EE232 Project1

Name: DIJIA FAN UID: 704588291
Name: Qinyi Yan UID: 704406413
Name: Lele GAO UID: 494395468

In this project, we are going to study a real world social network from Facebook. In detail, the network we will discuss is a specified user's personal friendship network, which means the graph(i.e. the personal network) only consists of the node and its neighbors meanwhile the edges have both ends within this set of nodes. Understanding the structural roles of significant people in online social network neighborhoods is a broad question that requires a combination of different approaches[1]. We download the dataset which is stored in the edgelist file *facebook_combined.txt*, and find that there are 4032 nodes and 88234 edges in this Facebook graph. Besides, it is connected but not directed. In the following, we will carry out corresponding experiments to probe into the characteristics inherent in both this graph and one of the personal friendship networks.

Question 1

After we downloaded the edgelist file, we summarized some statistical properties of the Facebook graph which is shown in table 1.

Table 1

Properties	**Value**
No. nodes	4032
No. edges	88234
Connected	TRUE
Diameter	8
Average Degree	43.69101
Directed	FALSE

Next, we draw the degree distribution of the graph in a barplot which is displayed in Fig 1.

In order to fit the distribution of degrees, we have to put all the bars in Fig 1 on a curve which is shown in Fig 2.

In the following, we must choose a model that best matches the curve. As can be seen in Fig 2, the degree distribution is most likely to conform to a power-law

[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.

distribution, i.e. y ~ I(exp(1)^(a + b * x). For the sake of certainty, we try to fit the curve in different models including y ~ I(1/x*a) + b*x, y ~ (a + b*log(x), y ~ I(1/x*a)+b and y~I(1/x). The fitted models are shown in Fig 3.
As we can see in Fig 3, the yellow curve fits the distribution best. To display the conclusion clearly, we capture the yellow curve and show it in Fig 4.
Finally, the fitted model and it's corresponding parameters are obtained in Fig 5, namely, y=exp^(-3.594-0.029*x). We calculated the total mean squared error(MSE), and the value is 4.016458e-07.

