The George Washington University
Columbia College of Arts and Science
Department of Statistic
Section 6289

Midterm Report

Student Name and GWid
Jiaying Liu G45268292

Finished Date
OCT/22/2018

Contents

1 Introduction

Social, technological and information systems can often be described in terms of complex networks that have a topology of interconnected nodes combining organization and randomness. A promising approach consists in decomposing the networks into sub-units or communities, which are sets of highly interconnected nodes. The identification of these communities is of crucial importance as they may help to uncover a priori unknown functional modules such as topics in information networks or cyber-communities in social networks. Moreover, the resulting meta-network, whose nodes are the communities, may then be used to visualize the original network structure.

For the contents in paper Fast Unfolding of Communities in Large Network,they propose a simple method to extract the community structure of large networks. their method is a heuristic method that is based on modularity optimization. It is shown to outperform all other known community detection methods in terms of computation time. Moreover, the quality of the communities detected is very good, as measured by the so-called modularity. This is shown first by identifying language communities in a Belgian mobile phone network of 2 million customers and by analysing a web graph of 118 million nodes and more than one billion links. The accuracy of our algorithm is also verified on ad hoc modular networks.

In this article, there are two impotent equation used in the paper.

- Community, it is defined as

$$Q = \frac{1}{2m}\sum_{i,j}[A_{ij} - \frac{k_i k_j}{2m}]\delta(c_i, c_j)$$

  where A_{ij} represents the weight of the edge between i and j, $k_i = \sum_j A_{ij}$ is the sum of the weights of the edges attached to vertex i, c_i is the community to which vertex i is assigned, the δ function $\delta(c_i, c_j)$is 1 if $u = v$ and 0 otherwise and $m = \frac{1}{2}\sum_{i,j} A_{ij}$.

- the gain in modularity ΔQ obtained by moving an isolated node i into a community C can easily be computed by

$$\Delta Q = [\frac{\sum_{in}+2k_{i,in}}{2m} - (\frac{\sum_{tot}+k_i}{2m})^2] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2]$$

  where $\sum_{in}$in is the sum of the weights of the links inside C, $\sum_{tot}$ is the sum of the weights of the links incident to nodes in C, k_i is the sum of the weights of the links incident to node i, $k_{i,in}$ is the sum of the weights of the links from i to nodes in C and m is the sum of the weights of all the links in the network.

2 Prove of modularity equation

2.1 Question

$$Q = \frac{1}{2m}\sum_{i,j}[A_{ij} - \frac{k_i k_j}{2m}]\delta(c_i, c_j)$$

Since the only contributions to the sum come from vertex pairs belonging to the same cluster, we can group these contributions together and rewrite the sum over the vertex pairs as a sum over the clusters

$$Q = \sum_{c=1}^{n_c} [\frac{l_c}{m} - (\frac{d_c}{2m})^2]$$

Here, n_c is the number of cluster, l_c the total number of edges joining vertices of module c and d_c the sum of the degrees of the vertices of c.

2.2 Prove

- For the equation: we first assume that i and j belong to the one same community c, so the $\delta(c_i, c_j)$is 1

$$\begin{aligned} Q &= \frac{1}{2m} \sum_{i,j} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j) \\ &= \sum_{i,j} \frac{A_{ij}}{2m} - \sum_{i,j} \frac{k_i k_j}{4m^2} \end{aligned}$$

- A_{ij} represents the weight of the edge between i and j,this is also the edge between i and j

- Note that the inward and outward of i and j is the same which is $A_{ij} = A_{ji}$

- So when we add up all the A_{ij}, $\sum_{i,j} A_{ij}$, we can get 2 time of total number of edges joining vertices of module c which is $\sum_{i,j} A_{ij} = 2l_c$

- So we have the first term:

$$\begin{aligned} \frac{\sum_{i,j} A_{ij}}{2m} &= \frac{2l_c}{2m} \\ &= \frac{l_c}{m} \end{aligned}$$

- For the second term, $k_i = \sum_j A_{ij}$ is the sum of the weights of the edges attached to vertex i, which in the same time, is also the concept of degree of node i

- So sum up the k_i, $\sum_i k_i$, we have the sum of degree of the vertices of c, same as k_j

