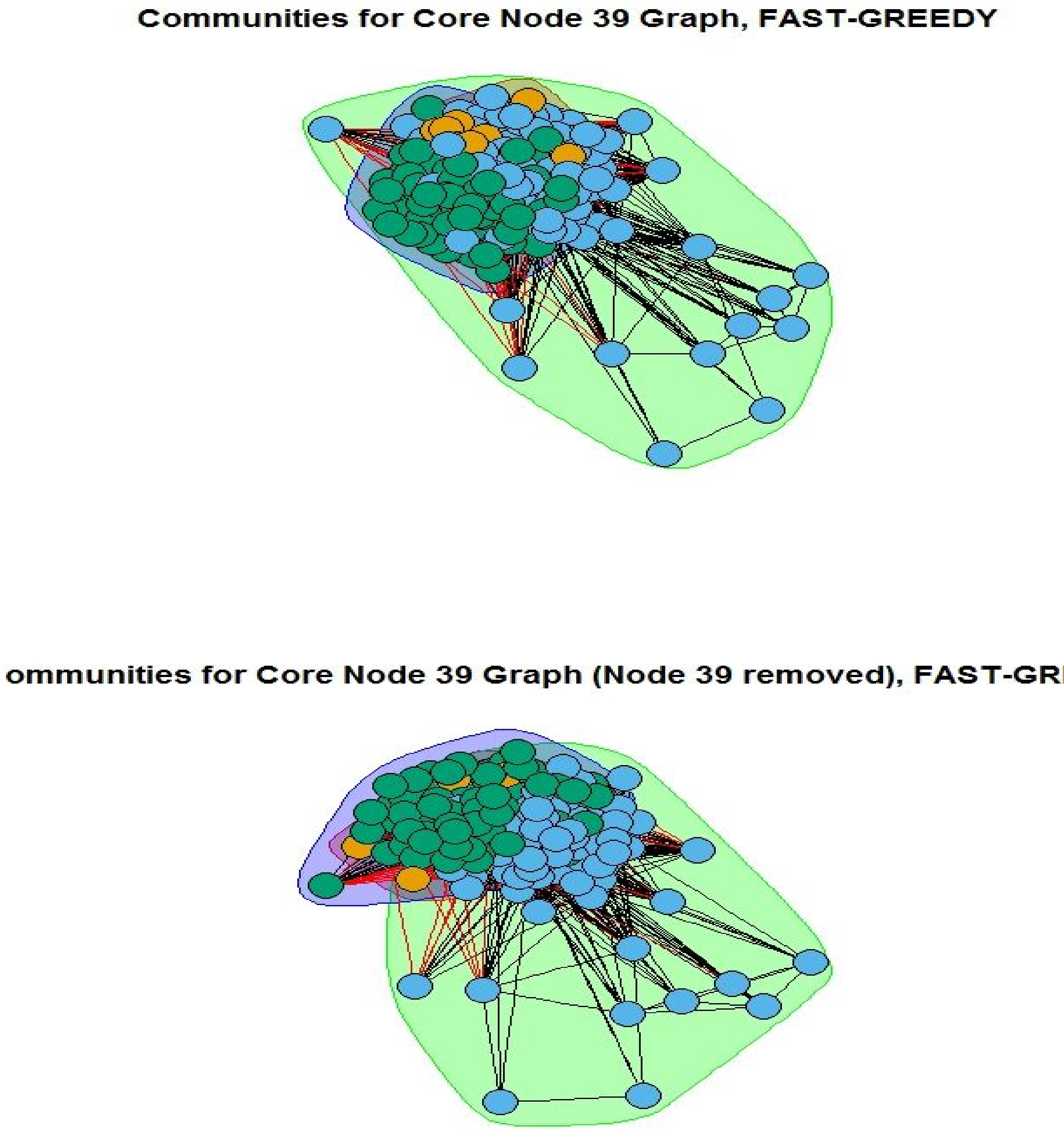


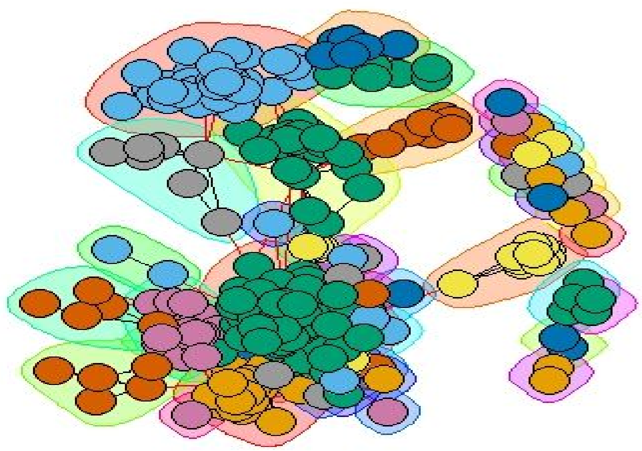
Fig 13 Fast-Greedy community structure of the personal network of core node 39, with core node and with core node removed.

Community detection algorithm for Core node 1 removed

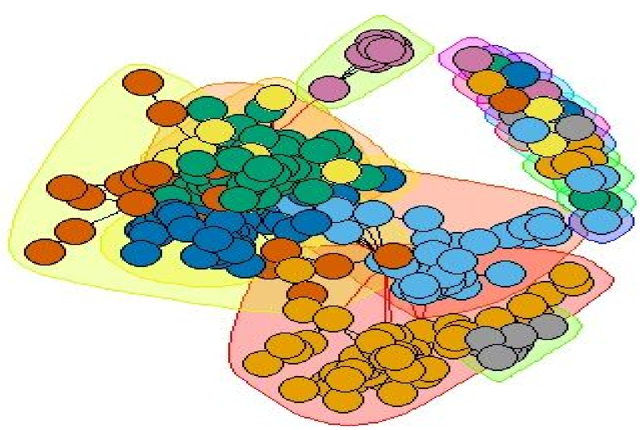
We created the personal network of the core node 1 with core node removed and used the community detection algorithms to analyze the community structures. Later we observed the effect of core node on these detection algorithms.

The plots below depicts the community structures of node 1 using different detection algorithm..

