

EE 232E  
Graphs and Network Flows  
Project 1  
Winter 2016

Liqiang Yu, Rongjing Bai, Yunwen Zhu  
904592975, 204587519, 104593417

05-14-2016

## Contents

<b>1 Problem 1</b>	<b>3</b>
<b>2 Problem 2</b>	<b>3</b>
<b>3 Problem 3</b>	<b>3</b>
<b>4 Problem 4</b>	<b>6</b>
<b>5 Problem 6</b>	<b>7</b>
5.1 Clustering Coefficient . . . . .	7
5.1.1 Global Clustering Coefficient . . . . .	7
5.1.2 Local Clustering Coefficient . . . . .	7
5.1.3 Network average clustering coefficient . . . . .	9
5.2 Density . . . . .	9
5.3 Results . . . . .	9
<b>6 Problem 7</b>	<b>10</b>

## 1 Problem 1

We read the txt file and construct the graph. After we constructed the graph, we found that the graph is connected.

The diameter is 8.

Intuitively, we believe that the distribution should be

$$\frac{1}{y} \sim a * x^p + b$$

which means frequency should be inverse proportional to  $a * degree^p + b$

After fitting the data, the coefficient is  $a=0.042, b=40.39, p=2.065$

So the model is

$$y = \frac{1}{0.042 * x^{2.065} + 40.39}$$

The MSE for this model is  $2.6 * 10^{-1}$  The average degree is 43.691.

## 2 Problem 2

We constructed the graph shown in figure. The graph has 2866 edges and 348 nodes in total.

## 3 Problem 3

We can find the neighbors of certain node by `neighborhood()` function. Then we find the core nodes with more than 200 neighbors. There are 40 core nodes in total, and their average degree is 279.375. To analyze the personal network, we choose node 1 as the core node. We can see the personal network of node 1 from the figure 1. In this part, we use Fast-Greedy, Edge-Betweenness, and Infomap community these three algorithms in `igraph` to find out the community structure of the personal network of node 1. We can get three different community structure detection results from the graph pairs shown below. In each graph pair, the left subgraph shows the partition results of the community, which are distinguished by different colors. The core nodes in the subgraph are highlighted with larger size. The right subgraph shows the community distribution of the network.

Based on the results of three community detection algorithms, we can find out that there are some differences such as modularity and number of communities besides the community structure. The result is shown in table 1. From the result we can see that these three algorithms have similar modularities, but have great difference in community numbers. Fast-Greedy has the coarsest clusters, while Edge-Betweenness is the finest. However, from figure2 shows several communities in Edge-Betweenness are actually composed by single node, i.e., these nodes do not have tight connection with other nodes in the network. Thus,

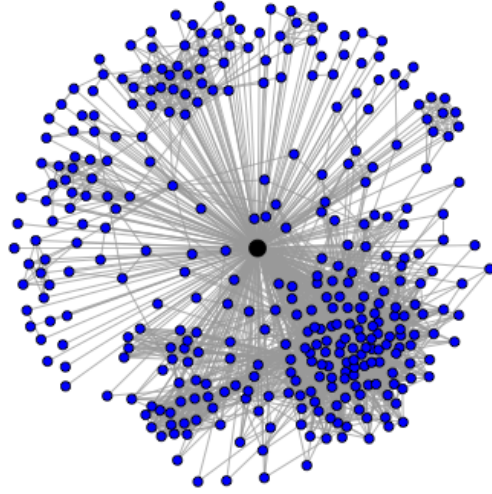
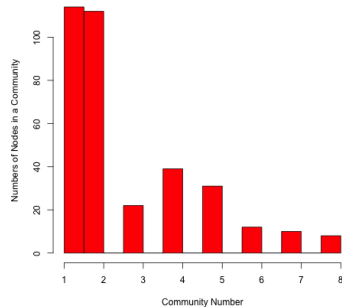
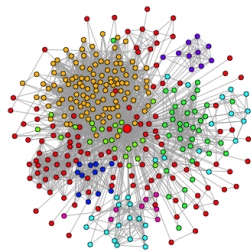


Figure 1: Personal Network of core node 1

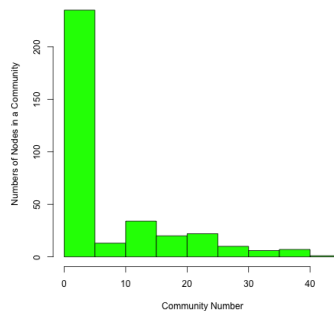
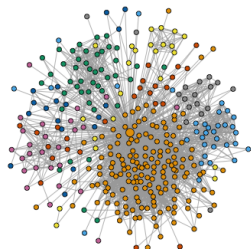
Table 1: The modularity and number of communities comparison results

	fast-greedy	edge-betweenness	infomap
modularity	0.4131014	0.3533022	0.3891184
number of communities	8	41	26