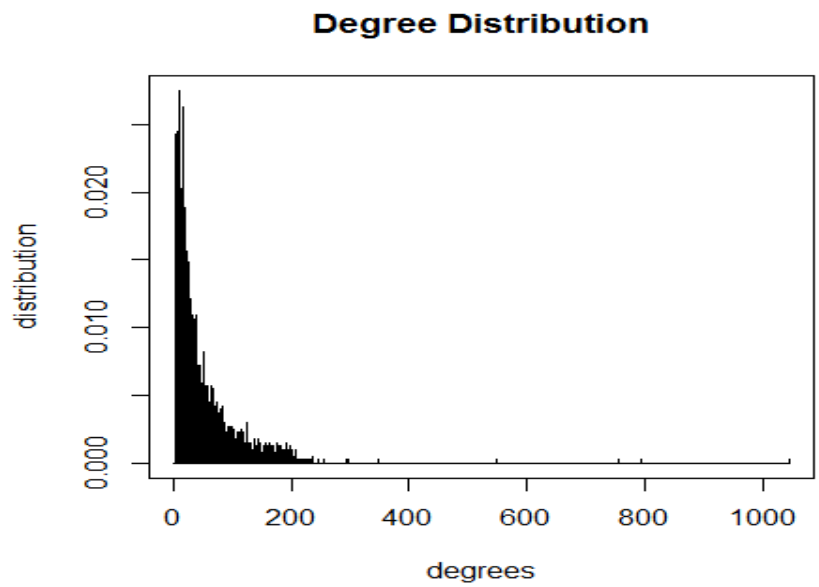


Fig 1. Degree distribution of the graph

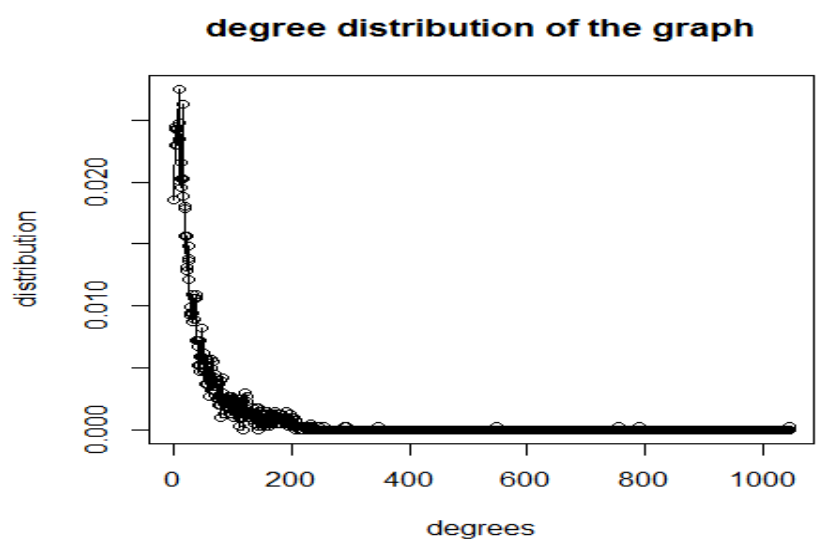


Fig 2. Fitting the distribution in a curve

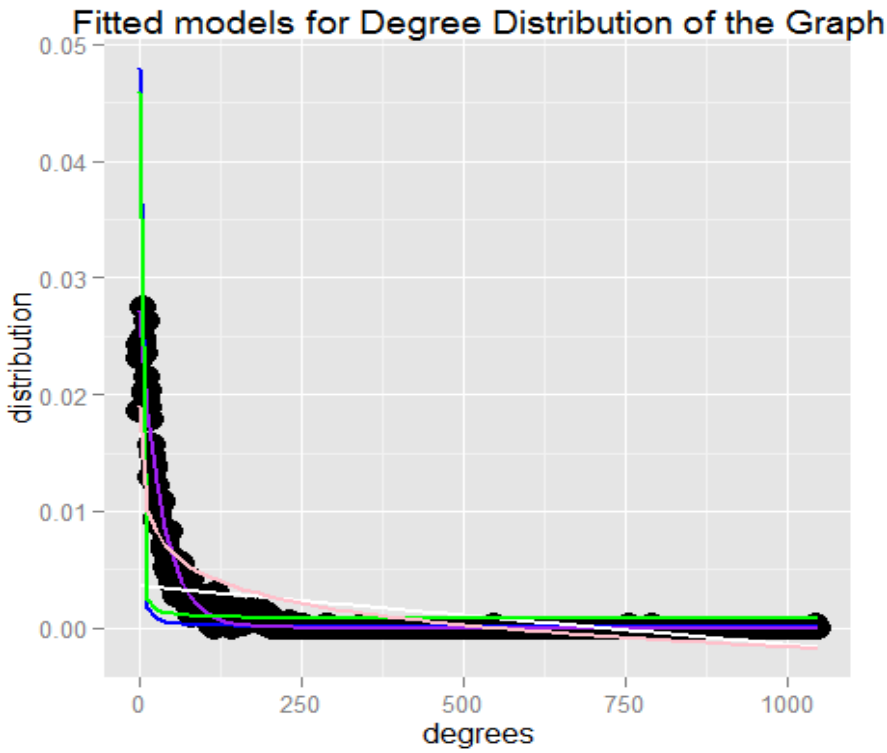


Fig 3. Fitting the distribution in multiple models

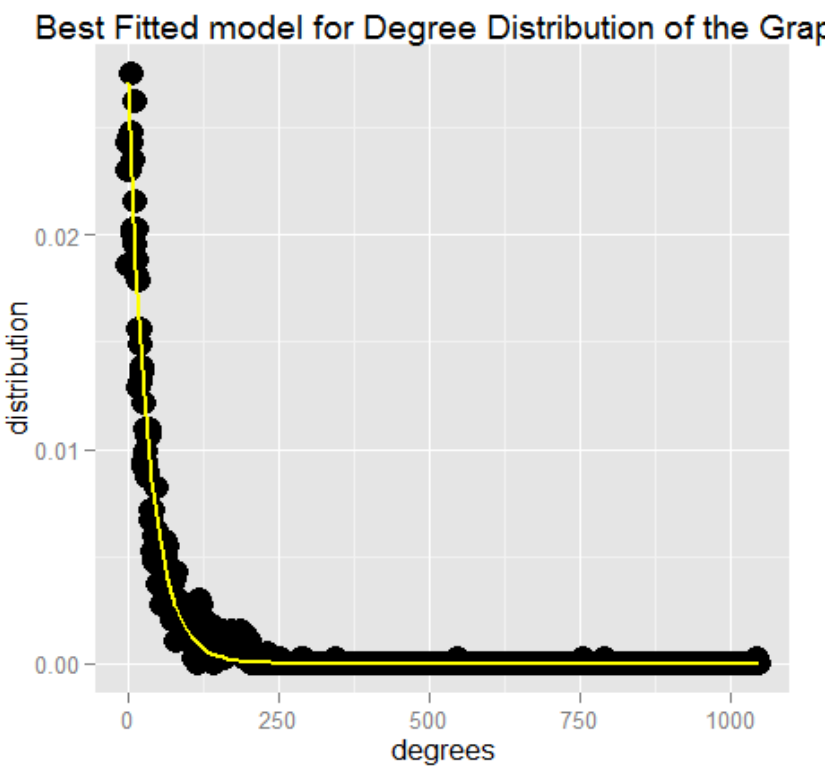


Fig 4. The best fitted model for degree distribution

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

Parameters:
    Estimate Std. Error t value Pr(>|t|)
a -3.5940060  0.0078770 -456.26   <2e-16 ***
b -0.0291487  0.0003247  -89.77   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0006339 on 1044 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 8.699e-06
```

Fig 5. Summary of the fitting model

Question 2

First of all, we have to construct the personal network of node 1. We find that, for this node, it has 347 neighbors. We obtain the subgraph and count up the number of nodes and edges. The result is shown below.

Number of nodes: 348

Number of edges: 2866

Question 3

We may as well see what the personal network looks like. In order to display the network structure, we abstract the sungraph's communities using different colors, and the overall appearance is shown in Fig 6.

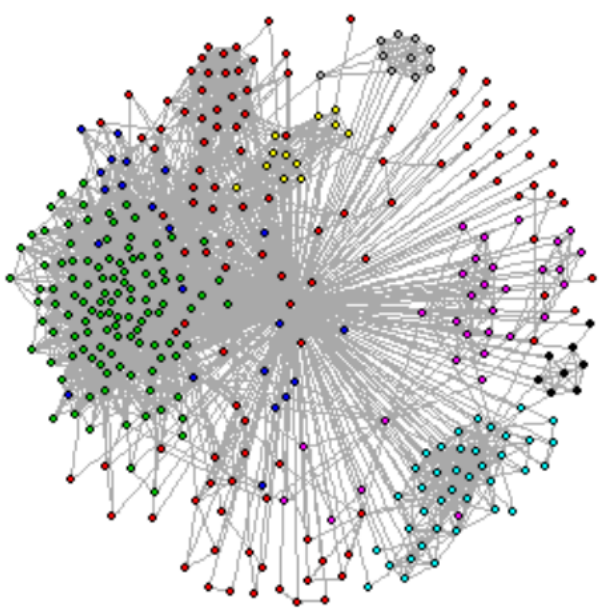


Fig 6. The personal network structure

We further try to find all the core nodes that have more than 200 neighbors in the subgraph. In the end, we get 41 core nodes in the subgraph and the average degree is 277.439.
In the next step, we select No.8 core node as our research object, and we find that there are 218 nodes in the core's personal network. The sub network of core node 8 is shown below in Fig 7.

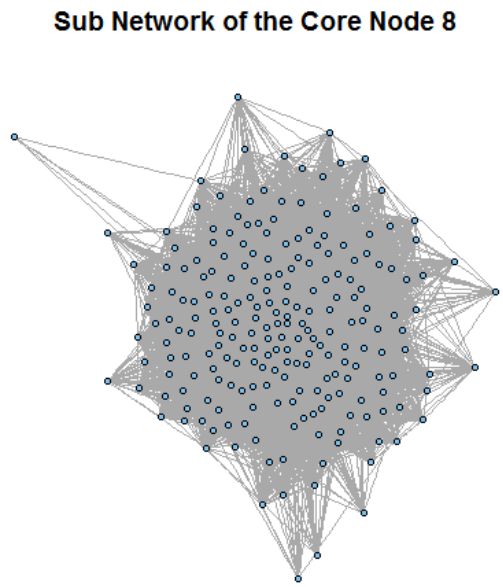


Fig 7. The personal sub network of core node 8

We then try to distinguish communities with color in the sub network.
Fig 8~10 display the detected communities using three methods

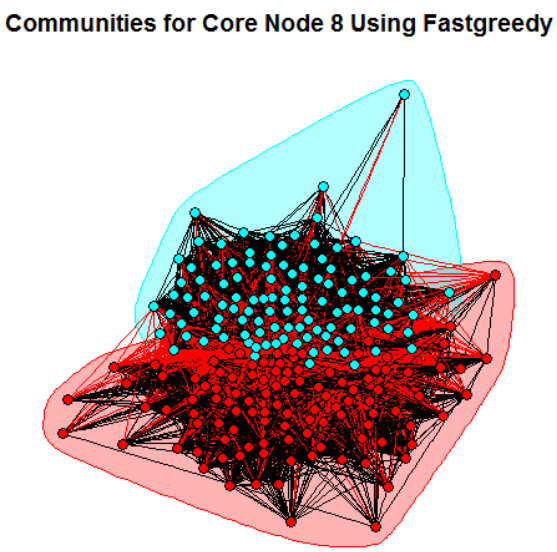


Fig 8. Communities for Core Node 8 Using Fastgreedy

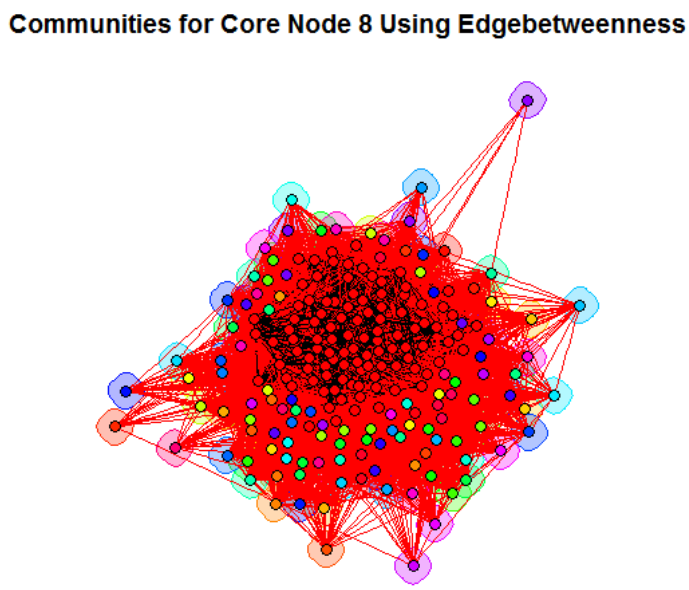


Fig 9. Communities for Core Node 8 Using Edgebetweenness

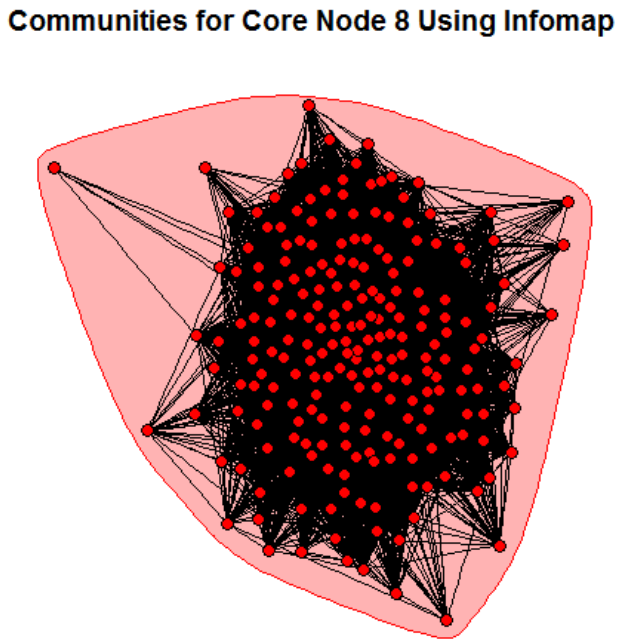


Fig 10. Communities for Core Node 8 Using Infomap

To compare the difference of the detected communities using three different methods, we summarize the number of detected communities and their modularity in Table 2 below.

Table 2

	Fastgreedy	**EdgeBetweenness**	**Infomap**
No. of communities	2	117	1
Modularity	0.1288126	0.01361108	0