ımunities for Core Node 1 Graph (Node 1 removed), EDGE-BETWEI



Communities for Core Node 1 Graph (Node 1 removed), FAST-GRE



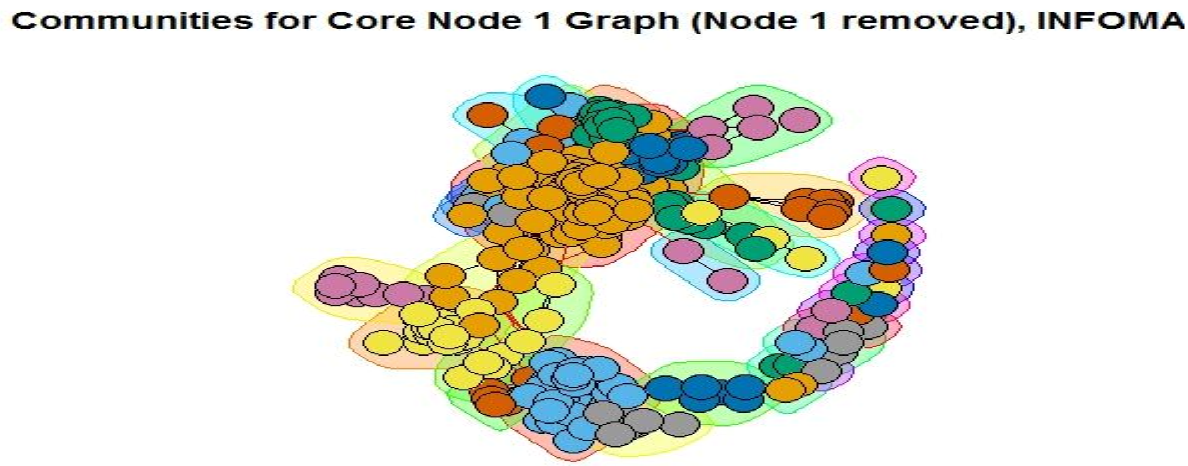


Fig 14 Communities detected by different algorithms on the personal network of core node 1, where the core node is removed.

Observation

Unlike core node 39, in core node 1 we observed a difference in the detection algorithm results when the core node was removed. The results of the two networks are tabulated as below.

Algorithm	Modularity Score (with Core Node)	Modularity Score (without Core Node)	No. of community (with core Node)	No. of community(without core Node)
Edge-Betweenness	0.41	0.44	8	26
Fast-Greedy	0.35	0.418	41	50
Infomap	0.389	0.41	26	40

The core node 1 personal network has edge density of 0.04 and hence it is a very sparse network. Therefore most of the nodes are not directly connected to each other and most of their connection is indirect and via the core node. Hence removing the core node separates them into more communities and as a result there is an increase in the number of communities. As a result, removing the core node does affect the community structure in this scenario.

Exercise 5

Facebook Social Ties

Facebook has been an integral part of society. Every account/user has to sign up by giving some of the personal information, such as,age, education, relationship status etc. In this exercise we used the facebook database network to analyze the social ties of the users.

Embeddedness:- The embeddedness of the network refers how strongly the nodes are connected or constrained in the network. Intuitively we can infer that if degree of the network is high then embeddedness of the network is high. In facebook, embeddedness can idealized as the number of mutual friends.

Dispersion:- Dispersion was coined by the authors of [1]. Dispersion is the measure to which the two nodes are dispersed. If the neighbors of the two nodes are not well-connected then the dispersion is high. In facebook, dispersion can be idealized as the embeddedness of the mutual friends of two users.

The authors of [1] analyzed a facebook network to determine possible romantic partner of an user in the network. The paper states the possible romantic partner of an user should have high dispersion to embeddedness ratio. Hence, in facebook the possible romantic partner of someone is the person in the friendlist, who is not well-connected [Low embeddedness] and their mutual friends are also not well-connected [High dispersion].

We used the following algorithm to calculate the embeddedness and dispersion of personal networks in our facebook dataset.

1. Generate personal network of each core nodes (node with more than 200 neighbors) with the core node removed.
2. Select one node from the personal network, and remove all the links of that node.
3. Degree of the resulting network of step 2 is the embeddedness of the network.
4. Calculate the distance between each pair of nodes in the resulting network of step 2. As the network is unweighted so the distance between node is assumed to be 1.
5. If the distance is more than or equal to 2, then update the distance as 0, Else update the distance as 1 and then calculate the average of the distance for all the pair in the network.
6. Represent the node with the highest dispersion to embeddedness ratio, as well as the nodes with maximum embeddedness and dispersion.
7. Repeat steps 2 to 5 for each node in the personal network.
8. Repeat the above steps for all the core networks.

We followed the algorithm and calculated the dispersion and embeddedness value of all the core nodes. The plot below depicts the histogram plot of dispersion and embeddedness.