- So for the second term:

$$\begin{aligned} \sum_{i,j} \frac{k_i k_j}{4m^2} &= \frac{d_c^2}{4m^2} \\ &= (\frac{d_c}{2m})^2 \end{aligned}$$

- So we prove that for one cluster, there have such equation, add up all the cluster we can have the equation

$$Q = \frac{1}{2m} \sum_{i,j} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j) = \sum_{c=1}^{n_c} [\frac{l_c}{m} - (\frac{d_c}{2m})^2]$$

3 Prove of ΔQ equation

3.1 Question

$$\Delta Q = [\frac{\sum_{in}+2k_{i,in}}{2m} - (\frac{\sum_{tot}+k_i}{2m})^2] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2]$$

where

$$Q = \frac{1}{2m}\sum_{i,j}[A_{ij} - \frac{k_i k_j}{2m}]\delta(c_i, c_j)$$

haveing $\sum_{in}$in is the sum of the weights of the links inside C, $\sum_{tot}$ is the sum of the weights of the links incident to nodes in C, k_i is the sum of the weights of the links incident to node i, $k_{i,in}$ is the sum of the weights of the links from i to nodes in C and m is the sum of the weights of all the links in the network.

3.2 Prove

- From the definition of gain modularity, it can be write as moving an isolate node i into a community C
- So it can be write as

$$\Delta Q = \text{Q (adding node } i\text{)-(Q(original)+Q(node } i\text{))}$$

- For the first Q, we assume a,b in community C, node i not in the community C,we already have

$$Q = \frac{1}{2m}\sum_{a,b}[A_{ab} - \frac{k_a k_b}{2m}]\delta(c_a, c_b) \quad = \sum_{c=1}^{n_c}[\frac{l_c}{m} - (\frac{d_c}{2m})^2]$$

  $\sum_{ab} A_{ab}$ represent the sum of weight of links inside C, $\sum_{in} = 2l_c$
- When we add the node i, we add the edges inside community C, for the only node i, the total edges of i link to the node in the community C is l_i which is the $k_{i,in}$, note that the inward and outward are the same, so we actually add 2 time of l_i
- Above, we can get the first term of Q(adding node i)

$$\frac{\sum_{in}+2k_{i,in}}{2m}$$

- In the same time, for the second term, $sum_{a,b}k_a k_b = (\sum_a k_a)^2$it represent the total degree of community C which is $\sum_{tot}$, the sum of weight of the link incident to nodes in C.
- So when we add the node i inside the community, we add the degree of node i for second term, for the only node i, the degree of node i is k_i.
- We have $\sum_{tot} + k_i$
- Above, we have the Q(adding node i)

$$Q(add) = \frac{\sum_{in}+2k_{i,in}}{2m} - (\frac{\sum_{tot}+k_i}{2m})^2$$

- For the second Q, we want to estimate the Q of original and the Q of isolate node i
- So it can be written as

$$\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 + \text{Q(node i)}$$

  the original Q is straight forward

- But for the Q(node i), we notice that for the isolate node i, there are no any edge between i and other node, so the first part of Q is 0, however, the degree of node i still contains
- So the Q(node i) can be written as

$$\text{Q(node } i) = 0 - (\frac{k_i}{2m})^2$$

- Above, adding all the fraction of Q, we can get the ΔQ

$$\Delta Q = [\frac{\sum_{in}+2k_{i,in}}{2m} - (\frac{\sum_{tot}+k_i}{2m})^2] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2]$$

4 Network with moduality no larger than -0.5

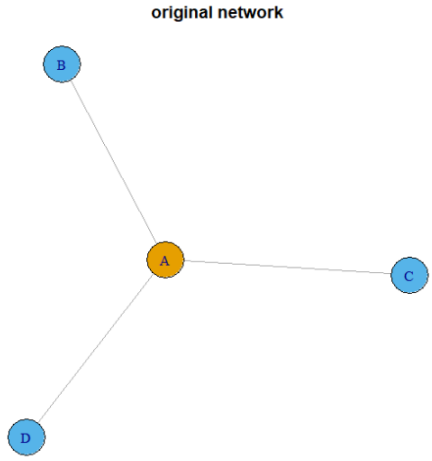
4.1 Question

To construct a network with a community assignment such that the modularity is not large than -0.05

