EE232E Graphs and Network Flows

Project 1 Report

Team Members:
Yutong Zhang, 204538859
Zihao Zhang, 004593253
Junchi Zhang, 804593126

2016/5/16

1. Introduction

We studied the social networks and graphs of users' personal friendship on facebook and Google+, explored the community structures in the friendship network and their interpretation and applications. Through this project, we got more familiar with igraph, learned some techniques to extract graph information like community structures, dispersion and embeddedness.

2. Problems

2.1 Problem1

We downloaded the Facebook graph edgelist file "facebook_combined.txt" and found that the graph is connected. The diameter of the graph is 8.

In order to fit the curve, we try many different models like:

```
nls(y ~ I((1/x*a) + b*x), data = df, start = list(a = 1, b = 0),trace=T),
nls(y ~ I((a + b*log(x))), data=df, start = list(a = 0, b = 0),trace=T),
nls(y ~ I((exp(a + b * x))), data=df, start = list(a=0,b=0),trace=T),
nls(y ~ I((1/x*a)+b), data=df, start = list(a=1,b=0),trace=T))
```

And the fitting are shown as below:(using ggplot)

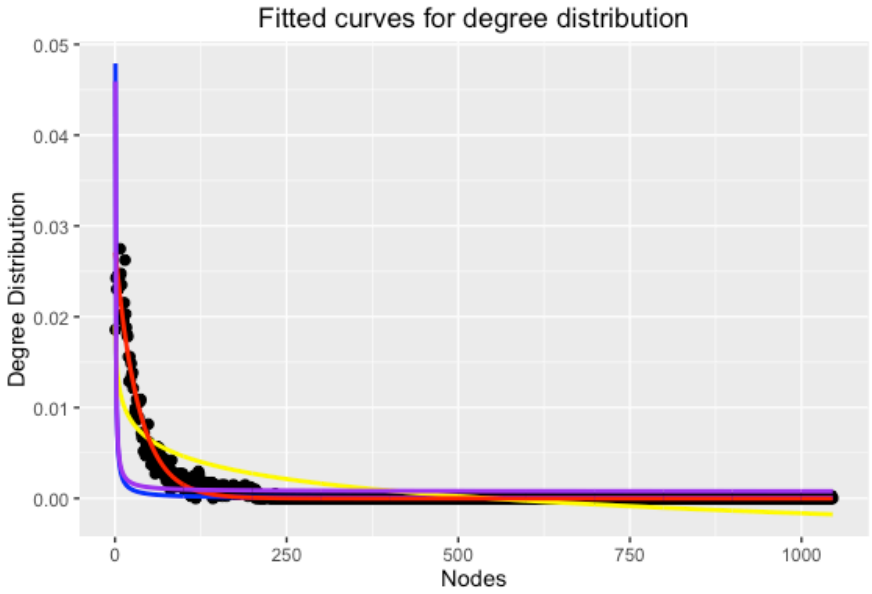


Figure 2.1.1Different fitted curves using different models

Finally we choose the function model = nls(y ~I(exp(1)^(a+b*x)) to fit the model, and we got the result of a is -3.59, b is -0.02. The curve's total mean squared error is 1.446e-06, and the average degree is 43.691.
We plot the degree distribution and fitted curve as below:

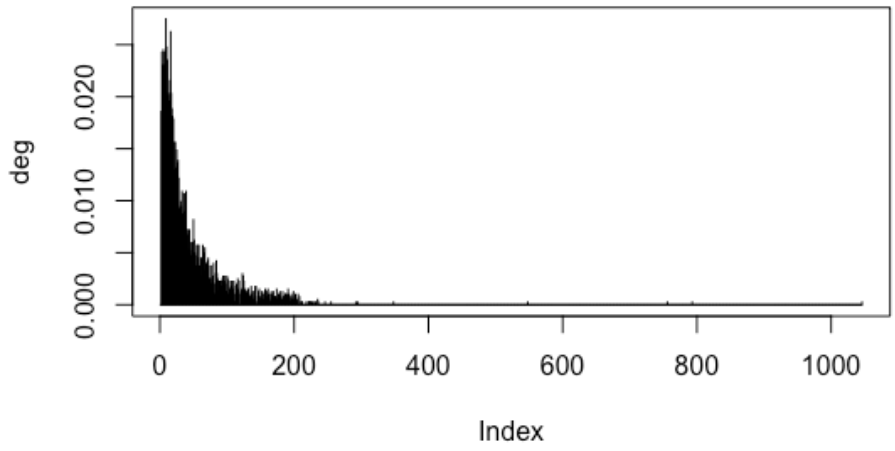


Figure 2.1.1 The degree distribution of the graph

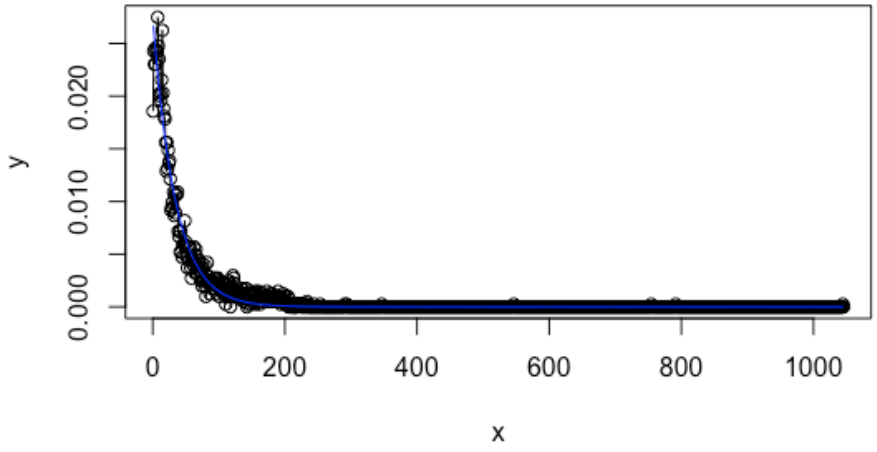


Figure 2.1.3 The fitted curve

2.2 Problem2

In this problem, we studied the personal network of node 1. Personal network of node 1 indicates a graph that consists of node 1 and its neighbors and the edges that have both ends within this set of nodes. After calculation, there are 348 nodes and 2866 edges in total in the graph. The personal network of node 1 is shown as follows:

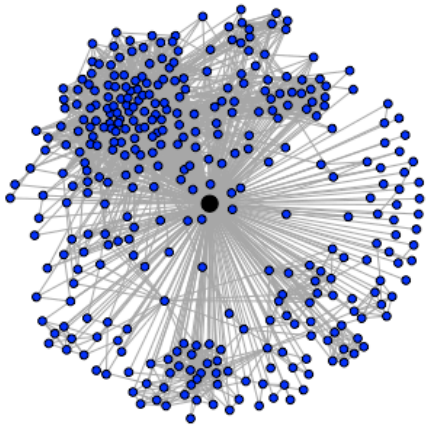


Figure 2.2.1 Personal network of node 1

The black point is node 1 and blue points are other nodes. We can easily see from the figure that all nodes except node 1 share node 1 as their common friend. That is the personal network of node 1.