To illustrate the difference visually, we plot the histograms to display the distribution of nodes among various communities.

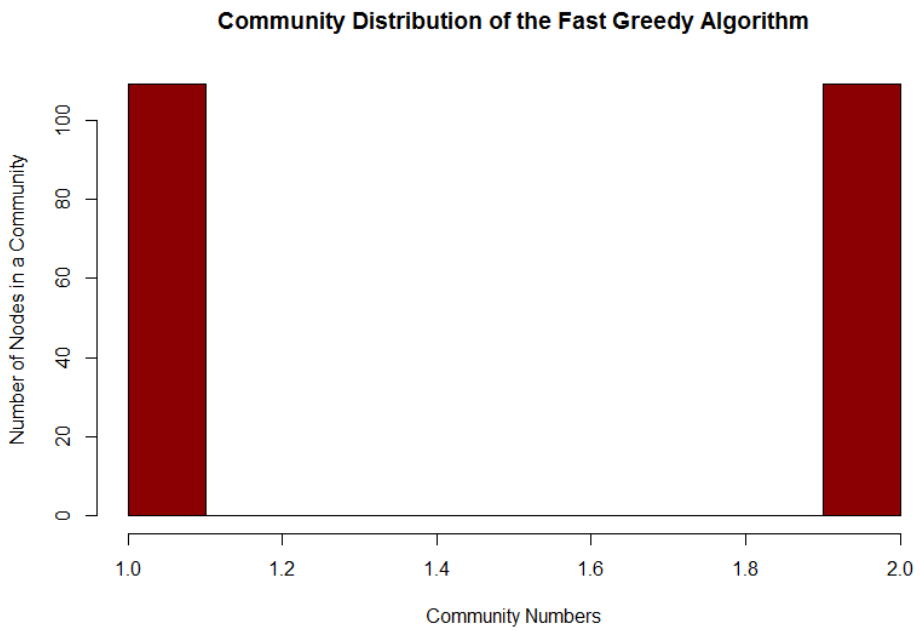


Fig 11. Community Distribution of the Fast Greedy Algorithm

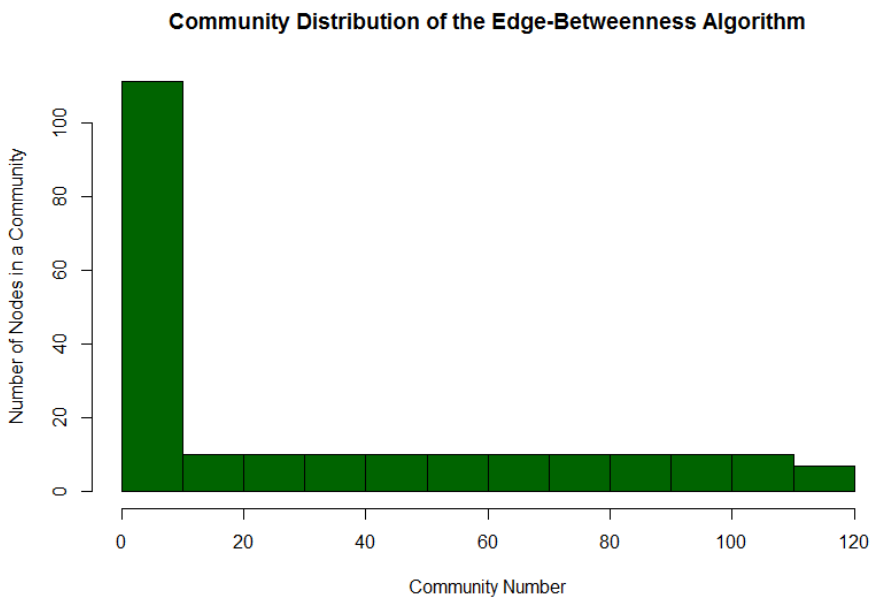


Fig 12. Community Distribution of the Edge-Betweenness Algorithm

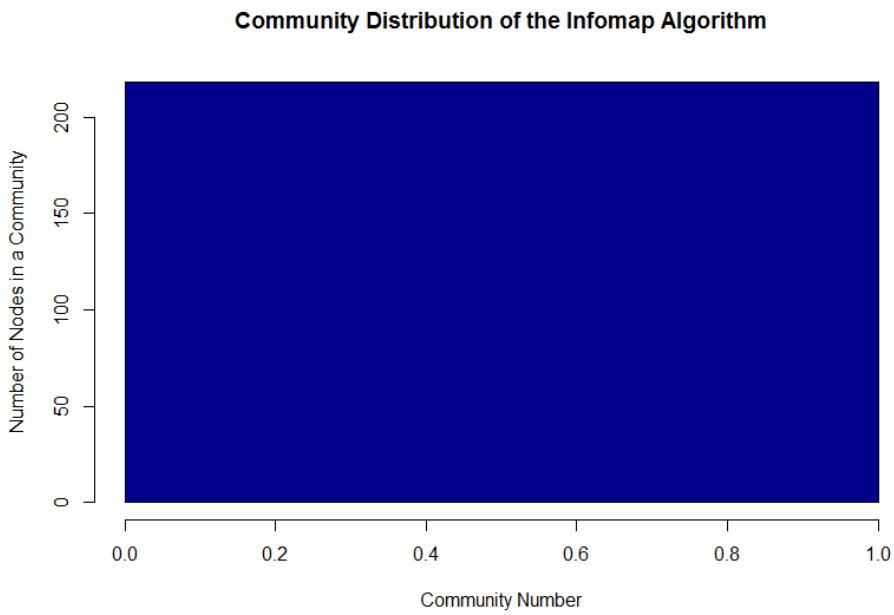
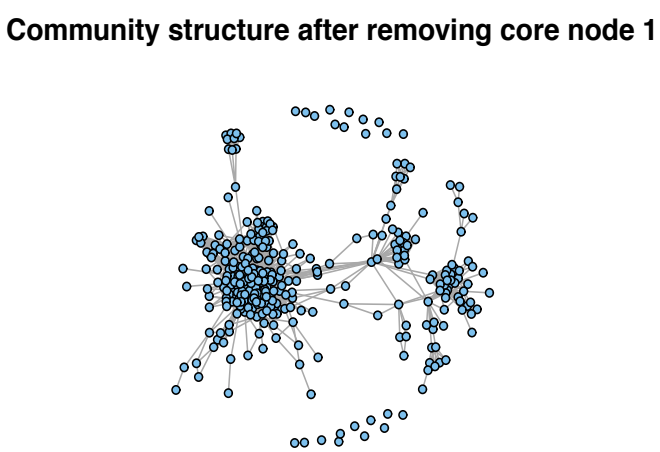


Fig 13. Community Distribution of the Infomap Algorithm

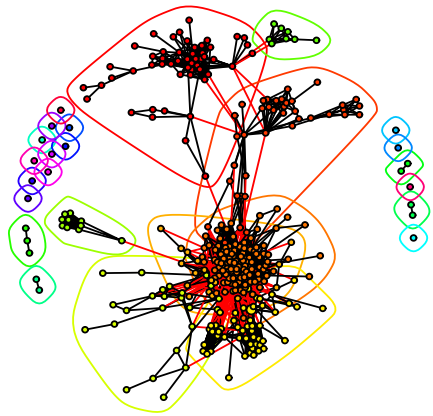
Question 4: Try removing the core node itself from its personal network and running the above community detection algorithms again. Are there any differences in the results?

Solution:

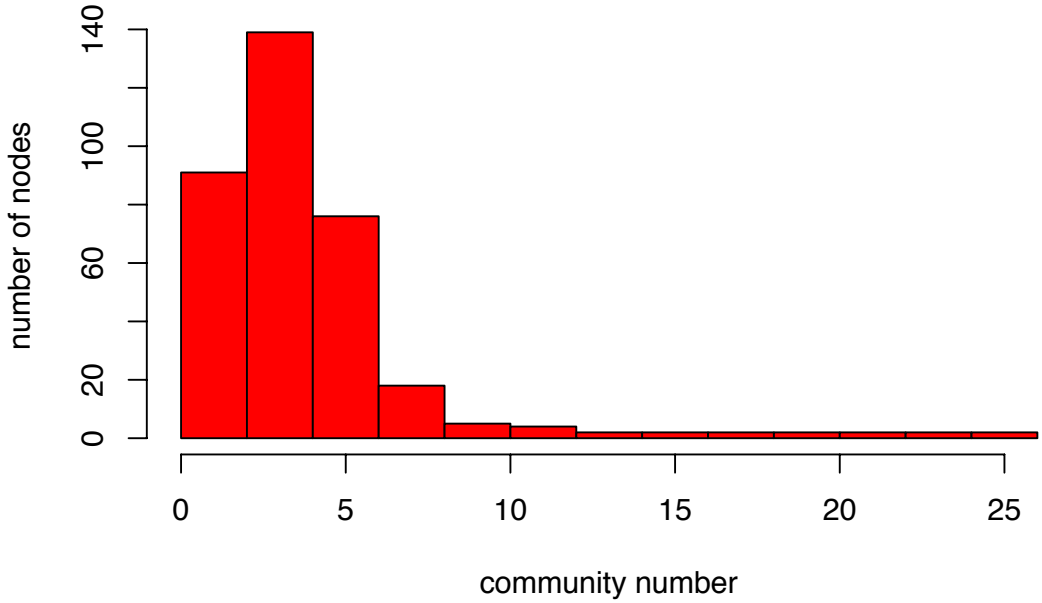
The structure after removing core code is shown below



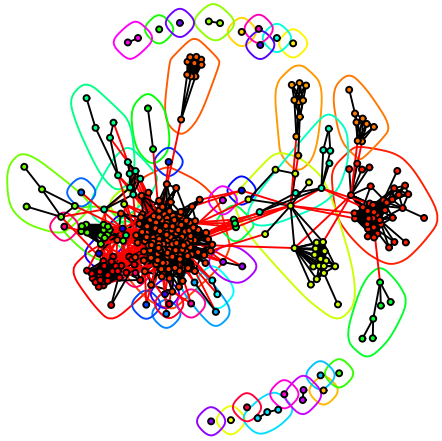
Community structure after removing core node 1 – Fast–Greedy



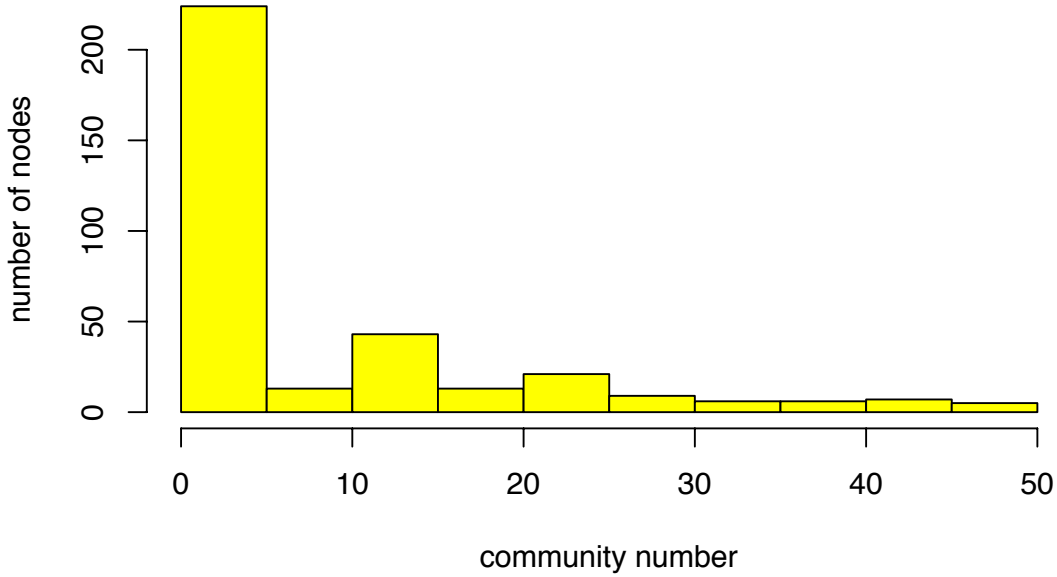
Community distribution of Fast–Greedy Algorithm



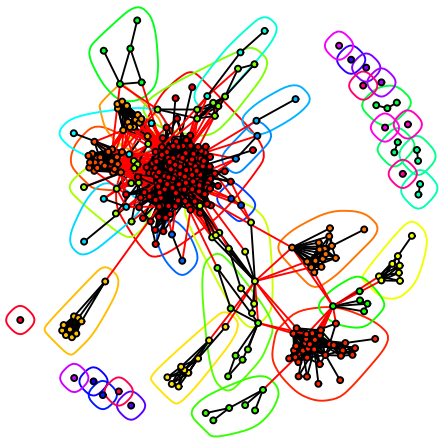
Community structure after removing core node 1 – Edge–Betweenn



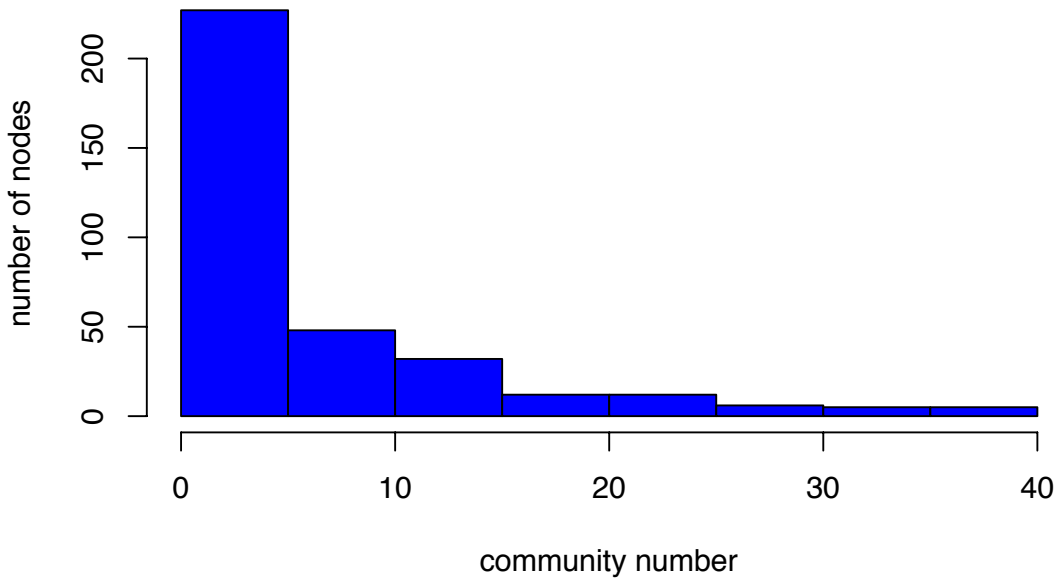
Community distribution of Edge–Betweenness Algorithm



Community structure after removing core node 1 – Infomap commu



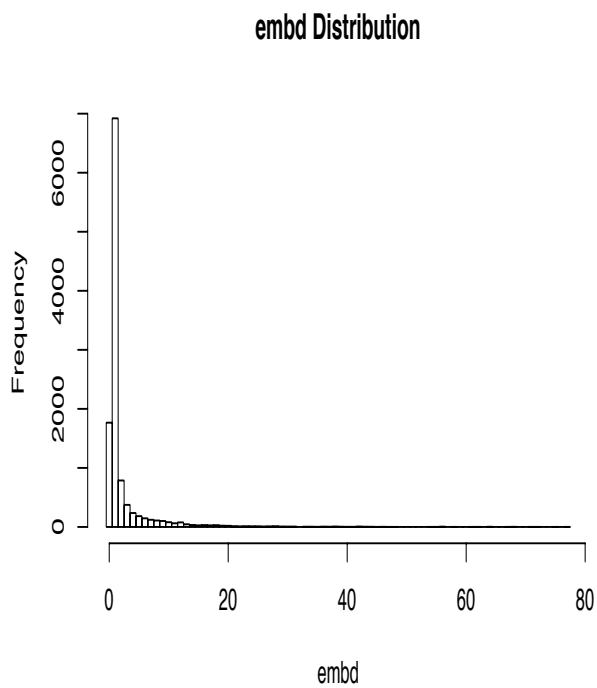
Community distribution of Infomap Algorithm

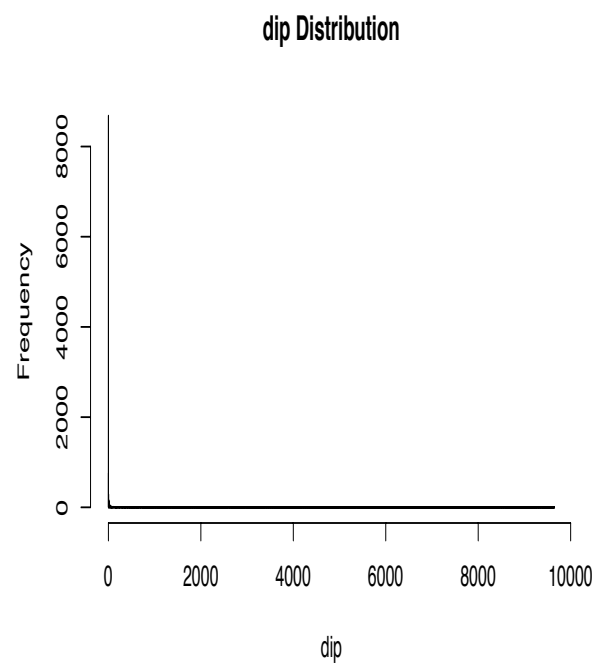


Question 5: Find dispersion and *embeddedness* for all nodes in the personal network. *Embeddedness* is the number of mutual friends a node shares with the core node and *dispersion* is the sum of distances between every pair of the mutual friends a node shares with core node (These distances should be calculated in a modified graph where the said node and the core node are removed. 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. Read details in the paper referred. Plot the distribution of *embeddedness* and *dispersion* over all personal networks created with core nodes from part 3. Plot 3 personal networks showing their community structure with colors and highlight the node with maximum dispersion in each network. Highlight the edges incident to this node as well. On each network, do the same thing for the node with maximum *embeddedness* and the node with maximum $\frac{embeddedness}{dispersion}$. Can you show and explain what characteristic of a node is revealed by each of these measures?

Solution:

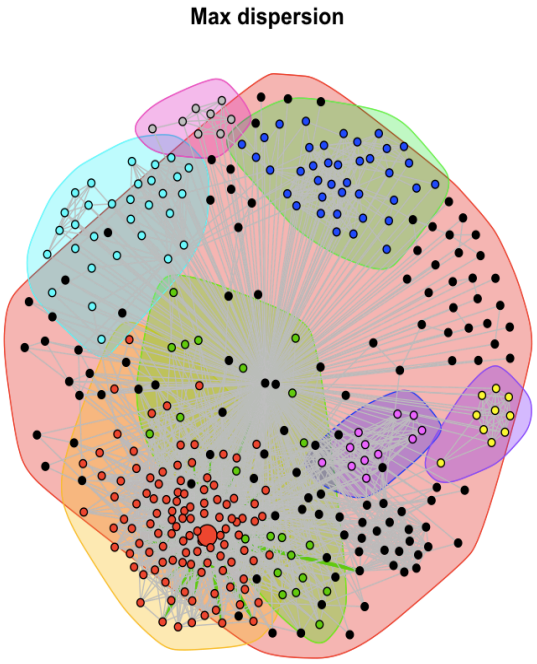
The distribution of *embeddedness* and *dispersion* over all network are show below.