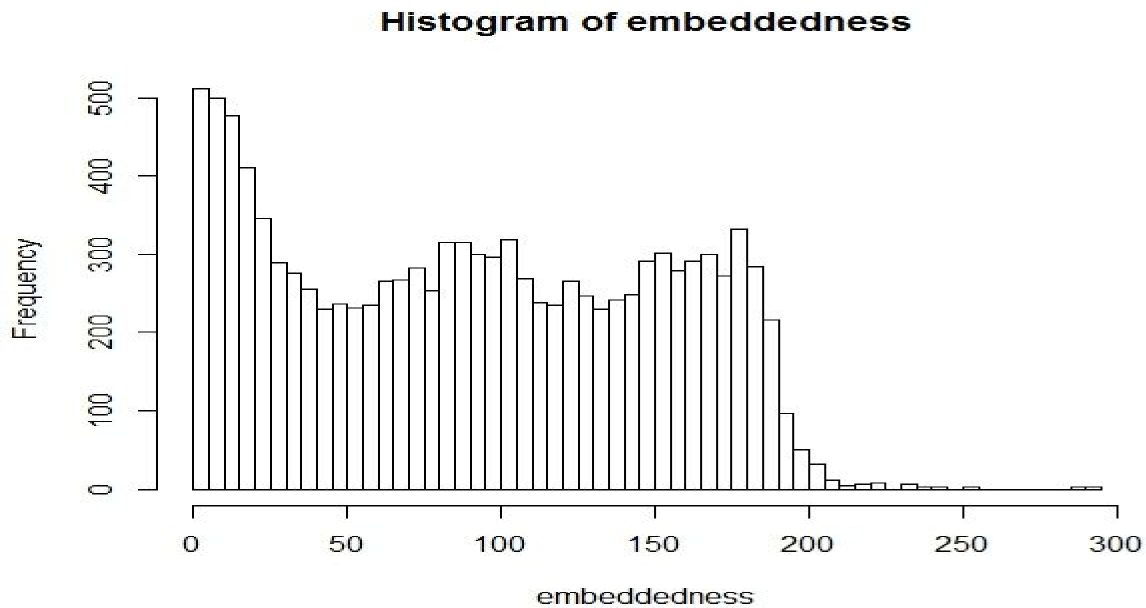


Fig 15 Histogram for embeddedness

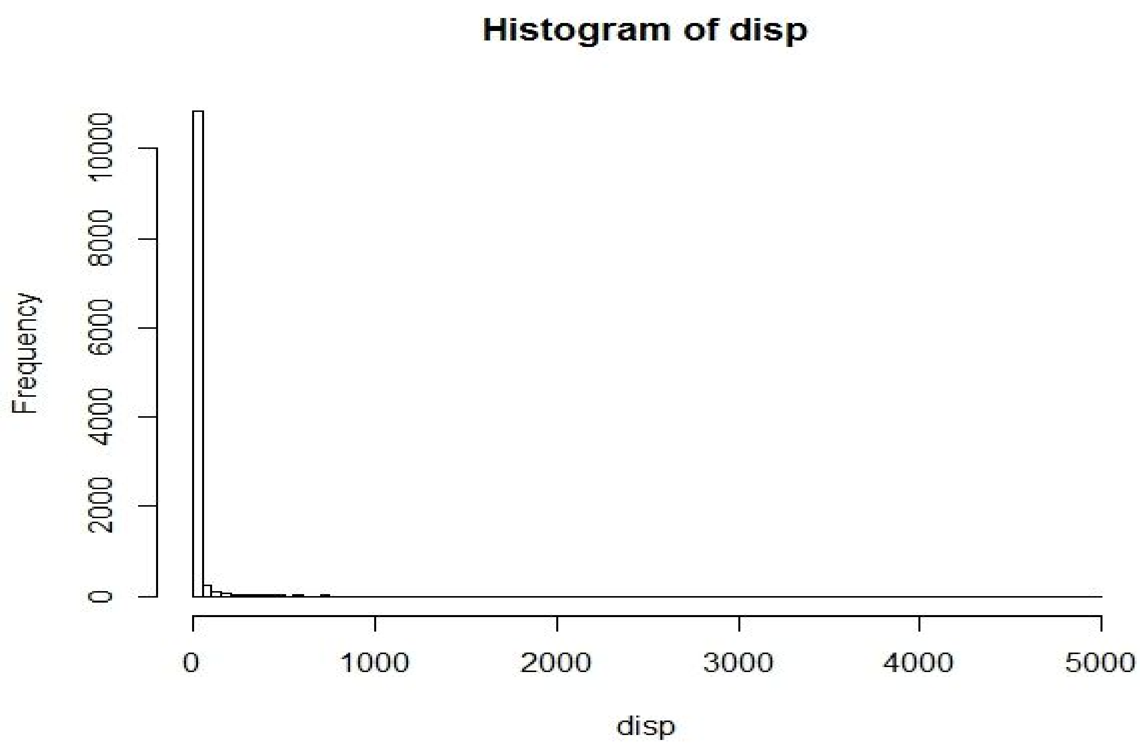


Fig 16 Histogram for dispersion

Characteristics of nodes with maximum dispersion and embeddedness

We also used the core node 1,3 and 7, to analyze the characteristics of the embeddedness and dispersion in a personal network. We found the node with highest embeddedness, dispersion and dispersion to embeddedness ratio in each of the core node's personal network.

Node with maximum embeddedness: The node with maximum embeddedness signifies that the node is deeply embedded in the network. We can infer that the node might be connected to all other neighbors of the core node's personal network. In facebook, we can idealize as the friend in the user's friend-list who has maximum number of mutual friends. He/She might be a high -school friend or college friend where they share a common ground of the friendship network.

Node with maximum dispersion: The node with maximum dispersion signifies that the node is dispersed, i.e. the mutual neighbors of the node and the core node are not well connected. We can infer that the mutual neighbors of the node and core node cannot be accessed without the said nodes. In facebook, we can idealize as the friend in the user's friend-list whose mutual friends are not well connected. He/She share a some common friends who do not share a common ground of friendship network.

Node with maximum dispersion to embeddedness ratio: The node with maximum dispersion to embeddedness ratio signifies that the node has high dispersion and low embeddedness. The node has less mutual neighbors with core node and the mutual neighbors are also not well connected. In facebook, we can idealize as the friend who has very less mutual friend and highly dispersed mutual friends. Hence, we assume that node to be potential romantic tie.

The figures below the personal network of core node 1,7 and 13, where the blue node is the core node, black is the node with maximum embeddedness, red is the node with maximum dispersion and green is the node with maximum dispersion to embeddedness ratio.

