EE 232E
Graphs and Network Flows
Project 1
Winter 2016

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

05-14-2016

Contents

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. The degree distribution is shown in figure 1.

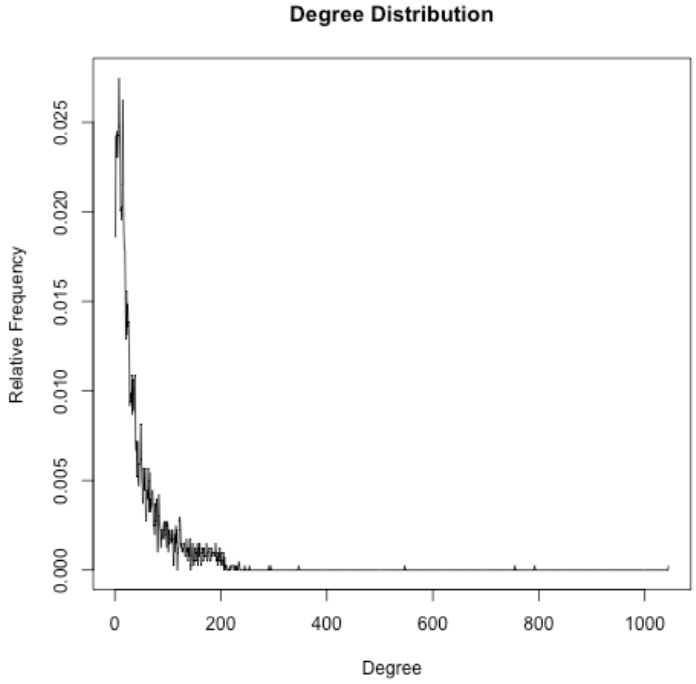


Figure 1: degree distribution

By looking at the graph, 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 coeficient 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 $2.6336 * 10^{-7}$,which is quist low. On top of that, we can see the fitted model in the figure 2, which is very reasonable. So I think is model is very good to describe the degree distribution.

The average degree is 43.691.

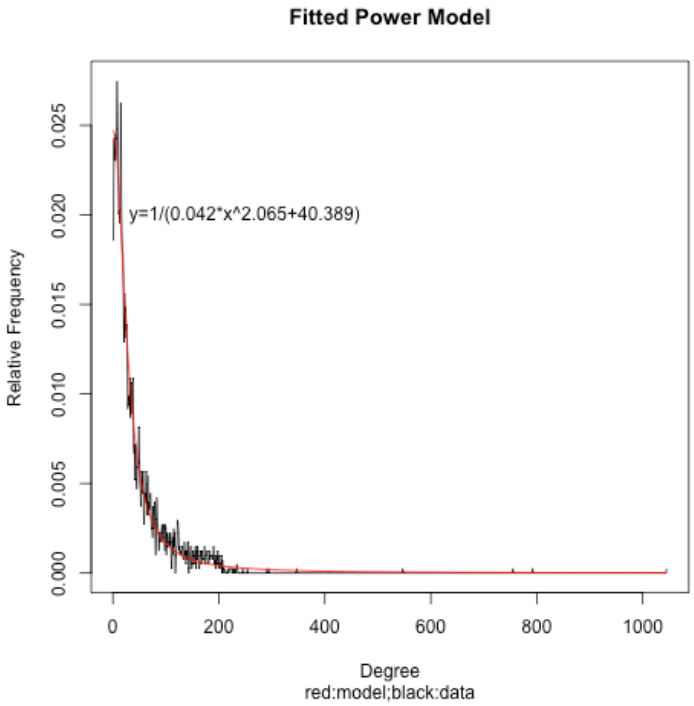


Figure 2: Personal Network of core node 1

2 Problem 2

We constructed the graph shown in figure 3. 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 3. 3 In this part, we use Fast-Greedy, Edge-Betweenness, and