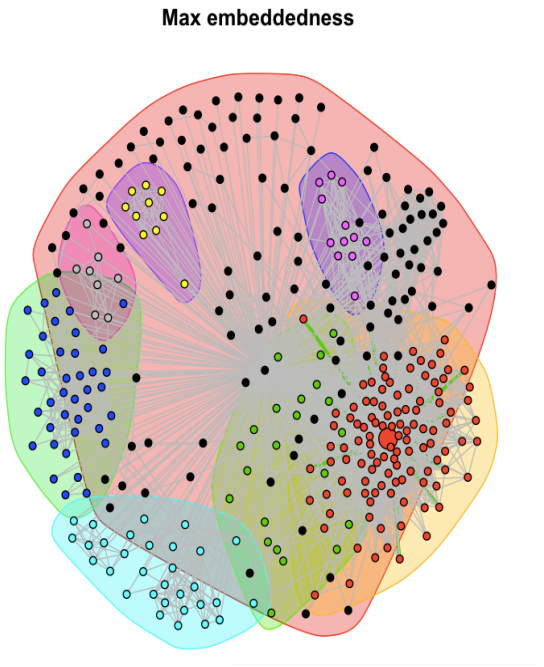


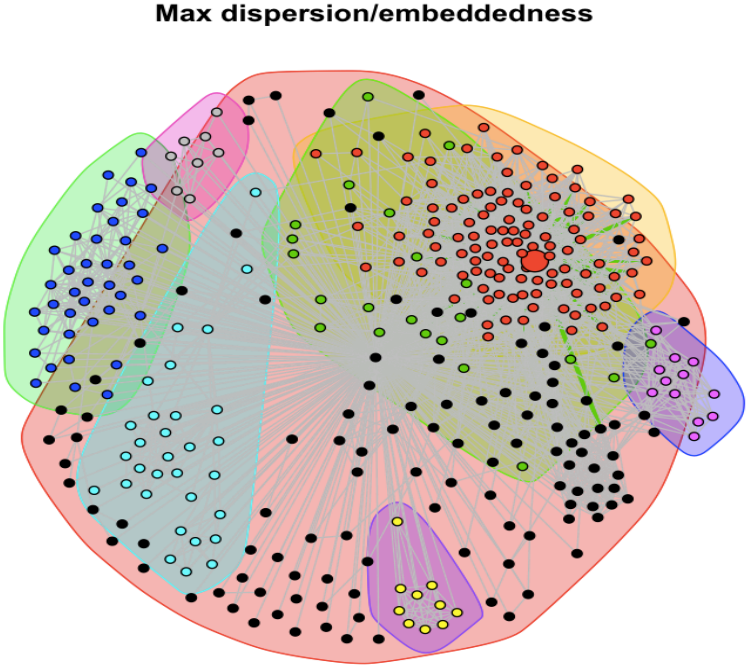


We select three personal networks showing the structure. And highlight the maximum *embeddedness, dispersion* and *dispersion/ embeddedness*.

Personal network of core node 1

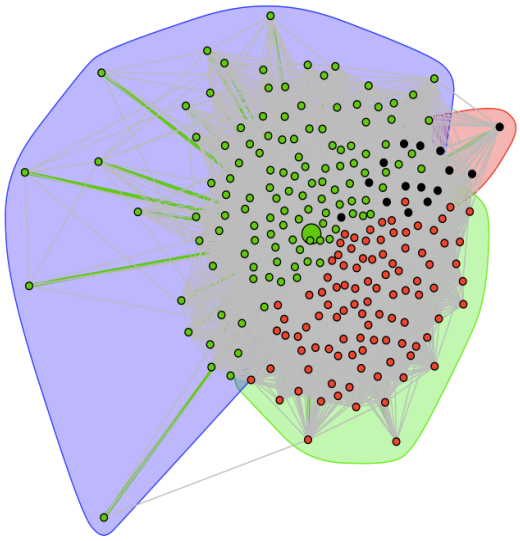




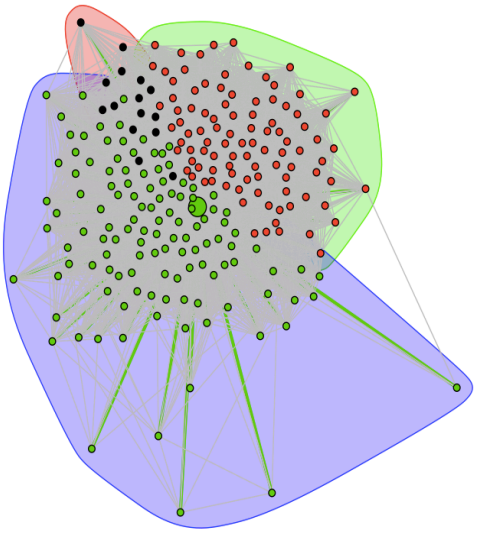


Personal network of core node 9

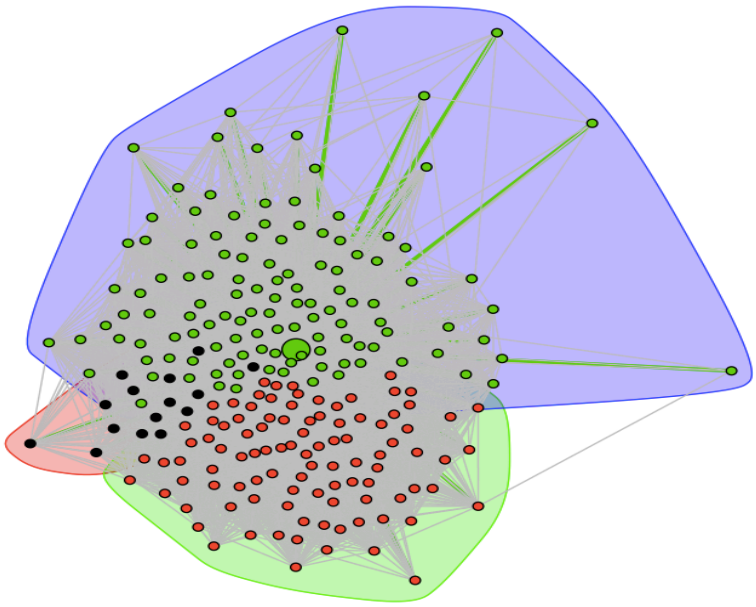
Max dispersion



Max embeddedness

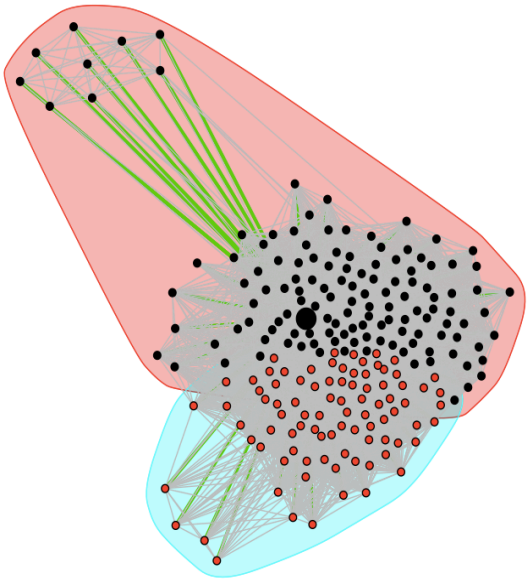


Max dispersion/embeddedness

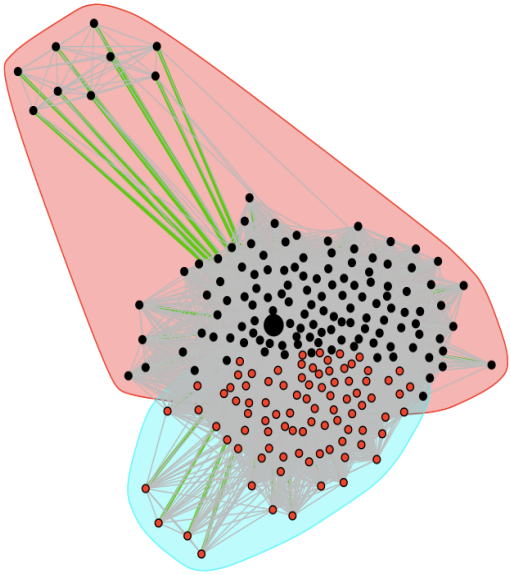


Personal network of core node 10

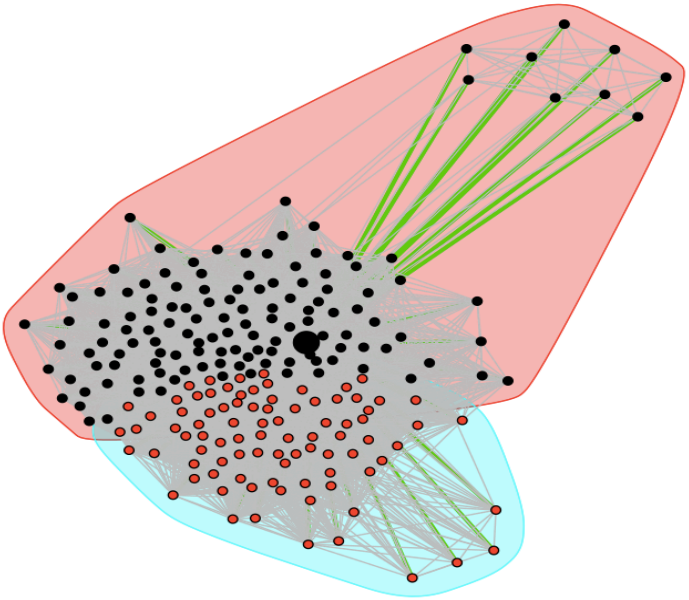
Max dispersion



Max embeddedness



Max dispersion/embeddedness



Question 6

Illustration:

In this problem, we need to find out some features that help to filter out which communities is which kind. These communities could be, for an example, “college friends”, “family members”. Even if we don’t know exactly what communities they are, we are able to find out the shared features of one community. With these features, we are able to classify the communities. For instance, we are able to decide which community is the closest among all the communities and which community is the least closest. The closer the community, the better members in the community know each other. These kinds of community could be, for an example, family or close friends. Another example of classification could decide on the size of the community. For an example, the community of small size is very likely to be good friends or family while the communities with large size could be college friends or colleagues.

The following chart shows the significant parameters we extracted from the communities of several nodes (which we call them core nodes), such as modularity index, clustering coefficient, community density, community size, size of the largest clique, average degree, maximum eccentricity.

Core node #1	modularity index	clustering coefficient	density	community size	size of largest cluster	max eccentricity	average degree
comm 1	0.2139307	0.3165125	0.03415619	114	14	2	7.719298
comm 2	0.2308127	0.4757197	0.110843	112	15	3	24.60714
comm 3	0.1774973	0.5749129	0.1147186	22	5	7	4.818182
comm 4	0.2171191	0.587013	0.1248313	39	9	4	9.487179
comm 5	0.2104921	0.6122449	0.09677419	31	7	6	5.806452
comm 6	0.244219	0.8713693	0.4242424	12	9	2	9.333333
comm 7	0.2612735	0.9354839	0.4333333	10	9	2	7.8