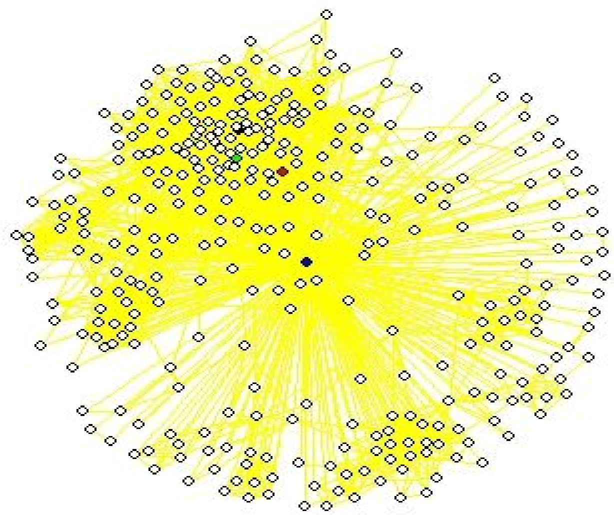


Fig 17 Personal network of core node 1 with highlighted nodes

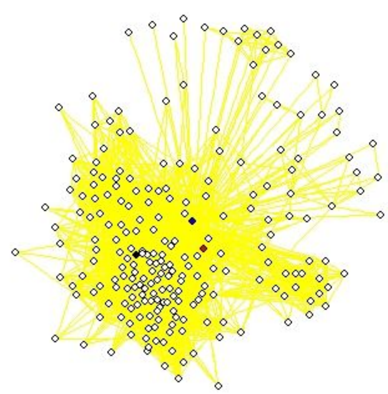


Fig 18 Personal network of core node 3 with highlighted nodes

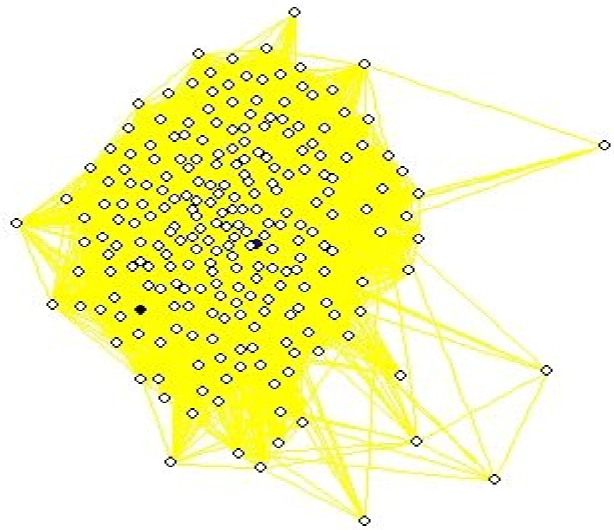


Fig 19 Personal network of core node 7 with highlighted nodes

Community structure of three personal networks

We selected the core nodes 1, 3 and 7 to analyze the community structure by using various community detection algorithms, namely Edge-Betweenness (E-B), Fast-Greedy (F-G) and Infomap (IM). The results of community detection algorithms on core nodes are tabulated below.

Core Node	Modularity Score (E-B)	Modularity Score (F-G)	Modularity Score (IM)	No. of communities (E-B)	No. of communities (F-G)	No. of communities (IM)
1	0.3533022	0.4131014	0.3891185	41	8	26
3	0.133528	0.2517149	0.203753	104	5	10
7	0.01232689	0.1273412	0	122	3	1

The below figures depicts the community structure of each personal network using only infomap community detection algorithm.

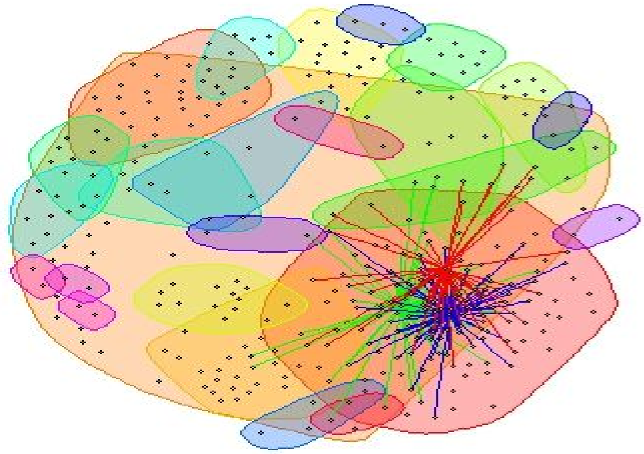


Fig 20 Community structure of personal network of core node 1[infomap].

The blue edges belong to the node with highest embeddedness, green to highest dispersion and red to highest ratio. We can see that both these nodes have edges that connect to members in different communities. In this case it would be difficult to judge based on just these scores which node is the romantic partner. However we have narrowed it down to one of two choices.

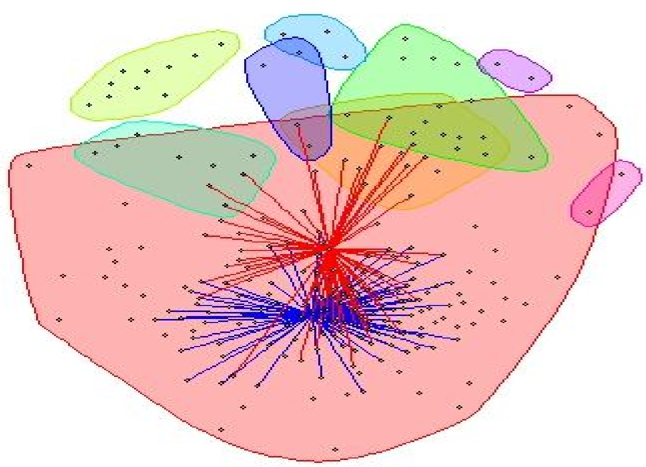


Fig 21 Community structure of personal network of core node 3 [Infomap].

The red edges originate from the node with the highest dispersion while the blue edges originate at the node with high embeddedness. Different colored nodes belong to different communities and are encircled by different colored boundaries. We can see that the node with the high value of embeddedness connects to node within one community which is the big red community. The node with high value of dispersion however, connects to members of different communities and hence confirming our suspicion about how the romantic partner acts like a bridge between members of different communities.

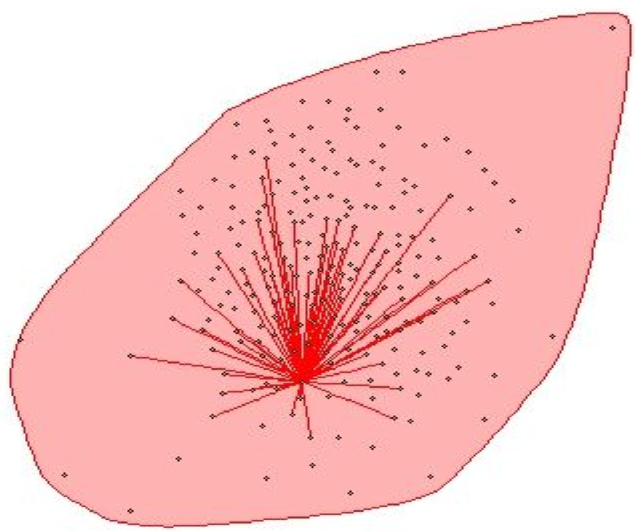


Fig 22 Community structure of personal network of core node 7 [Infograph].