2.3 problem 3

By calculating the number of neighbors for every node in the graph, we find there are **40** nodes in the graph that have more than 200 neighbors. We name these nodes as ‘coreNodes’. The index of this nodes in the Graph are as follows:

1	108	349	352	353	**367**	573	686	839	918
1071	1076	1150	1216	1232	1254	1286	1313	1374	1491
1822	2127	2131	2146	2155	2996	3003	3059	3097	3107
3147	3158	3166	3170	3170	3297	3339	3371	3408	3446

Table 2.3.1 Node index in the original graph with more than 200 neighbors

The average degree is **279.375**

We choose coreNode 367 (highlighted in table 2.3.1) as an example to explore the community structure using Fast-Greedy, Edge-Betweenness, and Infomap community detection algorithms in ‘igraph’. The structure of the subgraph is shown as below. The coreNode of this subgraph is marked with a larger size than the other nodes.

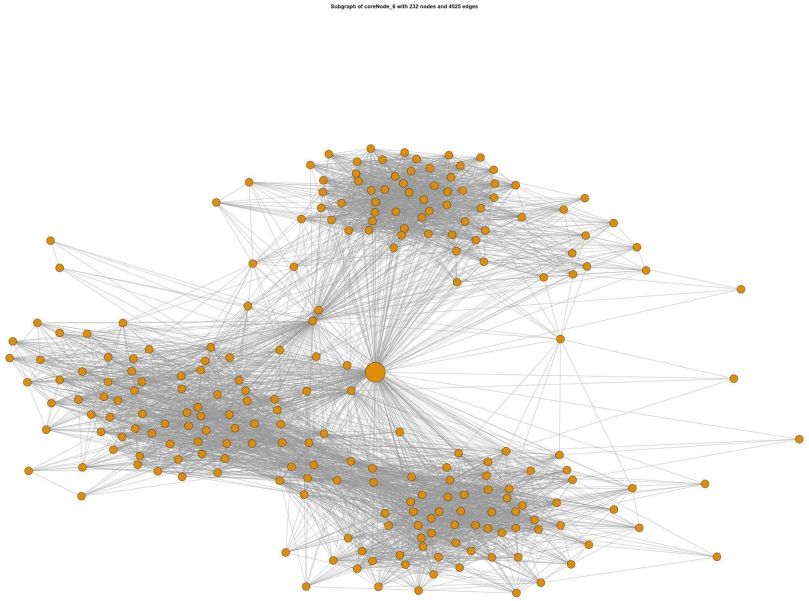


Figure 2.3.1 Subgraph of coreNode_6 with 232 nodes and 4525 edges

This subgraph has 232 nodes and 4525 edges in total。

1) By using **Fast-Greedy** algorithm to detect the community structure, we got

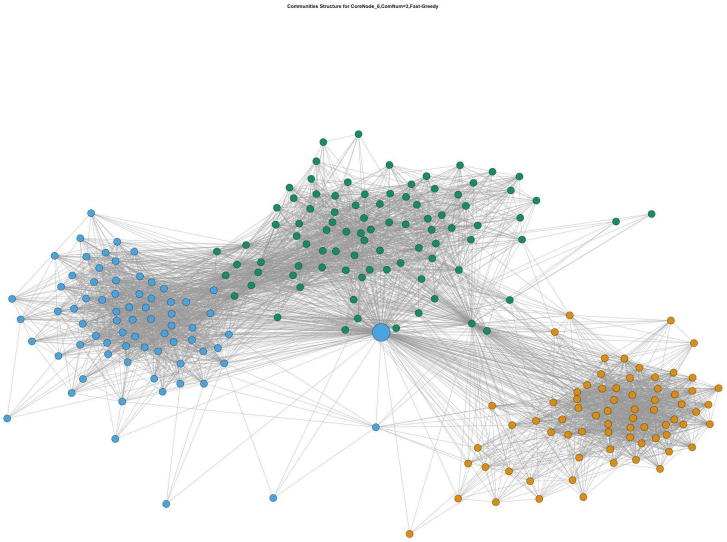


Figure 2.3.2 Communities Structure for CoreNode_6, ComNum=3, Fast-Greedy

There are **3** communities which are marked with different colors in blur, green and khaki. In order to distinguish the communities clearer, we add different background to the original figure, and the relation of subordination is as follows,

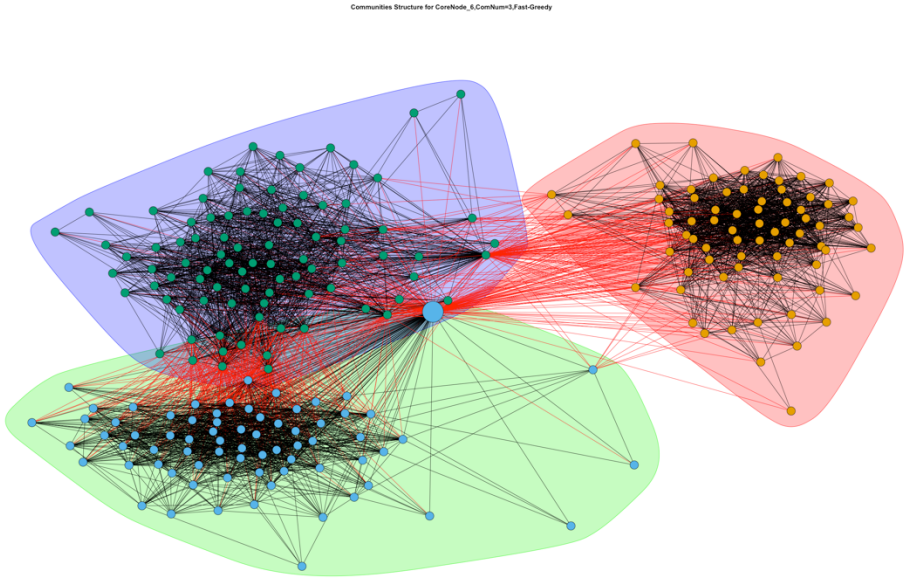


Figure 2.3.3 Communities Structure for CoreNode_6, ComNum=3, Fast-Greedy

2) By using **Edge-Betweenness** algorithm to detect the community structure,

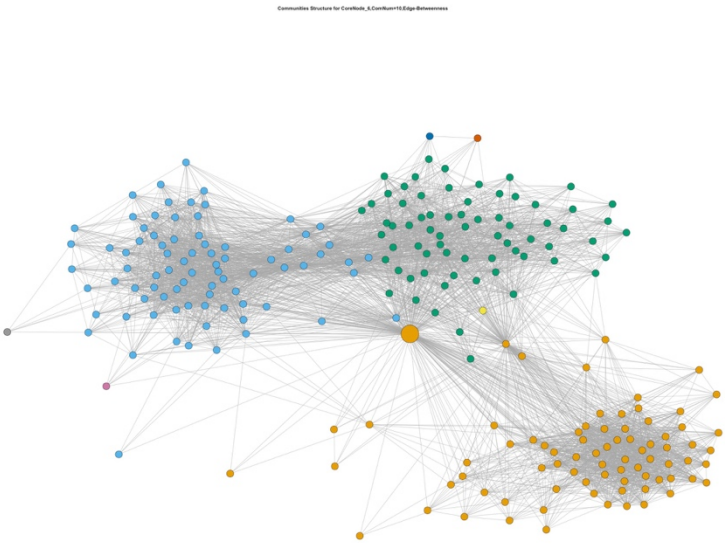


Figure 2.3.4 Communities Structure for CoreNode_6, ComNum=10, Edge-Betweenness

There are **10** communities which are marked with different colors. In order to distinguish the communities clearer, we add different background to the original figure, and the relation of subordination is as follows,