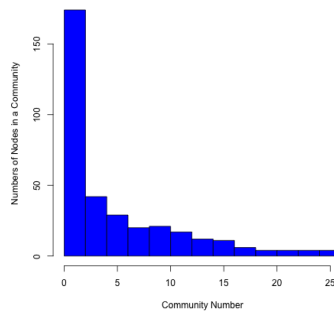
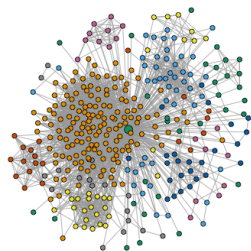
The community structure for fast-greedy method

Community distribution for fast-greedy method



The community structure for edge-betweenness method

Community distribution for edge-betweenness method



The community structure for infomap method

Community distribution for infomap method

Figure 2  
5

the modularity of Edge-Betweenness is relatively lower than that of Fast-Greedy and Infomap.

## 4 Problem 4

After deleting the core node 1, the personal network of core node 1 is shown in figure 3 After applying Fast Greedy, Edge Betweenness and Infomap algorithms



Figure 3: in degree distribution

to the personal network of core node 1 after the deletion, we can explore the community structures and compare them with the network before the deletion. The results are shown in figure 4. The left column are the community structures after applying three kinds of algorithms and the right column are the comparison results of the community distribution before and after the deletion.

From figure 4 we can see that edge betweenness and infomap method almost produced the same results, but fast greedy produced the distribution that moved right a little bit.

Moreover, the modularity results before and after the deletion are shown in

table 2. From the results we can see that infomap produced exactly the same results, however fast greedy and edge betweenness both produced larger results after the deletion, which makes sense. Because the modularity is, quoted from wikipedia, "designed to measure the strength of division of a network into modules (also called groups, clusters or communities). Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules." After deleting the core node, which is pivot node of the its personal network, the network should be easier to divide into modules and therefore the modularity should increase.

Table 2: The modularity comparison results

	fast-greedy	edge-betweenness	infomap
Before deletion	0.4131014	0.3533022	0.3891184
After deletion	0.4418533	0.4161461	0.4180077

## 5 Problem 6

In this problem, we choose to use the cluster coefficient and density to define two types of communities in the 41 personal networks. We try to find the community indices that have maximal and minimal results and compare them to draw the conclusion.

### 5.1 Clustering Coefficient

#### 5.1.1 Global Clustering Coefficient

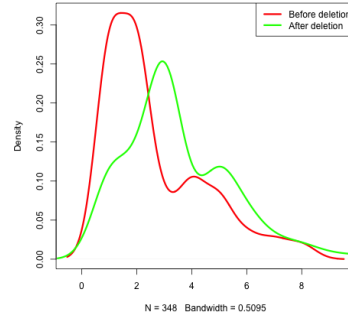
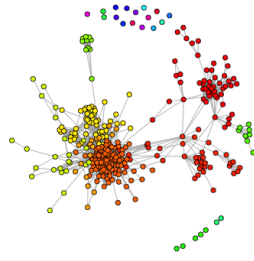
The global clustering coefficient is based on triplets of nodes. A triplet consists of three connected nodes. A triangle therefore includes three closed triplets. Therefore the global clustering coefficient is the number of closed triplets (or 3 x triplets) over the total number of triplets.

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triplets}}$$

#### 5.1.2 Local Clustering Coefficient

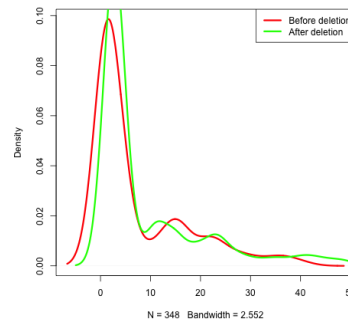
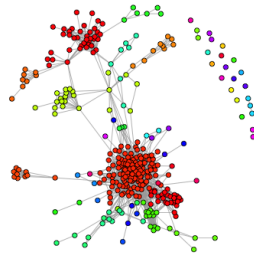
The local clustering coefficient of a vertex in a graph quantifies how close its neighbors are to being a clique. A graph  $G = (V, E)$  formally consists of a set of vertices  $V$  and a set of edges  $E$  between them. A edge  $e_{ij}$  connects vertex  $v_i$  with  $v_j$ . The neighborhood  $N_i$  for a vertex  $v_i$  is defined as its immediately connected neighbors as follows

$$N_i = \{v_j : e_{ij} \in E \cup e_{ji} \in E\}$$



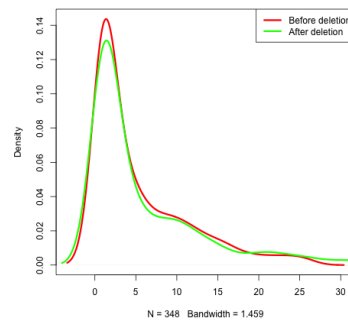
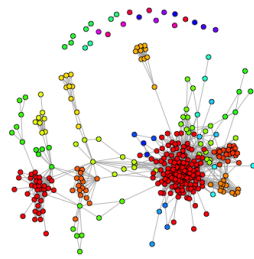
The community structure for fast-greedy method