The red edges originate from the node with the highest dispersion and the node with the high value of embeddedness connects to node within one community which is the big red community. This network is very well connected and hence the community detection failed. The core node neighbors are not very disperse, as a result the romantic partner detection algorithm failed to detect the node with high dispersion to embeddedness.

Exercise 6

The communities in personal network can translate into different aspects of that person's life. These can be friends from high school, college friends, colleagues, etc. There are certain types of communities just like stated above, which are present in almost every individual's personal network. In this part, we try to determine whether a community in one personal network is equivalent to another community in second personal network. In other words, we are trying to find the similarities between communities in different personal networks, so that we can map communities across different personal network.

The theme of the algorithm is to find a feature space of relevant features such as Modularity Index, Clustering Coefficient, Density, Community size and then cluster the communities across all the personal networks in this feature space. The feature space that we selected consist of following dimensions (attributes) :

1) Modularity Index
2) Clustering Coefficient
3) Density
4) Community size

We have selected above statistical analytical tools as they provide a large information gain and based on the information gain, we can find clusters of similar community easily. Other way to visualize is that, the similar communities across all personal networks have some intrinsic properties and the above statistical tools provide a wide range of these properties and represent the similar communities in the feature space.

Algorithm

1. Run fast-greedy community detection algorithm to find the communities in an individual's personal network. Perform this step for all the personal networks and find all the communities across all personal networks.
2. From a feature vector for each community of size greater than 10 (ignore communities with size less than or equal to 10) found in step 1. The dimension of the feature vector is 4 and the attributes of the feature vector are: modularity score, clustering coefficient, density and community size.
3. Once the feature space is formed, see whether each feature vector has finite values and then, run a k-means clustering algorithm to cluster the communities with the feature vectors assigned in step 2.

4. Repeat step 3 for k=2,3,4,5,6.
5. Select the value of k which gives two large clusters, such that the spread of different personal network in the clusters is maximum, that is, select the value of k which has two large clusters, representing communities from almost all the personal networks
6. For the selected value of k, output the number of networks represented by cluster 1 and cluster 2

The above algorithm found 105 communities of size greater than 10 across all 40 personal networks. Since the modularity score is defined for a given partitioning and not individual communities in that partitioning, so we assigned the communities belonging to the same personal network to have the same modularity score. Also, the clustering coefficient and density were computed for a community by forming a subgraph out of that community. The result of the above algorithm for the various values of k is listed below.

Clusters	Number of different personal network in Cluster 1	Number of different personal network in Cluster 2
2	40	3
3	40	12
4	38	12
5	**35**	**30**
6	27	20

From the above table, we can see that Clusters (k) = 5 gives the best results. Cluster 1 represents the communities from 35 personal networks (around 88%) and Cluster 2 represents the communities from 30 personal networks (around 75%). Hence, two types of clusters are recurring across almost all personal networks.

We also found the communities belonging to cluster 1 and cluster 2 for each personal network. The output is tabulated below :

Personal Network	Community ID belonging to cluster 1	Community ID belonging to cluster 2
1	0	1,2
2	3	0
3	2	1
4	0	2
5	3	4
6	1,2,3	0
7	1	2
8	0	1,2
9	2	3
10	2	1
11	1	2
12	2,3	0
13	2	1
14	2	3
15	2	1
16	3	2
17	3	2
18	3	2
19	1,2	3
20	0	1,2
21	7	2

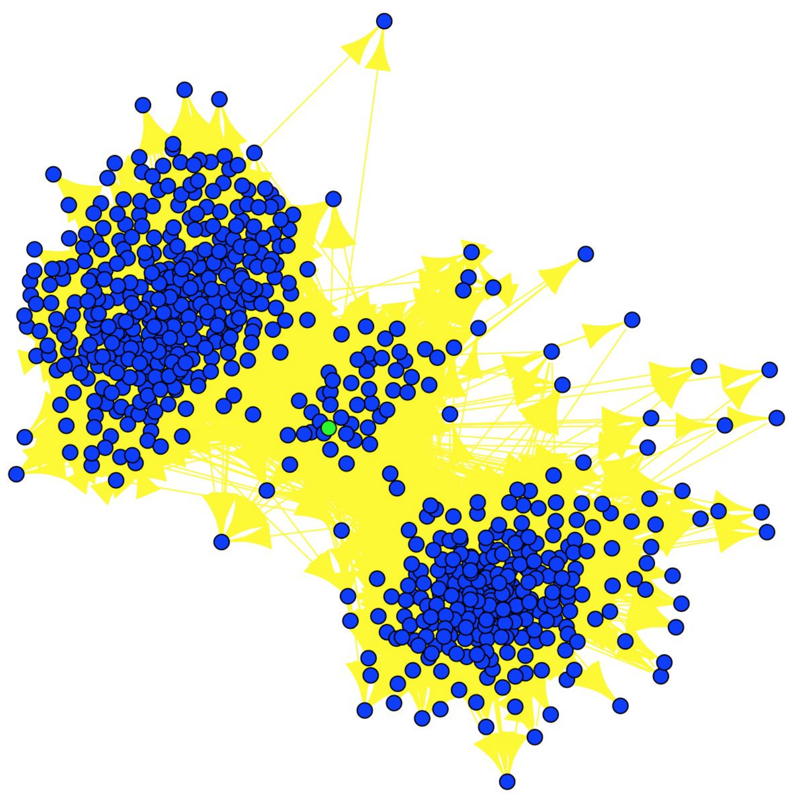
22	1	2
23	3	2
24	2	3
25	0	1,2
26	1	3
27	2	1
28	1,3	0
29	2,3	0
30	1	2
31	1,2	0
32	2	1
33	2	1
34	1	2
35	2,3	0
36	1	2
37	2,3	0
38	1,2	0
39	2,3	0
40	1	2

Exercise 7

Analysing the social network: Google+network

Google+ ,unlike Facebook, has a directed network structure, where you can have someone in your circles regardless of whether they have you in their circles or they don't. Circles are tags you put on your relationships when you add people.
E.g. you can have two circles, one named "friends" and the other one named "family", and when you add someone you can put them in one or both of these circles.