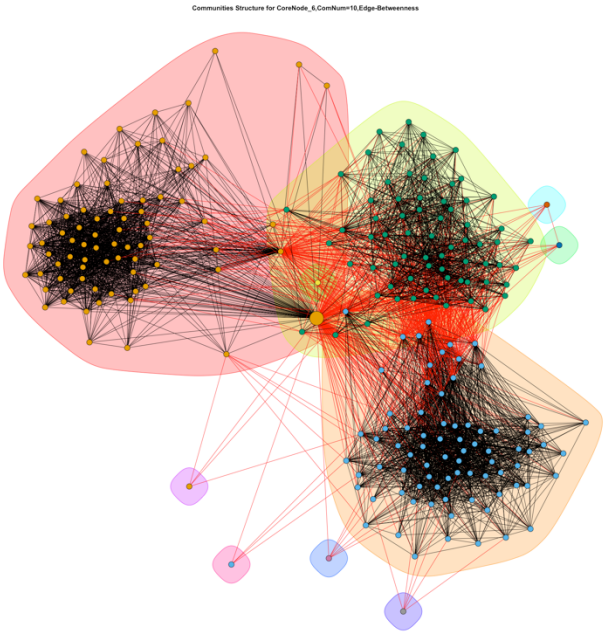


Figure 2.3.5 Communities Structure for CoreNode_6, ComNum=10, Edge-Betweenness

3) By using **Infomap** algorithm to detect the community structure, we got

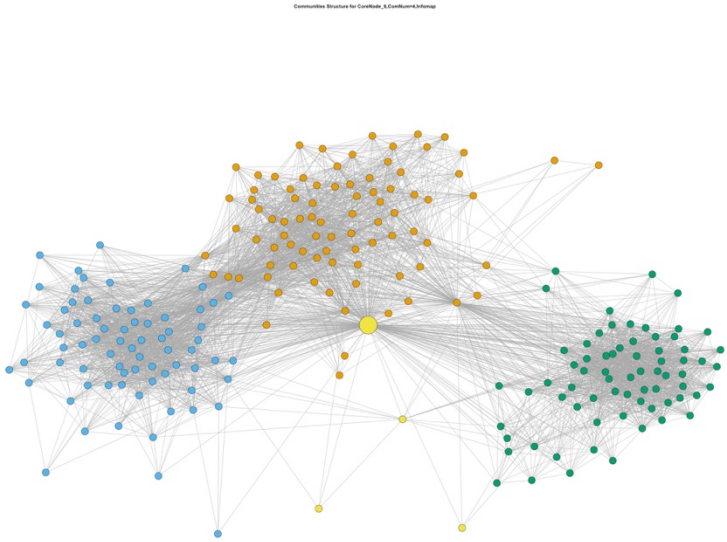


Figure 2.3.6 Communities Structure for CoreNode_6, ComNum=4, Infomap

There are **4** communities which are marked with different colors. In order to distinguish the communities clearer, we add different background to the original figure, and the relation of subordination is as follows,

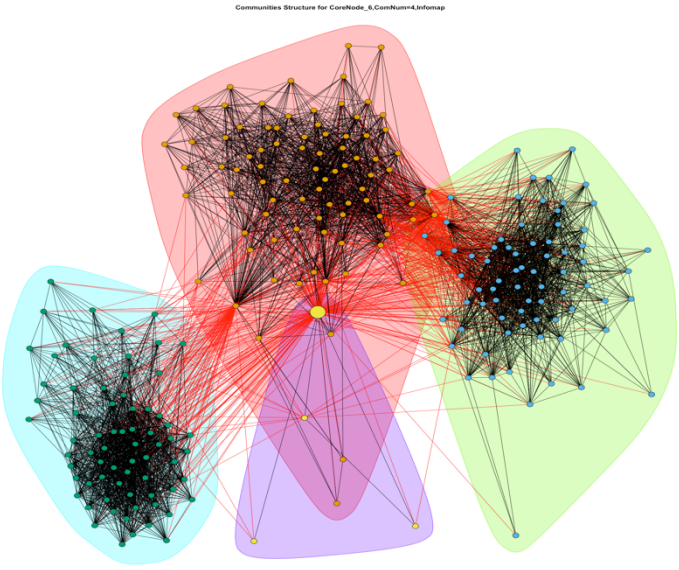


Figure 2.3.7 Communities Structure for CoreNode_6, ComNum=4, Infomap

Compared with the figures we get from three different algorithms, we find that:

① Edge-betweenness algorithm has a higher sensitivity of dividing the graph into more different communities than the others. For the personal network we picked above, number of communities is 10 from edge-betweenness detection, 4 for infomap, and 3 for fast-greedy.

(In our another try of different coreNode, NodeID=349,with the same community detection algorithms, we could get **5** communities using fast-greedy algorithm, **9** communities using infomap algorithm and **104** communities using edge-betweenness algorithm. This is much more obvious to see ① in this case.)

② There are three major communities in this personal network.

③ All of these three algorithms are single community decision, which mean it will not put a node into more than one community. Thus in some part of the personal network, background overlapping occurs.

2.4 Problem 4

Remove the coreNode for its personal network, and running the above community detection algorithms, we get,

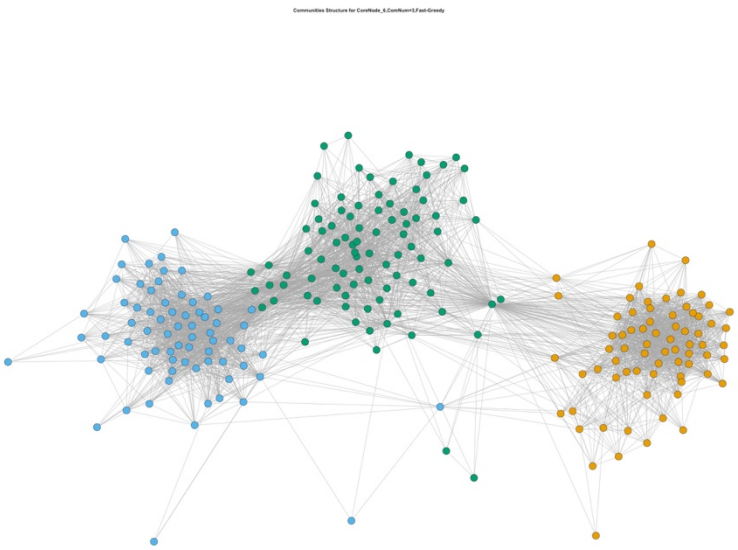


Figure 2.4.1 Communities Structure for CoreNode_6(coreNode removed), ComNum=3, Fast-Greedy