The comparison results for fast-greedy method



The community structure for edge-betweenness method

The comparison results for edge-betweenness method



The community structure for infomap method

The comparison results for infomap method

Figure 4  
8



we define  $k_i$  as the number of vertices in  $N_i$ . So the local clustering coefficient for undirected graph is

$$C_i = \frac{2|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$$

### 5.1.3 Network average clustering coefficient

As an alternative to the global clustering coefficient, the overall level of clustering in a network is measured as the average of the local clustering coefficients of all the vertices  $n$

$$\bar{C} = \frac{1}{n} \sum_{i=1}^n C_i$$

## 5.2 Density

Network density describes the portion of the potential connections in a network that are actual connections. A potential network is a connection that could potentially exist between two nodes - regardless of whether or not it actually does. That is, for a cluster  $c$ , the density  $d(c)$  is defined as

$$d(c) = \frac{2 \times E(c)}{|c|(|c| - 1)}$$

where  $E(c)$  is the number of edges in cluster  $c$  and  $|c|$  is the number of vertices in cluster  $c$ . High density means there are dense connections inside the cluster therefore a network with high density has large possibility to be a clique.

## 5.3 Results

The results are shown in table 3. We calculate both the cluster coefficient and density of each community that is larger than 10 in each personal network and store the community indices of the largest and smallest values.

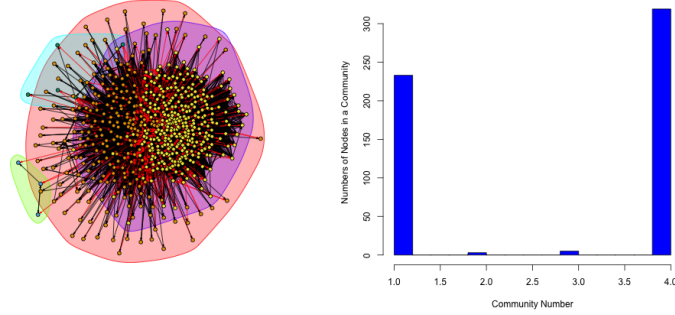
Table 3: The two type communities' indices

type1 cluster	6 3 2 4 1 1 1 2 1 2 1 1 2 2 2 2 1 3 2 1 1 1 2 3 1 2 1 2 3 2 3 1 1 2 1 1 2 1 2 1 2
type1 density	6 3 2 4 1 1 1 2 1 2 1 1 2 2 2 2 1 3 2 1 1 1 2 3 1 2 1 2 3 2 3 1 1 2 1 1 2 1 2 1 2
type2 cluster	7 4 3 5 4 3 2 1 3 1 2 3 1 1 1 1 2 1 3 2 8 2 1 1 2 1 2 1 1 1 1 2 2 1 2 2 1 2 1 2 1
type2 density	7 4 3 5 4 3 2 1 3 1 2 3 1 1 1 1 2 1 3 2 8 2 1 1 2 1 2 1 1 1 1 2 2 1 2 2 1 2 1 2 1

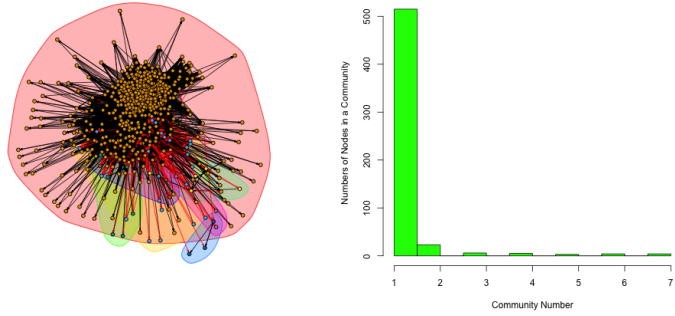
Type 1 are indices with largest values and type 2 are indices with smallest values. From the table 3 we can see that clustering coefficient and density gave exactly the same results. So we can draw the conclusion that both clustering coefficient and density can distinguish two types of communities, one is densely connected inside, maybe a network of high school classmates and another sparsely connected inside, maybe the fans of a sport team.

## 6 Problem 7

In this problem, we analyze Google+ with tagged relationships. Since the core node is not available in the edgelist, we first manually add the ego node and all the corresponding edges into the graph. After exporting the whole network, we can use Walktrap and Infomap algorithms to find the community structure, and we randomly choose to plot the ego node 2's community structure for Walktrap and Infomap algorithm in figure 5.