Personal Network(Properties):
In this part we formed the personal network of users who have more than two circles. The network was created by reading in the edge list files for the users with more than two circles. We know that the edge list file do not contain the *Core node*, and hence we created the same adding directed edges to all the nodes in the edge list (assuming the core node follows all users whose node IDs are present in the edge list file). Here we plotted the personal network of user "118379821279745746467" for visualization purposes. The plot is given below:



In the above plot, the core node is colored red and all the other nodes in its personal network is plotted green. The directed edges between nodes are colored yellow. The personal network plotted above has 695 nodes (including the core node) and 37404 edges. Therefore, the edge density of the personal network is 0.07 and the personal network is sparse. The diameter of the personal network is 6.

Personal networks(Community Structure):

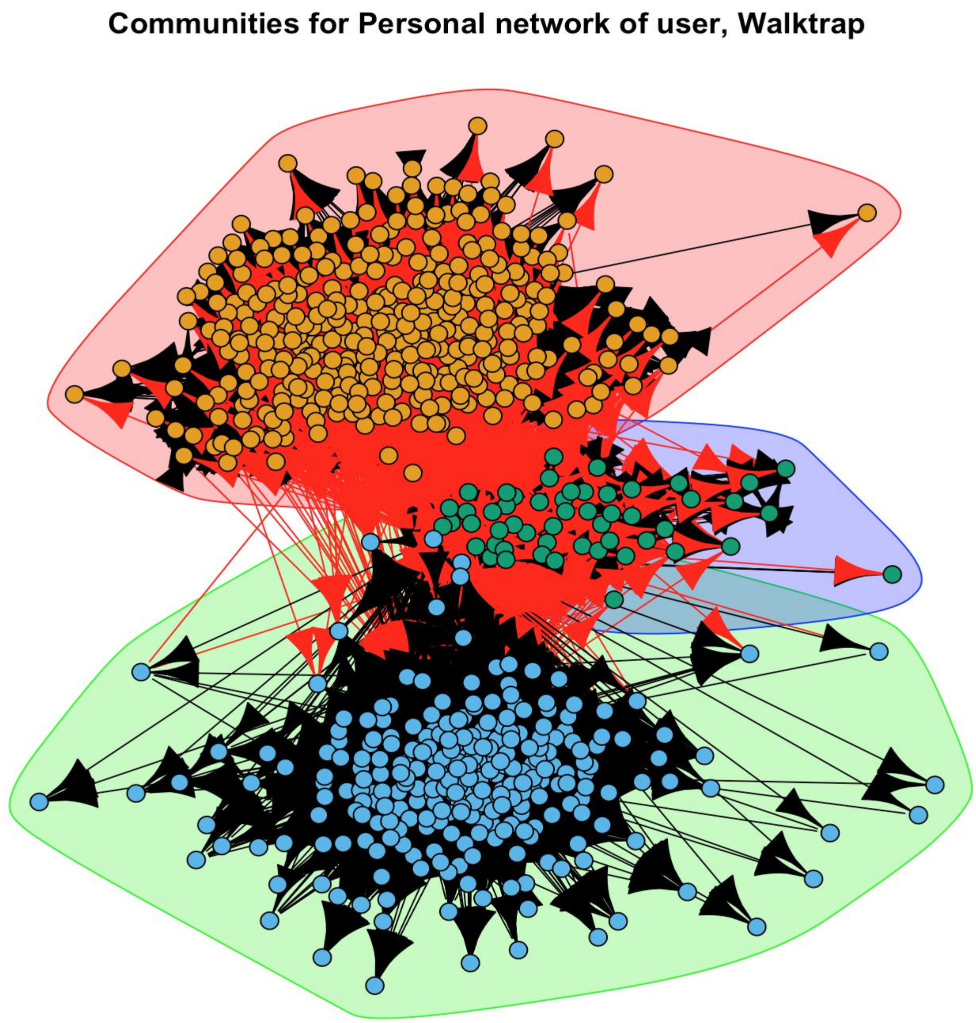
This part includes understanding Community structures of the core nodes of Personal networks using two community detection algorithms as below:

- Walktrap community detection
- Infomap community detection

Walktrap community detection

It is an approach based on random walks. The core idea is that if random walks are performed on the graph, then the walks are more likely to remain within the same community since there are only a few edges that lead outside a given Community. Walktrap runs short random walks of 3-4-5 steps (depending on one of its parameters) and uses the results of these random walks to merge separate communities in a bottom-up manner.
It is a bit slower than the *fast greedy approach*(bottom-up approach) but also slightly more accurate (according to the original publication).

The Walktrap Community Detection algorithm was run on personal network of user "118379821279745746467".The algorithm found 3 communities. The community structure of the personal network is shown below:



In the above plot, the edges are colored depending on their type and nodes according to their community membership.

For example, the nodes are colored as *blue, green and orange*. The intra-community edges are colored black and the inter-community edges are colored red. It can be seen that most of the nodes are colored blue or orange, implying that most of the nodes in core nodes personal network belong to either of the two communities. Further,the modularity of the partitioning is low (0.4787) as there are lots of edges connecting communities (red edges). The community id's and their sizes were computed and are tabulated below:

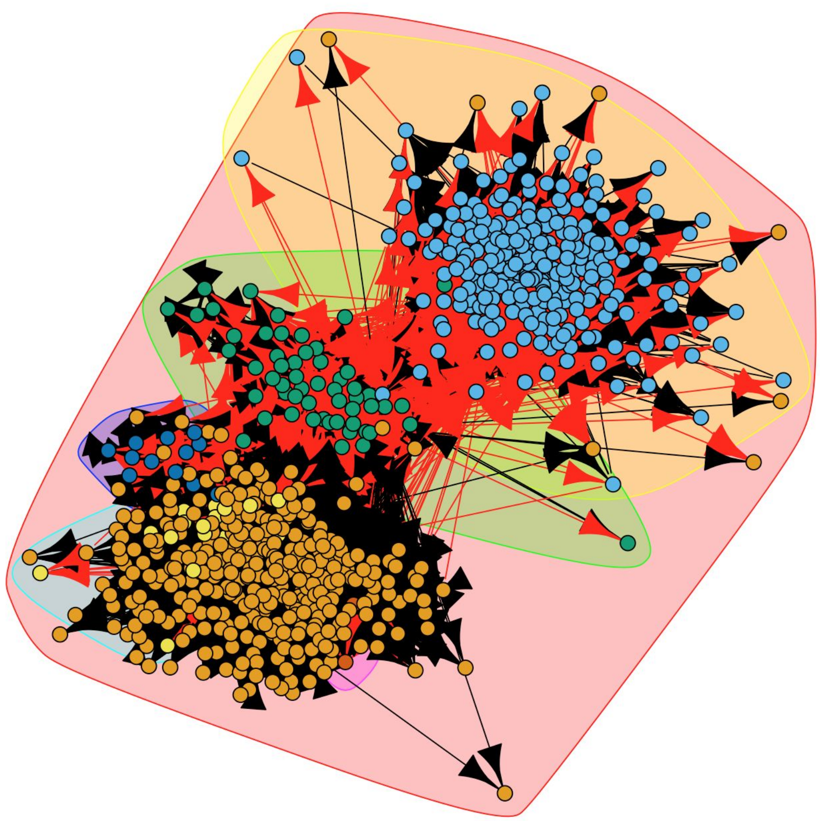
Community ID	Community size (# of nodes)
1	363
2	280
3	52