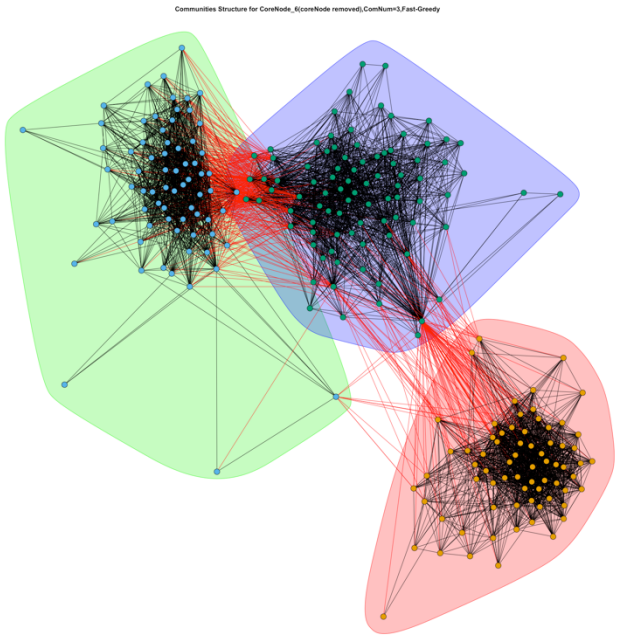


Figure 2.4.2 Communities Structure for CoreNode_6(coreNode removed), ComNum=3, Fast-Greedy

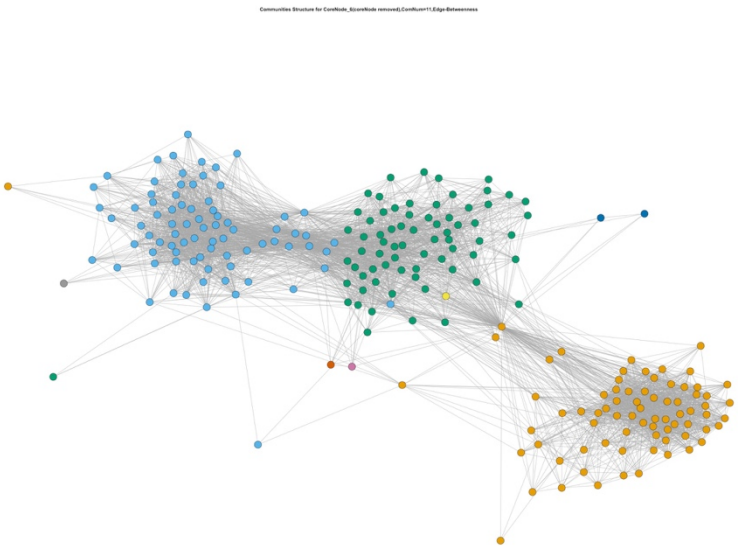


Figure 2.4.3 Communities Structure for CoreNode_6(coreNode removed), ComNum=11, Edge-Betweenness

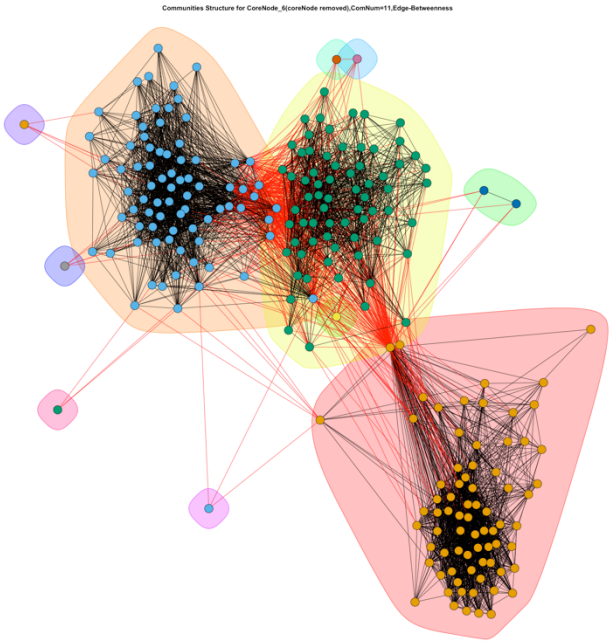


Figure 2.4.4 Communities Structure for CoreNode_6(coreNode removed), ComNum=11, Edge-Betweenness

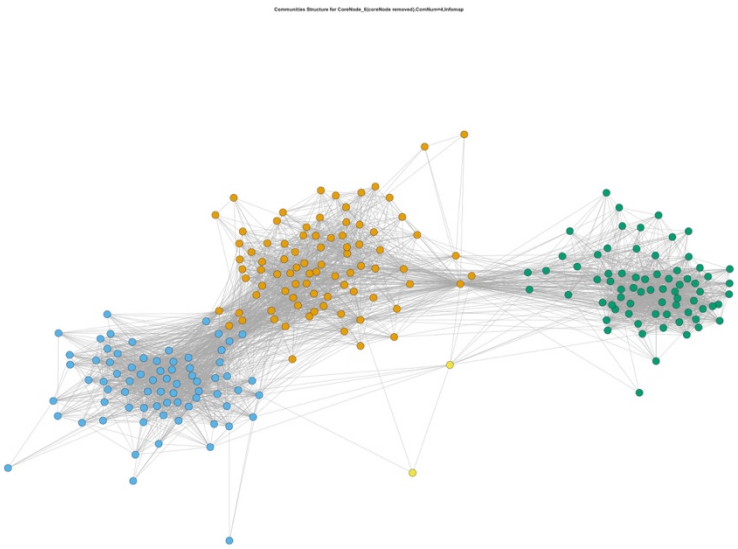


Figure 2.4.5 Communities Structure for CoreNode_6(coreNode removed),ComNum=4,Infomap

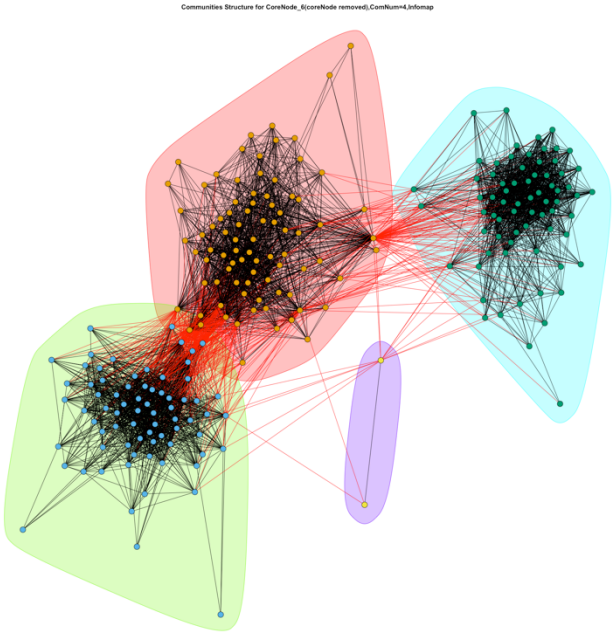


Figure 2.4.6 Communities Structure for CoreNode_6(coreNode removed),ComNum=4,Infomap

After removing the coreNode from the personal network, the number of communities detected has some changes as below,

Number of communities	Fast-Greedy	Edge-Betweenness	Infomap
With the coreNode	3	10	4
Wihout the coreNodes	3	11	4

Table 2.4.1 Number of communities using different detection algorithms

From the figures and table above, we can find that this personal network is relative stable.