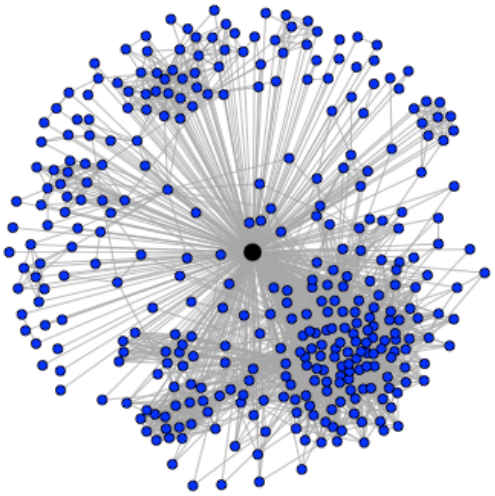


Figure 3: Personal Network of core node 1

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

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

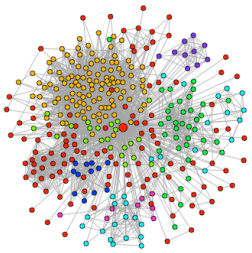
the result we can see that these three algorithms have similar modularities, but have great difference in community numbers. Fast-Greedy has the coarsest clusters, while Edge-Betweenness is the finest. However, from figure 4 shows several communities in Edge-Betweenness are actually composed by single node, i.e., these nodes do not have tight connection with other nodes in the network. Thus, the modularity of Edge-Betweenness is relatively lower than that of Fast-Greedy and Infomap.

4 Problem 4

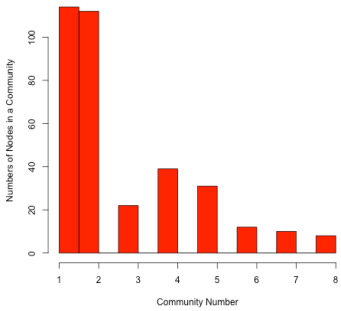
After deleting the core node 1, the personal network of core node 1 is shown in figure 5 After applying Fast Greedy, Edge Betweenness and Infomap algorithms to the personal network of core node 1 after the deletion, we can explore the community structures and compare them with the network before the deletion. The results are shown in figure 6. 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 6 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.

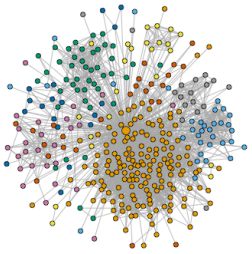
Moreover, the modularity results before and after the deletion are shown in table 2. From the results we can see that all three methods produced larger results after the deletion, which makes sense. Because the modularity is, quoted from wikipedia, "designed to measure the strength of division of a network 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.



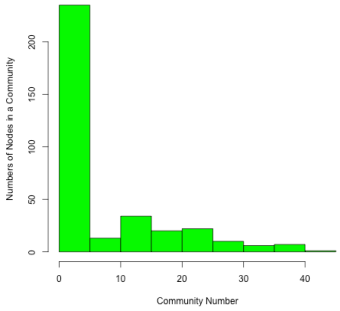
The community structure for fast-greedy method



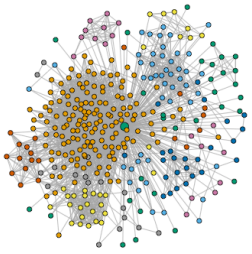
Community distribution for fast-greedy method



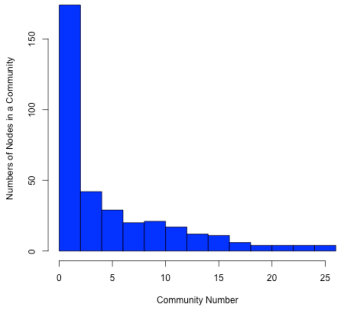
The community structure for edge-betweenness method