comm 8	0.2147126	0.7808219	0.3214286	8	5	3	4.5

Core node #108	modularity index	clustering coefficient	density	community size	size of largest cluster	max eccentricity	average degree
comm 1	0.2141424	0.4854857	0.0801286	463	24	6	37.01944
comm 2	0.2322883	0.5079454	0.1501626	482	38	2	72.22822
comm 3	0.2139188	0.8118557	0.2527163	71	22	8	17.69014
comm 4	0.2715318	0.9434389	0.9454545	11	9	2	9.454545
comm 5	0.1513584	0.375	0.5	5	3	2	2
comm 6	0.1205064	1	1	3	3	1	2
comm 7	0.1791547	0.6	0.5333333	6	3	3	2.666667
comm 8	0.2779341	NaN	1	2	2	1	1
comm 9	0.1705192	1	1	3	3	1	2

Core node #349	modularity index	clustering coefficient	density	communit y size	size of largest cluster	max eccentricit y	average degree
comm 1	0.1369819	0.4403209	0.1363794	108	12	2	14.59259
comm 2	0.1204852	0.5724374	0.3489435	103	16	5	35.59223
comm 3	0.1059969	0.4285714	0.5	5	3	3	2
comm 4	0.1536945	1	1	4	4	1	3
comm 5	0.0235753	0.8212291	0.6888889	10	7	3	6.2

Core node #384	modularity index	clustering coefficient	density	communit y size	size of largest cluster	max eccentricit y	average degree
comm 1	0.2472777	0.3967163	0.09475066	381	27	2	36.00525
comm 2	0.158787	0.470085	0.08093842	155	14	7	12.46452
comm 3	0.2326706	0.5111417	0.2202557	226	23	5	49.55752
comm 4	0.2835523	0.8698594	0.7953216	19	13	2	14.31579
comm 5	0.4465094	1	1	6	6	1	5
comm 6	0.2075474	0.9230769	0.9333333	6	5	2	4.666667

Above are the features extracted from 4 core nodes, namely core node #1, #108, #349 and # 384. The features of the other core nodes can be found in the output file generated by our R code: Q6_output_features.txt.
Based on the above examples, we are able to draw some conclusions. As we have discovered from the previous analysis, each core node belongs to multiple communities. Each of these communities are different from each other in at least one of the features.
For an example, for a specific core node, the communities with large **community sizes** are very probable to be a large community belongs to an organization such as college or a big company. On the other hand, the communities with small community sizes could be small teams or groups such as close friends or family. In addition, we are also able to see that the feature "**size of largest cluster**" shows almost the same thing as the community size, meaning the larger the community size, the larger the size of the largest cluster. Another feature that draws our attention is the **density** of the community. Usually the density grows with the decrease of the