①After removing the coreNode which is connected with all other nodes in the personal network, the rest nodes have a tend to break into more parts than the original one. So we can see that the number of communities using edge-betweenness is increase to 11 compared with the original 10. However, the other two does not increase. This means that the rest nodes of this network are connected to others with higher Possibilities thus a remove of a coreNode will not bring a very big influence of this network. By compare the modularity of this network with the one in part 3, we can find there is only 5% changes.

②The remove of the coreNode has a influence of the assignments of communities of the nodes, especially the ones with a relative small degree but not the smallest in this personal network. The membership of nodes could be reassigned to another community compared with the membership in part 3. Take figure 2.3.7 and 2.4.6 as an example. (shown below)

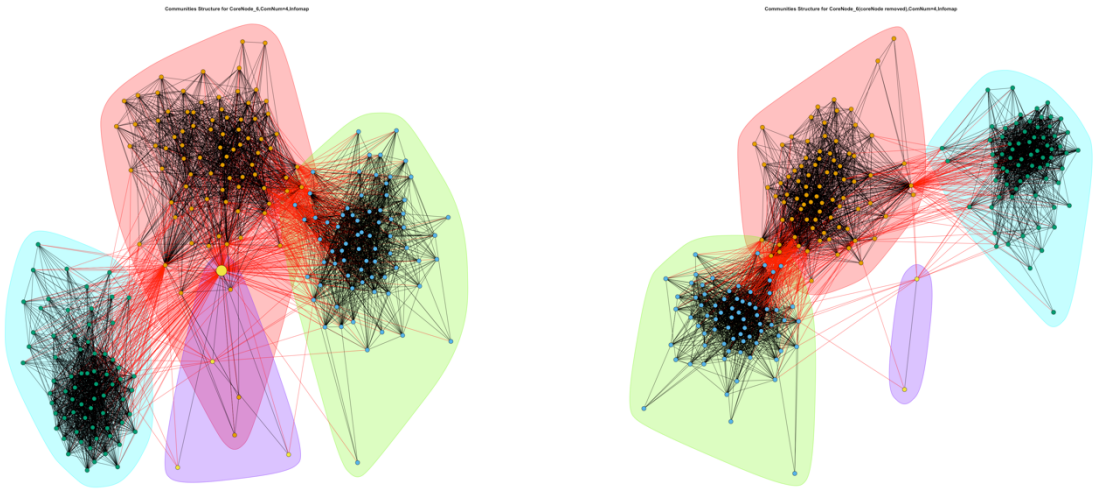


Figure 2.3.7 Communities Structure for CoreNode_6, ComNum=4, Infomap and
Figure 2.4.6 Communities Structure for CoreNode_6(coreNode removed), ComNum=4,Infomap

2.5 Problem 5

Two new features are calculated in this part: embeddedness and dispersion. Embeddedness is the is the number of mutual friends a node shares with the core node in the personal network. Thus the higher the embeddedness, the more mutual friends are shared within the two people, which means the closer the two people are. Dispersion is the sum of shortest distances between every pair of the mutual friends a node shares with core node. The higher the dispersion, the more mutual friends from different areas(communities) are shared within the two people. which means the two people are closer in many areas so that this pair of people are very good friends with more common interests.

Instead of let the distance between people be 1 if there is a path from someone to another and 0 for the other situation in the paper. We use ‘get.shortest.path’ to calculate the distance between two nodes.
And the results are as follows,

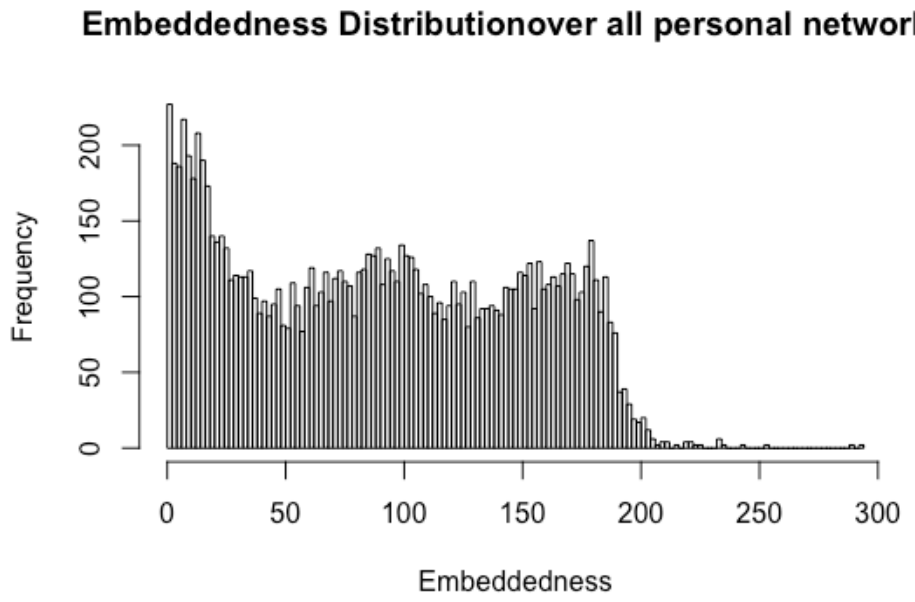


Figure 2.5.1 Embeddedness Distribution over all personal network

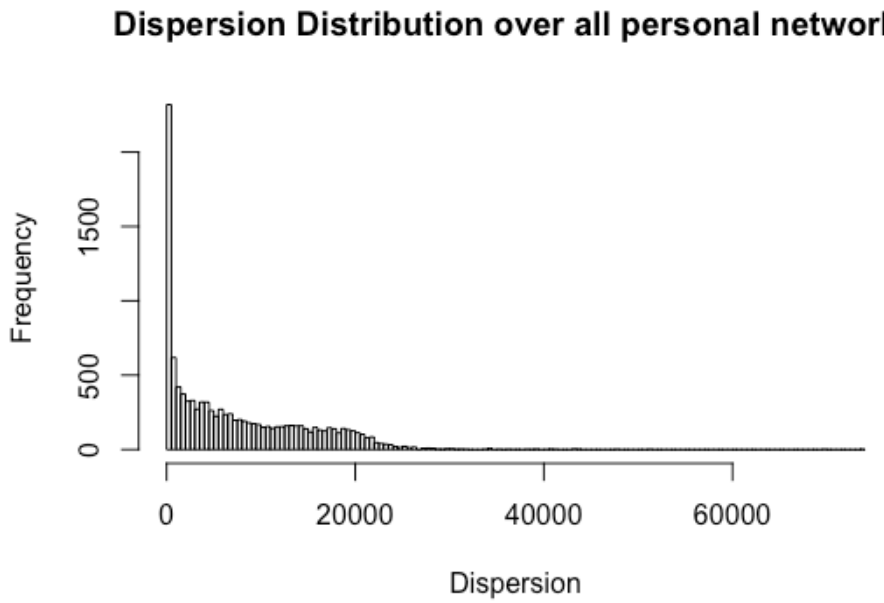


Figure 2.5.2 Dispersion Distribution over all personal network

When calculate the dispersion, the graph to use should not only be the corresponding personal networks which cut off many edges and a shortest path could be formed between a third person which is not in the personal network. Instead, we use the original graph g and remove the corresponding coreNode and target node to calculate the shortest path.