Community distribution for edge-betweenness method



The community structure for infomap method



Community distribution for infomap method

Figure 4

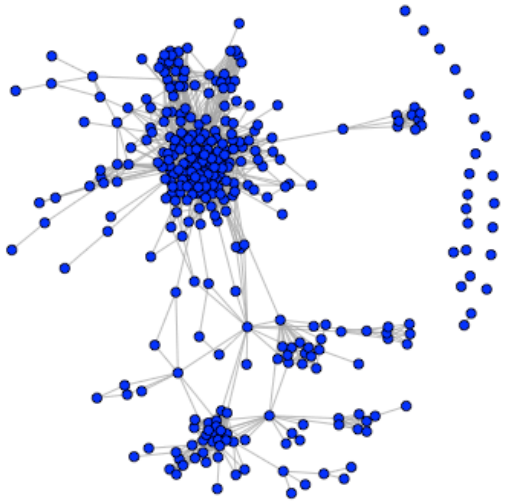
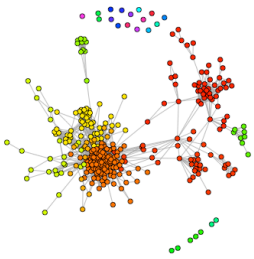


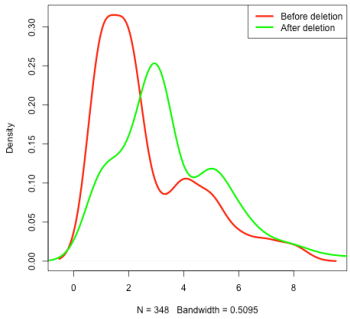
Figure 5: in degree distribution

Table 2: The modularity comparison results

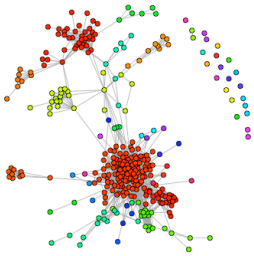
	fast-greedy	edge-betweenness	infomap
Before deletion	0.4131014	0.3533022	0.3891184
After deletion	0.4418533	0.4161461	0.4180077



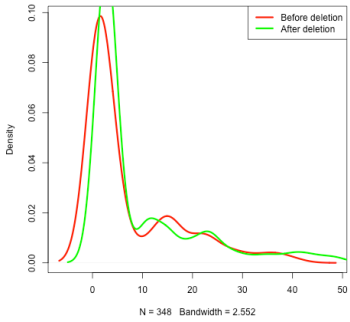
The community structure for fast-greedy method



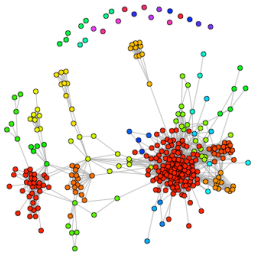
The comparison results for fast-greedy method



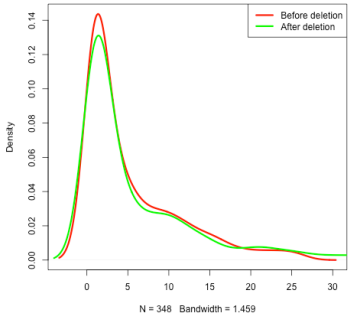
The community structure for edge-betweenness method



The comparison results for edge-betweenness method



The community structure for infomap method

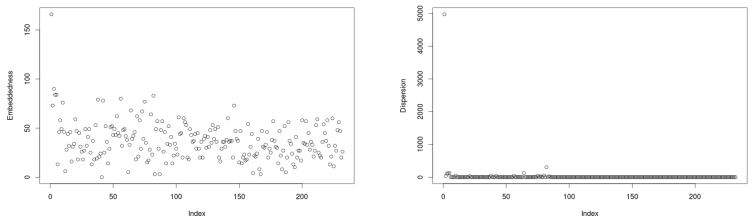


The comparison results for infomap method

Figure 6

5 Problem 5

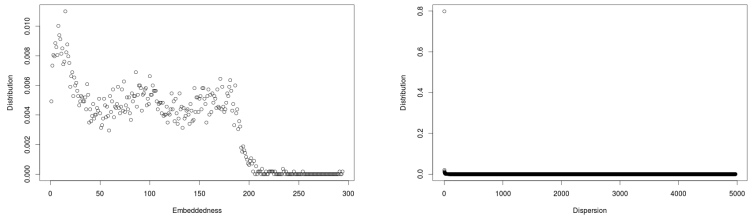
We follow the personal network built in Problem3 and analysis the embeddednes and dipersion of each node. We plot the embeddedness and dispersion in figure 7.



embeddedness dispersion

Figure 7: embedded and dispersion for each node

After collecting the dispersion and embeddedness, we plot the graph for embeddedness and dispersion against index in figure 8.