Infomap community detection

Infomap community detection algorithm is based on info theory principles. It tries to build a grouping which provides the shortest description length for a random walk on the graph , where the description length is given by expected number of bits per vertex required to encode the path of a random walk.

The infomap community detection algorithm was run on personal network of user “118379821279745746467”. The algorithm found 6 communities. The community structure of the personal network is shown below:

Communities for Personal network of user, Infomap



As shown above the nodes are colored according to their community membership and the edges are colored depending on their type. For example, the nodes are colored as *blue, green, yellow, orange, etc*. The intra-community edges are colored black and the inter-community edges are colored red.
As majority number of nodes are colored *blue or orange*, it can be safely concluded that most nodes in core nodes personal network belong to either of the two communities. Also, since there are lots of edges connecting communities (red edges), so the modularity of the partitioning is low (0.4635).The community id's and their sizes were computed and are tabulated below:

Community ID	Community size (number of nodes)
1	341
2	269
3	55
4	16
5	12
6	2

Communities and Circles overlap:

Here we evaluate the overlap between the circles and communities of personal networks. The circles information of a user's personal network is considered to be the ground truth, and we try to come up with an algorithm that helps to compute the overlap between the community structure and the circles. A large overlap will mean that the community detection algorithm is successful in finding the circles in an user's personal network. The algorithm is given below

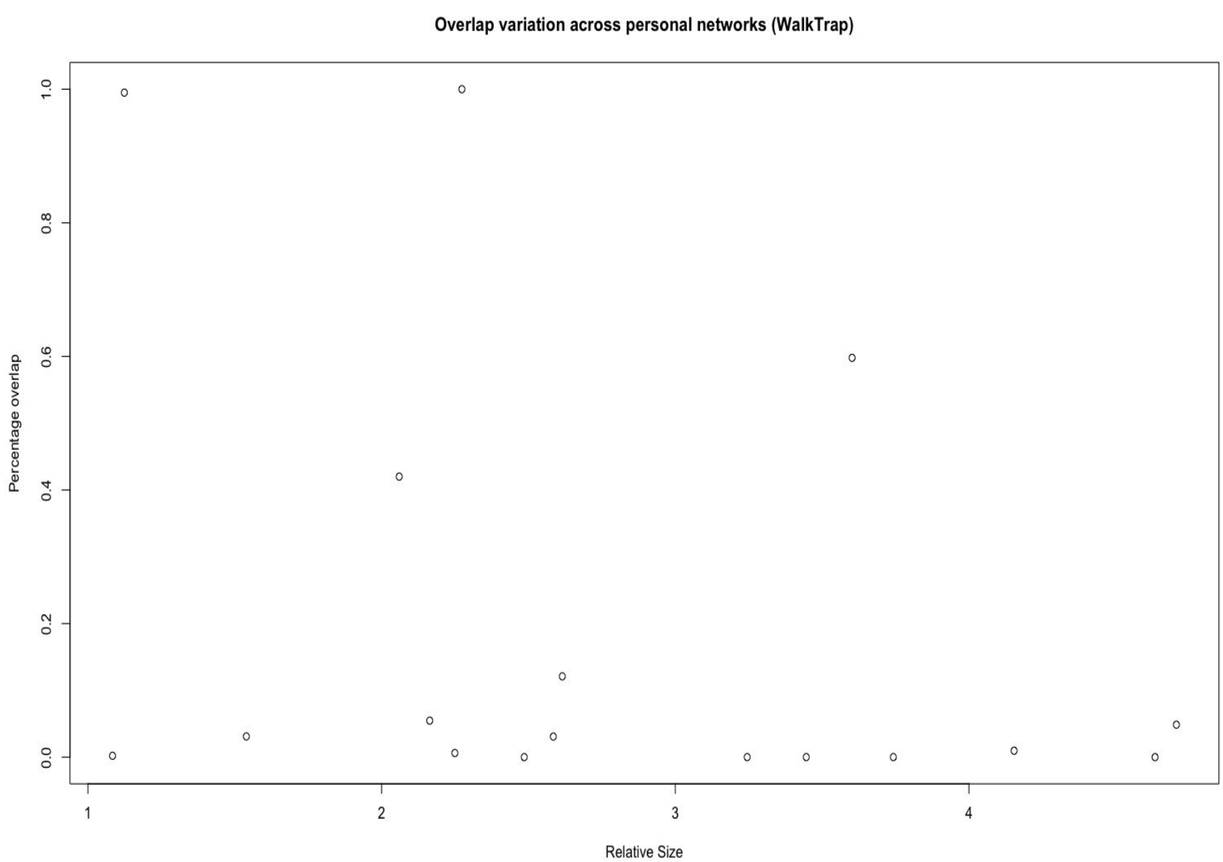
Algorithm

1. For a given personal network, find the communities in the network using one of the two community detection algorithms (Walktrap or Infomap)
2. Store the node ID's belonging to the largest community found in step 1
3. Store the node ID's belonging to the largest circle (from the circles file)

4. Compute the number of nodes that belongs to both the largest community and largest cluster
5. Divide the number of nodes (found in step 4) by the minimum [community size (found in step 2), circle size (found in step 3)]. This ratio is defined to be the percentage overlap between communities and clusters
6. Divide the community size (found in step 2) by the circles size (found in step 3). This ratio is defined as the relative size
7. Repeat steps 1 to 6 for all the personal networks