community size, but it's not always the case. It is reasonable to assume that the one community with the highest density is the closest community of the person. For an example, family or close friends. From the perspective of **average degree** of the subgraph, we can see that the larger community tends to have a higher average degree, but it's not always the case. For some nodes, a relatively smaller community has the highest average degree, which means it has the most adjacent edges. We are able to conclude that in these kind of community, people know most of the other persons in that community, even though it is a very big community. It is reasonable to assume that this kind of community could be a workplace where most people have worked there for a long time so they know most of their colleagues.
For this problem, in order to classify all the communities into two types, we use the feature "**community size"** to distinguish small group such as close friends and family from big community such as colleagues and alumni.
Some of the classification result is shown in the following charts:

Core Node #1

community number	1	2	3	4	5	6	7	8
community size	114	112	22	39	31	12	10	8
type	1	1	2	2	2	2	2	2

Core Node #108

community number	1	2	3	4	5	6	7	8	9
community size	463	482	71	11	5	3	6	2	3
type	1	1	2	2	2	2	2	2	2

Core Node #349

community number	1	2	3	4	5
community size	108	103	5	4	10
type	1	1	2	2	2

Core Node #384

community number	1	2	3	4	5	6
community size	381	155	266	19	6	6
type	1	1	1	2	2	2

The other results of type classification are in the output file: Q6_output_features.txt.

By using the community size feature to distinguish the “small but close communities” from the “large but sparse communities”, we are able to separate the neighbors of one nodes into two types: close related persons and acquaintances. the following two graphs show the two categories of people of node 1.

Question 7

In this problem, we first observed the given data set. The gplus data set consists of 6 kinds of files: 1)circles 2)edges 3)egofeat 4)feat 5)featnames 6)followers. In this problem, our mission is to find out the overlap between communities and circles tagged by the user. Therefore, we will only utilize the two types of files out of the 6: the circles file and the edges file. The circles file consists of the information of circle of the user, while the edge file provides the information of communities. We will treat the edge as a graph to extract the communities of the node just as the previous 6 problems we solved.