Embeddedness Dispersion

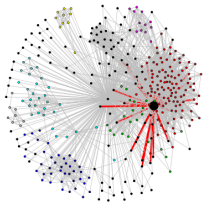
Figure 8: distribution of embeddedness and dispersion

We can see that most of the nodes have a very small dispersion, while embeddedness spread widely in the range in 0 to 200, while a small number of nodes have embeddedness greater than 200.

We choose node 1,379,484 and show using fast greedy to find the node with largest $embeddedness$, $dispersion$ and $\frac{dispersion}{embeddedness}$, shown in figure 9, figure 10 and figure 11

max dispersion

max embeddedness

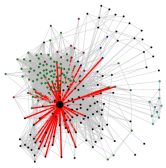


max ratio

Figure 9: Max dispersion, embeddedness and ratio for candidate node 1

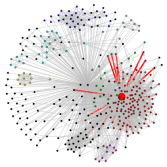
max dispersion

max embeddedness

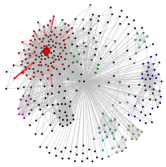


max ratio

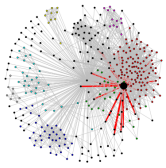
Figure 10: Max dispersion, embeddedness and ratio for candidate node 2



max dispersion



max embeddedness



max ratio

Figure 11: Max dispersion, embeddedness and ratio for candidate node 3

For embeddedness, obviously large embeddedness means two nodes or two persons are closed related because they have many shared friends. However when embeddedness is large, we cannot tell how close the two persons are based on the embeddedness. For example, you and girlfrineds just have 50 mutual friends, while you and your classmantes have 100 mutual friends.

For dispersion, large dispersion should be inversely related to how close it is between two persons. For example, two graduate students have separate social networks in their undergraduate school, so it is normal that one's friends does not know the other's friends in his undergraduate social cycles. However, if they are from the same college, they may know other's friends in their undergraduate social cycles.

For the ratio of embeddedness and dispersion, we think it is a normalized feature to describe the relation between two person. When two person are closely related, they usually have a large embeddedness and small dispersion because they may have almost same network. So it is a better indicator of two persons' relation, whereas high dispersion over embeddedness means not close relation and low ratio means close relation.

6 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.

6.1 Clustering Coefficient

6.1.1 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.

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

6.1.2 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

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

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

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

6.1.3 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

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

6.2 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.

6.3 Resutls

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 sparselyl connected inside, maybe the fans of a sport team.

7 Problem 7

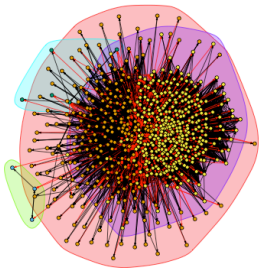
In this problem, we analyze Google+ with tagged relationships. Since the core node is not available in the edgelist, we first manually add the ego node and all the corresponding edges into the graph. After exporting the whole network, we can use Walktrap and Infomap algorithms to find the community structure, and we randomly choose to plot the ego node 2's community structure for Walktrap and Infomap algorithm in figure 12.

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 $C = \{C_1, ..., C_K\}$ and tagged circle as $C' = \{C'_1, ..., C'_K\}$.

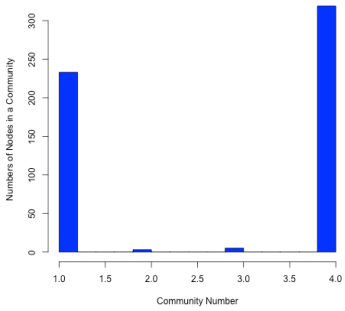
$$BER(C, C') = \frac{|C \backslash C'|}{2|C|} + \frac{|C^c \backslash C'^c|}{2|C^c|}$$

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 λindicates the overlap between each community and circles, i.e the larger λ 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.