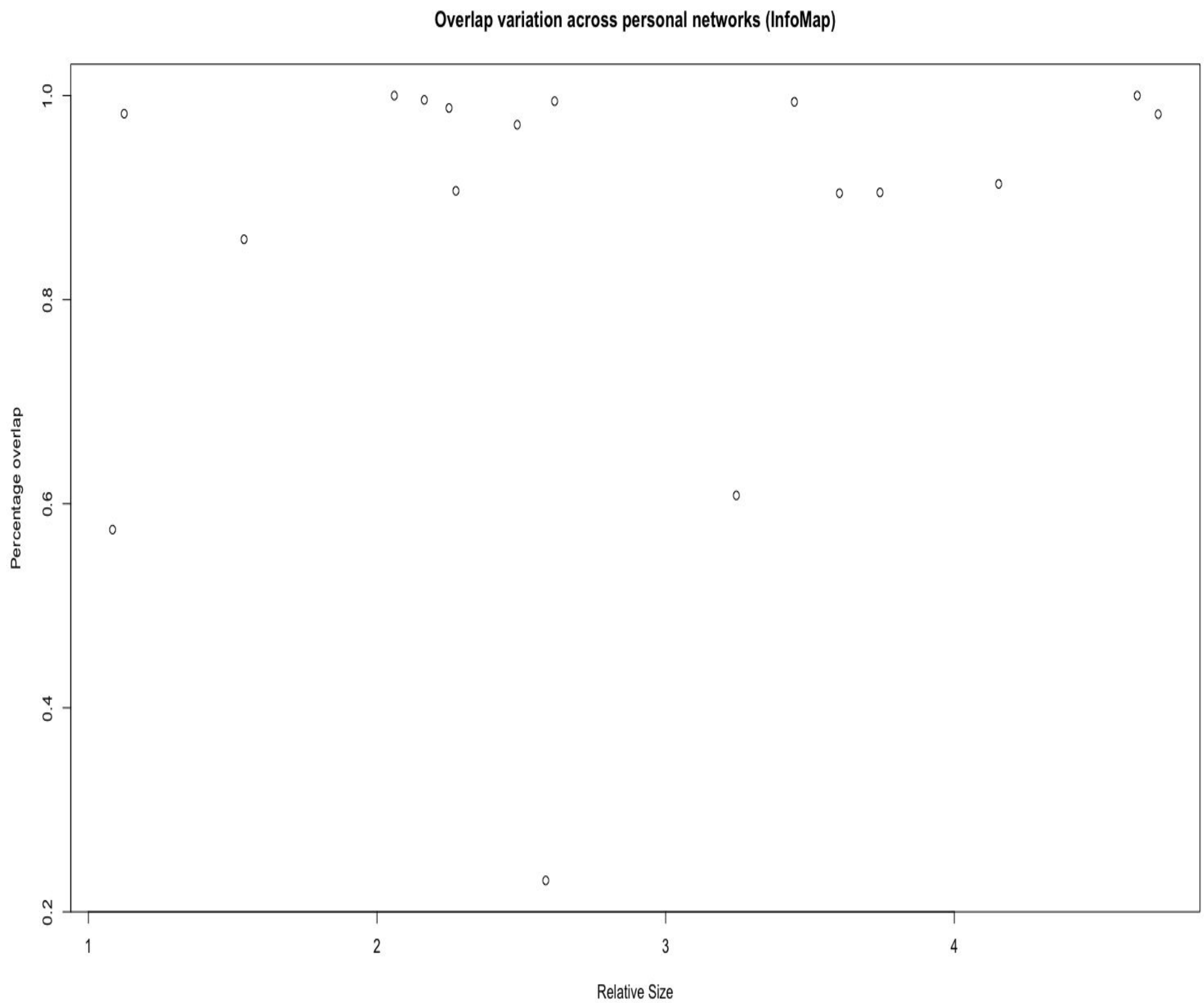
Results

We ran the above algorithm with *walktrap community detection*. The percentage overlap is plotted against relative size. The plot is given below



From the above plot we can see that the communities found by the walktrap community detection algorithm do not overlap with the circles of the users for most of the personal networks. However, there are very few users for whom the percentage overlap is significant.

Now running above algorithm with *Infomap community detection*. The percentage overlap is plotted against relative size. The plot is given below



From the above plot we can see that the communities found by the infomap community detection algorithm overlaps strongly with the circles of the users for most of the personal networks. There are very few users for whom the percentage overlap is insignificant.

Comparison

From the above plots, we can see that infomap community detection algorithm is most successful in finding the communities which overlaps with the circles (ground truth). Also, the overlapping pattern varies across users. This is due to the fact that we are very specific in adding people to circles, but the community detection is based on the interconnection between nodes. For example, we might form two circles, one for school friends and one for college friends. However, community detection might find one large community of friends since school and college friends might be strongly interconnected between them. Therefore, we found that the community sizes are generally larger than the circle sizes.

Conclusion

In this project we used a social network to analyze user's personal network and analyze its community structure. We used the community detection algorithm to study some fundamental elements of a personal network. In the first part of the project we dealt with a facebook social network and analyzed the personal network of the nodes with more than 200 neighbors. We briefly studied the characteristics of a social network and designed an algorithm for finding potential romantic partner. Lastly we designed an algorithm to find the clusters in a personal network. These clusters refers to particular friend network of the user such as highschool friends, office coworkers etc.

In second part of the project we dealt with google+ network, which is a directed network, unlike facebook. In the Google+ network, our main objective was to find overlapping between the communities and circles, and the overlapping pattern across users. We came up with a systematic method for finding the overlap and also visualized its variation across users. We found Infomap community detection to give optimal results.

References

[1] Backstrom, Lars, and Jon Kleinberg. "Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on facebook." *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 2014.
[2] Backstrom, L., Kleinberg, J. M., Lee, L., and Danescu-Niculescu-Mizil, C. Characterizing and curating conversation threads: expansion, focus, volume, re-entry. *Proc. 6th ACM WSDM*, 2013.
[3] Cornwell, B. Spousal network overlap as a basis for spousal support. *J. Marriage and Family 74* (2012).
[4] Jones, J. J., Settle, J. E., Bond, R. M., Fariss, C. J., Marlow, C., and Fowler, J. H. Inferring tie strength from online directed behavior. *PLoS ONE 8* (2013).