Besides the distributions over the whole network we select coreNodes 367(6^{th} in coreNodes), 573(7^{th} in coreNodes) and 3446(40^{th} in coreNodes) to plot their communitt structure.Node with max embeddedness or dispersion or the ratio of dispersion over embeddedness are highlighted.

- **Node 367(6th in the coreNodes)**

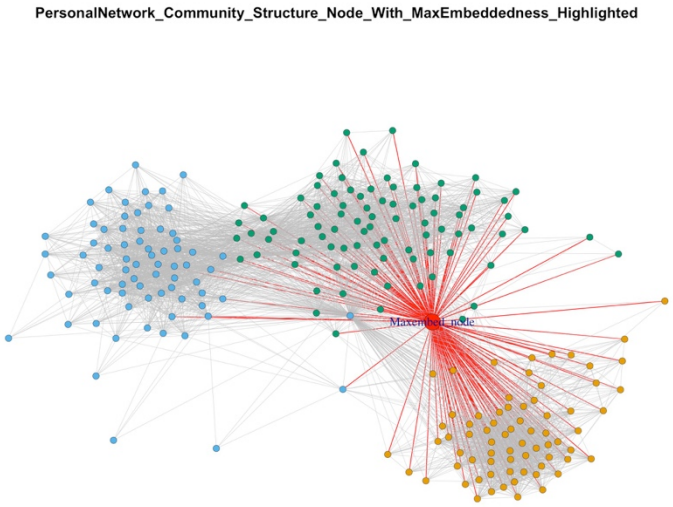


Figure 2.5.3 coreNode 6th - Personal Network Community Structure, Node With MaxEmbeddedness Highlighted

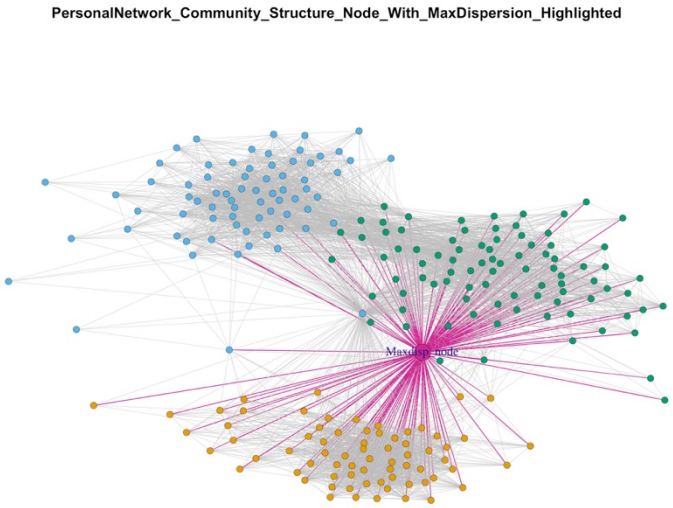


Figure 2.5.4 coreNode 6th - PersonalNetwork Community Structure, Node With MaxDispersion Highlighted

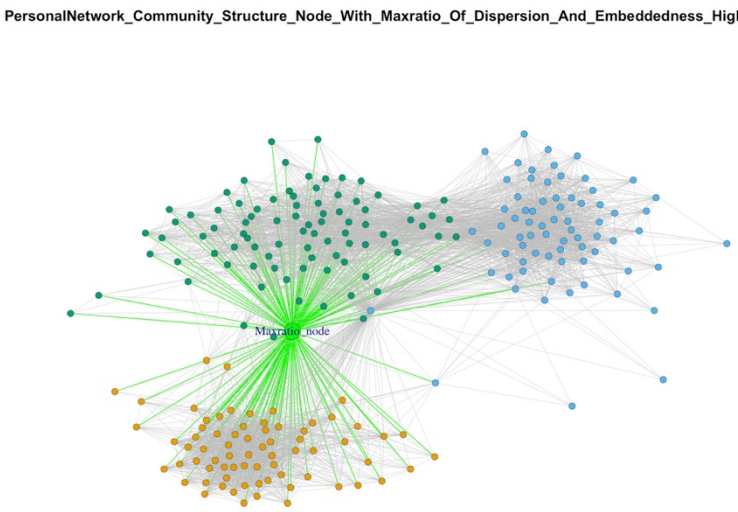


Figure 2.5.5 coreNode 6th - PersonalNetwork Community Structure Node With Maxratio_Of_Dispersion_And_Embeddedness Highlighted

- **Node 573(7th in the coreNodes)**

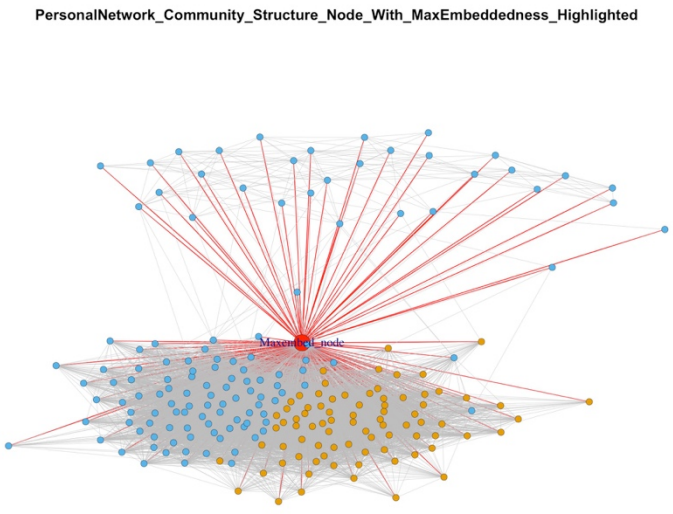


Figure 2.5.6 coreNode 7th - Personal Network Community Structure, Node With MaxEmbeddedness Highlighted

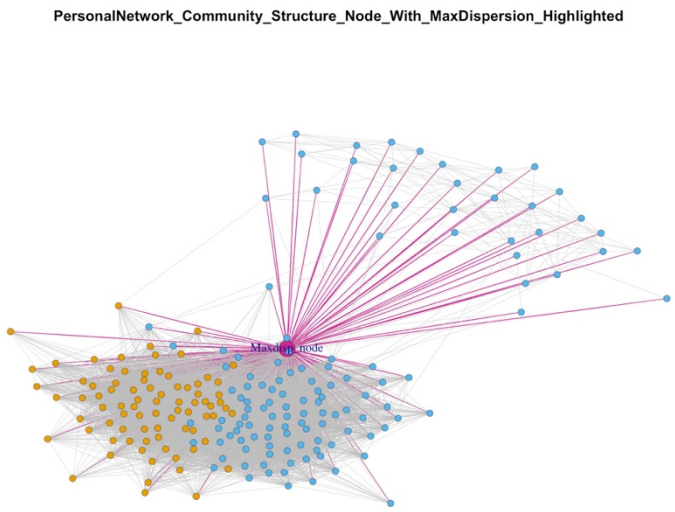


Figure 2.5.7 coreNode 7th - PersonalNetwork Community Structure, Node With MaxDispersion Highlighted