For each ego node, if it has more than 2 circles, we constructed the graph of communities of it with the edge list file by using read.graph() function from igraph. With the two entities, communities and circles of one node, we need to calculate the overlap percentage by comparing how many memberships in one community is also in the circles. This comparison is done by doing a two level looping, and utilizing the intersect function in R. The percentage means those membership overlapped over the total node umber of each community.

The communities of ego nodes are extracted using two algorithms: walktrap and info-map. We have been exposed to these two algorithms in the previous homework. For better understanding, I will highlight their difference here:

Walktrap.community():

This function tries to find densely connected subgraphs, also called communities in a graph via random walks. The idea is that short random walks tend to stay in the same community.

Usage: walktrap.community(graph, weights = E(graph)$weight, steps = 4, merges =TRUE, modularity = TRUE, membership = TRUE)

infomap.community():

This function finds community structure that minimizes the expected description length of a random walker trajectory.

Usage: infomap.community (graph, e.weights = NULL, v.weights = NULL, nb.trials = 10, modularity = TRUE)

Following are two figures showing the community structure of both the Walktrap and Info-map community for one node (node index: 100962871525684315897).

As we can see, by using WalkTrap algorithm, the structure is divided into 3 communities, dented by three different colors: blue, yellow, orange. Obviously, community Orange consists of the most population, while the blue community is the smallest. On the other hand, the Info-map community divides the community into two parts, where the Orange community has the most people and the blue community only consists of a very small portion of the people.

The overlapping table of some selected nodes for the two algorithms are shown as following.

node: 100535338638690515335			Walktrap.community()	
	Circle #1	Circle #2	Circle #3	Circle #4
community #1	0	0.01785714	0.05357143	0.03571429
community #2	0.01433121	**0.3550955**	0.6178344	0.08917197
community #3	0.1105398	0.01542416	0.02570694	**0.3701799**
community #4	0	0.1315789	**0.8070175**	0.01754386
community #5	0	0	0	0

node: 100535338638690515335			Infomap.community()	
	Circle #1	Circle #2	Circle #3	Circle #4