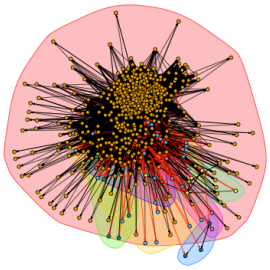
Moreover our analysis is based on two assumptions: (1) The division of community is based on the connectivity of each pair of nodes, thus it is objec- tive. However, the tagging of users are relatively subjective and inconsistent, because it depends on user. So we should use communities to measure circles, 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 overalp with circles in the two detected algorithms are shown in figure 13 and 14.
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.



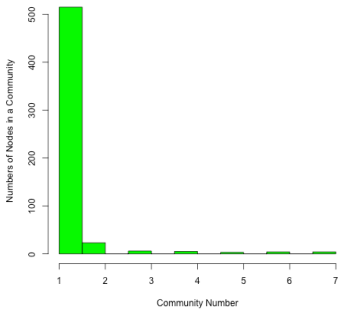
The community structure for walktrap method of ego node 2



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 12: community structure

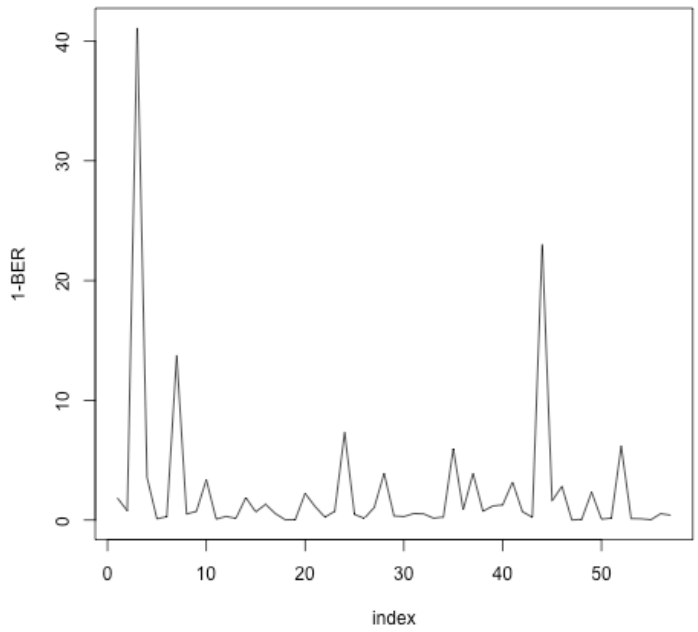


Figure 13: overlap on communities for walktrap method

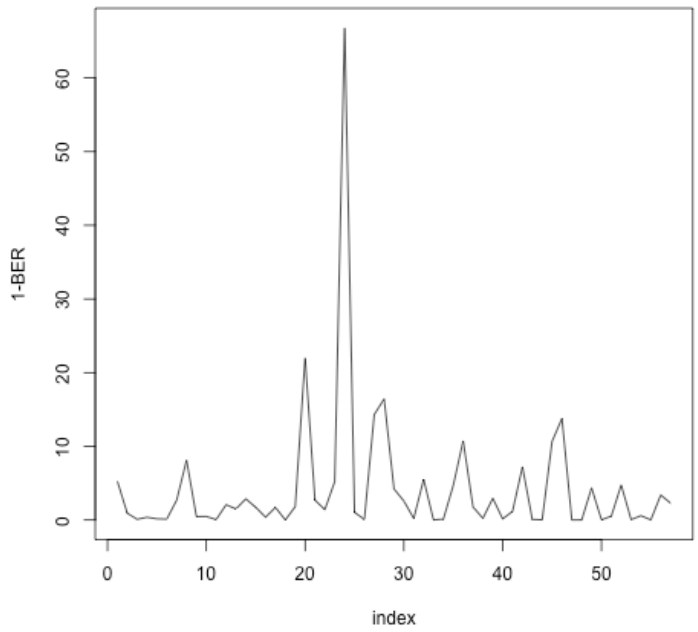


Figure 14: overlap on communities for infomap method