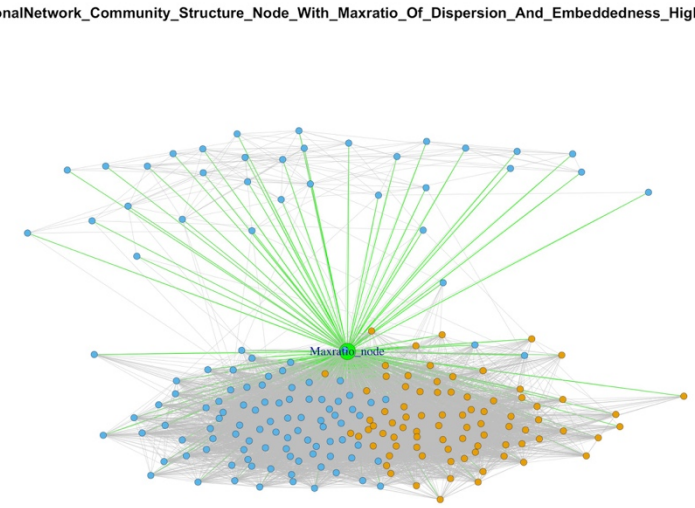


Figure 2.5.8 coreNode 7th - PersonalNetwork Community Structure Node With Maxratio_Of_Dispersion_And_Embeddedness Highlighted

- **Node 686(8th in the coreNodes)**

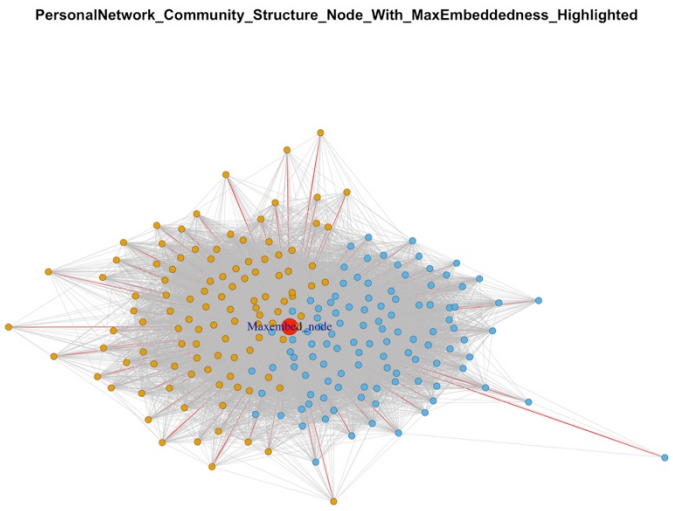


Figure 2.5.9 coreNode 8th - Personal Network Community Structure, Node With MaxEmbeddedness Highlighted

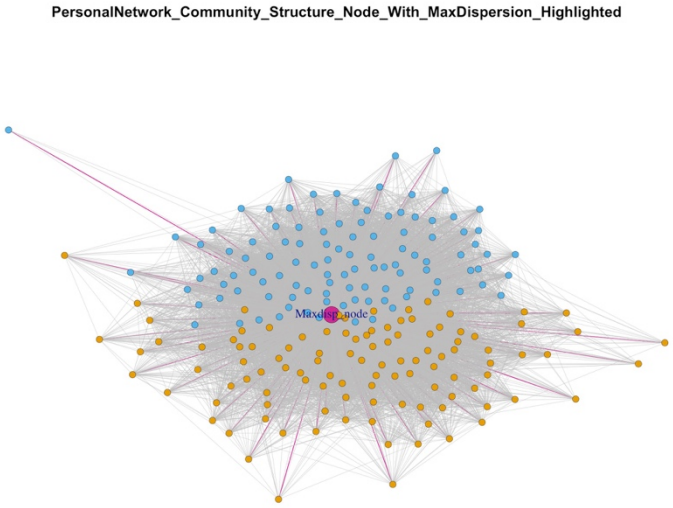


Figure 2.5.10 coreNode 8th - PersonalNetwork Community Structure, Node With MaxDispersion Highlighted

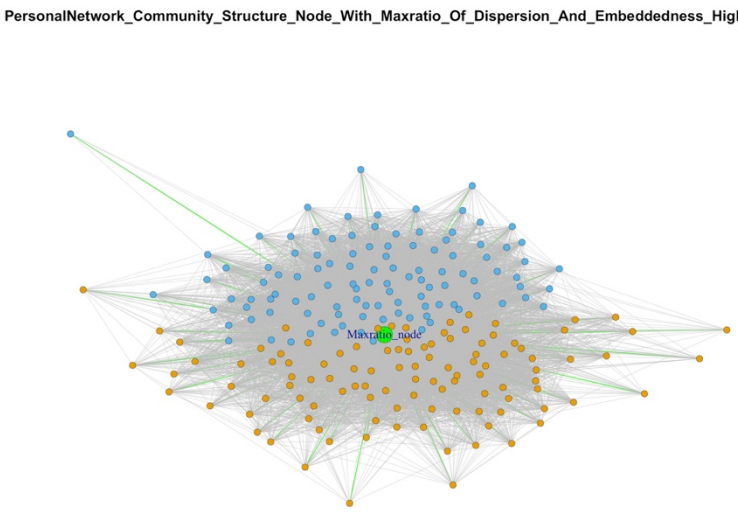


Figure 2.5.11 coreNode 8th – Personal Network Community Structure Node With Maxratio_Of_Dispersion_And_Embeddedness Highlighted

From the 9 community structure figures we can find that,

① In some personal networks, the node with max embeddedness and the node with max dispersion and/or the node with max ratio of dispersion over embeddedness are the same node. For example, in the personal networks formed by the coreNode 367(6th in the coreNodes),the three node with max values are the same one –node 108.

By further calculation we find node 108 is a node with very high degree. Node108 has 1045 neighbors out of 4039 nodes in original graph g. The hot nodes in a graph is much likely to be the max embeddedness/dispersion/ratio(dispersion/embeddedness) node.

②Personal networks with high max embeddedness value are relative stable. Combined with the results and analysis in part 4(Page 13), after removing the coreNodes, these personal networks will not break into many pieces which is not connected. The node with max embeddedness will acts as a sub-coreNode.

Part 5 conclusion

From part 5 we can summarize that:

①Embeddedness could indicate the closeness between two people. the higher the embeddedness, the more mutual friends are shared within the two people, which means the closer the two people are.But it is influenced by the structure of communities in personal network. For people in the same community, they could have many mutual friends. Compared with width in friendship, Embeddedness is more like a factor of depth.

②Dispersion: Large dispersion means your mutual friends are not likely to know each other，which is quite common in real life. Compared with depth in friendship, Embeddedness is more like a factor of width.In the other hand, The higher the dispersion, the more mutual friends from different areas(communities) are shared within the two people. which means the two people are closer in many areas so that this pair of people are very good friends with more common interests.

③Dispersion/Embeddedness: The ratio of dispersion and Embeddedness could be seen as an average dispersion for each mutual friend. It become larger when the dispersion become larger and the embeddedness become smaller. We could see that dispersion could be relative large when the embeddedness is large, even most their mutual friends knows each other. So the ratio is a kind of normalization. The combination of these two features could be more precise and reliable.