community #1	0.044750430292599	0.210843373493976	0.423407917383821	0.172977624784854
community #2	0	0	0	0
community #3	0	0	0	0
community #4	0	0	0	0
community #5	0	0	0.5	0
community #6	0	0	0	1
community #7	0	0	0	0

We can draw some conclusion from the results. For this user, each of his/her circle covers multiple communities.
But there is always one community that occupies the largest part, as the number in **bold**. Thus we speculate this person may prefer to gather a large group of people together from different communities. But his/her circles tagged mainly based on the communities. For an example, he/ she might have circles like “family”, “college friends”,”colleagues” etc,.

node: 100962871525684315897			Walktrap.community()
	Circle #1	Circle #2	Circle #3
community #1	0.055793991416309	0.0729613733905579	0.0515021459227468
community #2	0	0	0
community #3	0	0	0
community #4	0.705329153605016	0.313479623824451	0.112852664576803

node: 100962871525684315897			Infomap.community()
	Circle #1	Circle #2	Circle #3
community #1	0.426523297491039	0.209677419354839	0.0860215053763441
community #2	0	0	0

For the second user, there are only three circles and each of them mainly contains two communities(for walk trap algorithm). Also, the person also evenly put his/her communities into 3 circles. Therefore,we speculate that this person has a social network mainly within two large group and he is inclined to categorize the same group of people into different categories.

node: 101263615503715477581			Walktrap.community()	
	Circle #1	Circle #2	Circle #3	Circle #4
community #1	0	0	0	0
community #2	**0.0757749712973594**	**0.0103329506314581**	**0.0625717566016074**	**0.0774971297359357**
community #3	0.0108108108108108	0	0.0270270270270027	0.00540540540540541
community #4	0	0	0.0043859649122807	0
community #5	0	0	0	0
community #6	0	0	0	0
community #7	0	0	0	0
community #8	0	0	0	0
community #9	0	0	0	0
community #10	0	0	0	0
community #11	0	0	0	0

node: 101263615503715477581			**Walktrap.community()**	
	Circle #1	Circle #2	Circle #3	Circle #4
community #1	0.0545602605863192	0.007328990228013030	0.04682410423452770	0.0553745928338762

For the third user, it is clearer that each circle almost contains only one community(the other communities consist only a small portion which sometimes can be ignored), as the numbers in **bold**.So we speculate that this person is more likely to categorize his social network into smaller group with specific tag or circles.