The community structure for walktrap method of ego node 2      community distribution for walktrap method of ego node 2



The community structure for infomap method of ego node 2      community distribution for infomap method of ego node 2

Figure 5: community structure

To analyze the communities' overlap with the users' circle, we use the Balanced

Error Rate(BER) between the community and the tagged circles. We define the community detected by community detection algorithm as  $C = \{C_1, \dots, C_K\}$  and tagged circle as  $C' = \{C'_1, \dots, C'_K\}$ .

$$BER(C, C') = \frac{|C \setminus C'|}{2|C|} + \frac{|C^c \setminus C'^c|}{2|C^c|}$$

Then we use this equation to compute each community and circle's match to find the most matched circle for each community.  $\lambda = 1 - BER$ , the  $\lambda$  indicates the overlap between each community and circles, i.e the larger  $\lambda$  the community has with a circle, the closer this circle is with the community. On the basis of matching circle for each community, we calculate the number of nodes within the overlap area as well as the number of nodes located outside the circle to get the corresponding overlapping level.

Moreover our analysis is based on two assumptions: (1) The division of community is based on the connectivity of each pair of nodes, thus it is objective. However, the tagging of users are relatively subjective and inconsistent, because it depends on user. So we should use communities to measure circles, not vice versa. (2) Each node can belong to multiple circles, but only one community. So it is more feasible and rational to measure by community.

The results of communities overlap with circles in the two detected algorithms are shown in figure 6 and 7.

From the figure above, we can see that there are some high overlaps among certain users. That means the user tags one person into the circle where there are the majority of members of that community, then this tagging fits the community structure. Thus, it is highly dependent on user's habit on tagging relationships with circles.

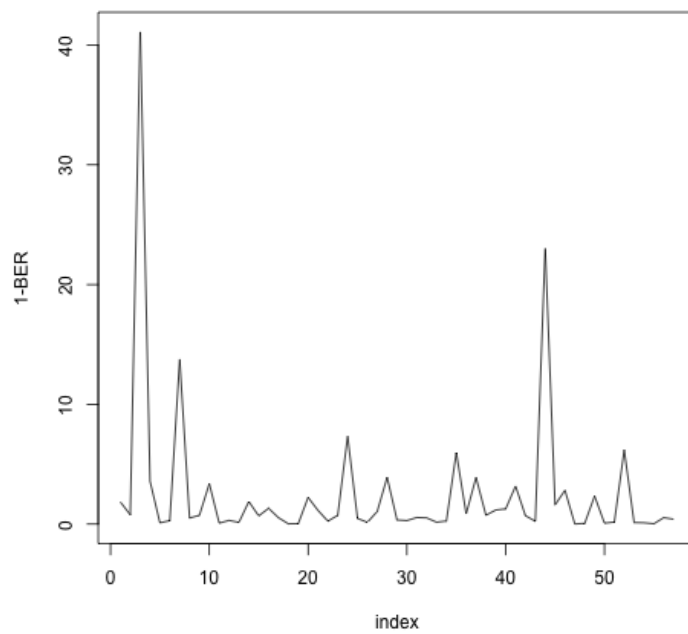


Figure 6: overlap on communities for walktrap method

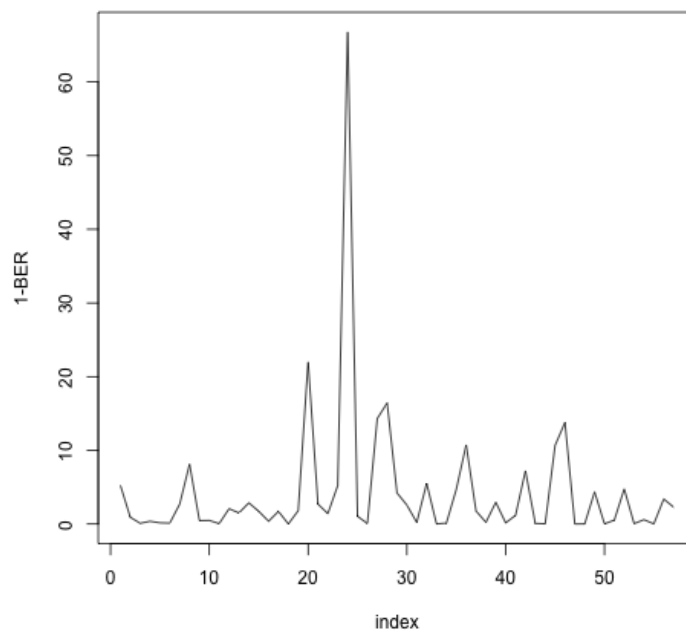


Figure 7: overlap on communities for infomap method