2.6. Problem6

Firstly, we generated personal networks with more than 200 neighbors. Then among all the communities of each network, we keep those communities with more than 10 nodes. After this, we choose two typical communities in each person's networks. Community type 1 has the minimum #edge/#vertex value, and type 2 has the maximum #edge/#vertex value. The community graphs of two users are illustrated correspondingly in the following figures:

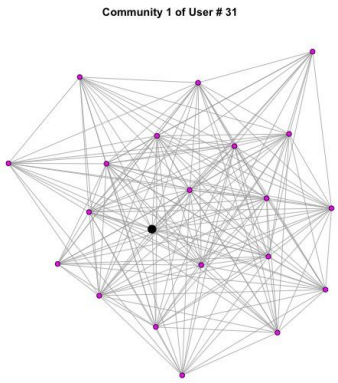
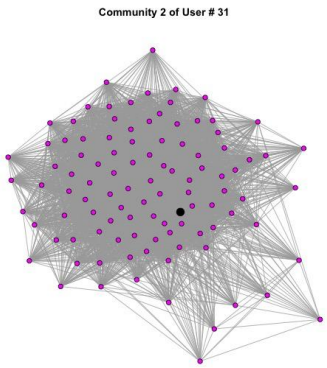
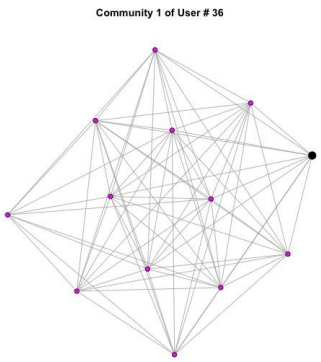
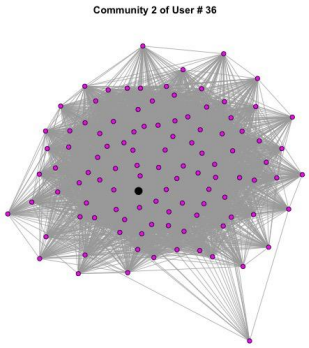
	Community type 1	Community type 2
User 1		
User 2		

Figure 6.1 Community Graphs

From the community graphs we can see that both user's community 1 graphs have fewer nodes and vertices, and community 2 graphs have closer connections between nodes. It is reasonable because community 1 may be just acquaintances and community 2 may be some close friends like college friends or high school friends.

We also calculated each community's average degree, clustering coefficient and density correspondingly. From the table below we can see that the two users share many similarities on each type of community.

	Average degree	Clustering coefficient	Density
Community 1 of user 1	0.8390	0.9007	0.8810
Community 1 of user 2	0.8639	0.9375	0.9359
Community 2 of user 1	0.7294	0.8165	0.7365
Community 2 of user 2	0.7828	0.8462	0.7903

Problem 7

In this problem, we studied a different real social network with tagged relationships, namely Google+ ego networks. The Google+ network, unlike the Facebook networks we studied in the above 6 problems, is a directed network. Firstly, we filter the networks and create personal networks for users who have more than 2 circles. There are 57 personal networks in total. Next, we can use Walktrap and Infomap methods to extract and analyze the community structure of each personal network. Because of the large number of personal networks here, we cannot put all of their results in report. We pick up three personal networks and show how communities overlap with the user's circles. We use "percentage" to indicate the overlapping. "Percentage" means the percentage of nodes that are overlapped between communities and circles. All of the output results are uploaded as well (walktrap.txt and infomap.txt).

Case Study 1: Node 100535338638690515335

Using Walktrap

	Circle.1	Circle.2	Circle.3	Circle.4
Community.1	0	0	3.45%	3.45%
Community.2	0.80%	35.58%	62.32%	8.05%
Community.3	12.17%	2.07%	2.85%	38.86%
Community.4	0	4.34%	8.69%	4.34%
Community.5	0	13.16%	80.70%	1.75%

Using Informap

	Circle.1	Circle.2	Circle.3	Circle.4
Community.1	4.47%	22.56%	45,23%	17.44%
Community.2	0	2.23%	6.81%	6.81%
Community.3	0	0	0	0
Community.4	0	0	0	0
Community.5	0	0	0	11.11%

Case Study 2: Node 100962871525684315897

Using Walktrap

	Circle.1	Circle.2	Circle.3
Community.1	0	0	0
Community.2	6.74%	8.81%	6.21%
Community.3	69.44%	30.86%	11.11%

Using Informap

	Circle.1	Circle.2	Circle.3
Community.1	67.07%	29.61%	10.87%
Community.2	9.24%	11.76%	6.72%
Community.3	33.33%	33.33%	0

Case Study 3: Node 101626577406833098387

Using Walktrap

	Circle.1	Circle.2	Circle.3	Circle.4	Circle.5	Circle.6
Community 1	0.29%	0.22%	0	0.14%	0.43%	1.82%
Community 2	15.61%	18.99%	0.09%	17.29%	0	0.09%

Using Informap

	Circle.1	Circle.2	Circle.3	Circle.4	Circle.5	Circle.6
Community 1	1.10%	1.86%	0	1.35%	0.25%	0.25%
Community 2	0	0.35%	0	0.17%	28.59%	1.24%

The whole community and circle overlapping percentage files are uploaded together with the report. Note that we may leave out some 0 values which means there is no overlapping between communities and circles. Also, we found that Infomap method will usually generate more communities. So we ensure the analysis on same number of communities between Walktrap and Infomap algorithms in order to compare them appropriately. From the output results, we can see that the overlaps vary a lot between different nodes as well as different algorithms. Take node 100535338638690515335 for instance, this node has 4 circles. And we take 5 communities for both Walktrap and Informap algorithm. We can see from the table above that there are more overlapping between communities discovered by Walktrap with its circles than that of Informap. Specifically, Walktrap's community 2 has 62.32% overlap with circle 3 and its community 5 has 80.70% overlap with circle 3. However, communities found by Infomap have much less overlap as shown in the tables. The same situation can also be applied to different nodes. So we can come to a conclusion that there is a huge variation between different nodes and different algorithms. And it is not difficult for us to see that if a person like to tag their community relationships into a circle, there will then be a high overlap between this community and its corresponding circle. For example, if a person tag all his classmates into the classmates circle, since the classmates know each other and there will be a community formed between them. So this classmates circle will overlap a lot with the community discovered by algorithms.

3.Conclusion

This project consists of two parts. In the first part, we are asked to study a real social network - facebook. Firstly, we review some knowledges about network such as connectivity, diameter, degree distribution. And then we studied personal network of a node which is defined to be a graph that consists of this node and its neighbors. Next, we have a review of some community detection algorithms such as fast-greedy, edge-betweeness, infomap and use them to discover the community structure of cores' personal network. Besides, we learned dispersion and embeddedness which are factors can be used to describe the personal networks. Lastly, we discover the inner aspects of one person's life from the personal network. In the second part of this project, we studied Google+ which is a different real social network from Facebook. We created personal networks for users who have more than 2 circles and discovered the communities structures of these personal networks. Then we looked into the data and found the overlaps between the communities and circles and analyzed the overlaps vary across users and algorithms.