4.2 Result

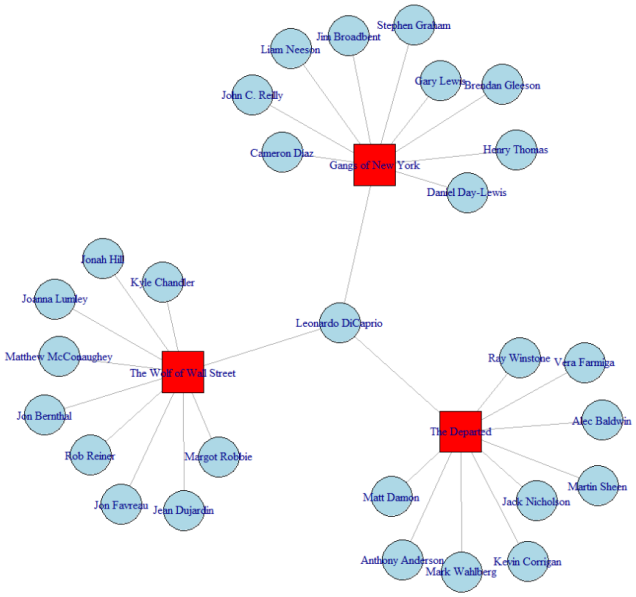
To find out the network which fulfill the above requirement, we first need to know the network performance that can make the modularity not large than -0.05. Notice that the modularity have the range from -0.5 to 1, so the smallest modularity can only be -0.05. Begin with the most simple one, two node defines as two community (cluster) link by only one edge, than the modularity is equal to -0.05.

Figure 1: Basic idea of network with modualrity equal to -0.05



So from the above figure 1, we can find that the two community connect with one edge and there are no edge between same community. So with this basic idea, we can create a more complex network. Reminder of the sub-network about movie and actor, this is a best network to show the modularity. The figure 2 is shown below and the modularity do eqaul to -0.05.

Figure 2: complex network with modualrity equal to -0.05

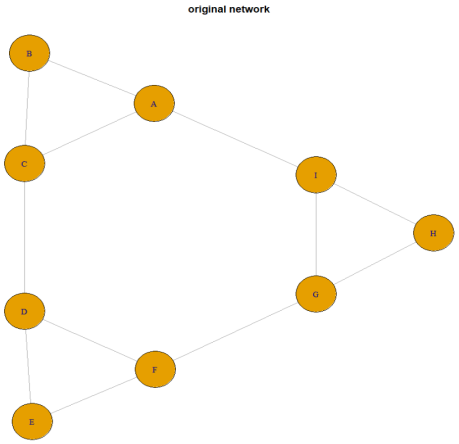


5 Manually community detect

5.1 Question

Based on the paper by Blondel , calculate manually (with stepwise details) the communities detected for the network as below.

Figure 3: Network for question



5.2 Result

5.2.1 Construct network and draw the network

The network show above can be construct as

$\{A\} \rightarrow \{B \quad C \quad I\}$
$\{B\} \rightarrow \{A \quad C\}$
$\{C\} \rightarrow \{A \quad B \quad D\}$
$\{D\} \rightarrow \{C \quad F \quad E\}$
$\{E\} \rightarrow \{D \quad F\}$
$\{F\} \rightarrow \{D \quad E \quad G\}$
$\{G\} \rightarrow \{F \quad I \quad H\}$
$\{H\} \rightarrow \{I \quad G\}$
$\{I\} \rightarrow \{G \quad H\}$

By construct by above method, the network show in the figure 3.

5.2.2 Calculate community

According to the paper, the algorithm is divided into two phases that are repeated iteratively. Assume that starting with a weighted network of N nodes. First, assigning a different community to each node of the network. So, in this initial partition there are as many communities as there are nodes. Then, for each node i, considering the neighbours j of i and we evaluate the gain of modularity that would take place by removing i from its community and by placing it in the community of j.The node i is then placed in the community for

which this gain is maximum (in the case of a tie we use a breaking rule), but only if this gain is positive. If no positive gain is possible, i stays in its original community.This process is applied repeatedly and sequentially for all nodes until no further improvement can be achieved and the first phase is then complete. By using the equation

$$\Delta Q = [\frac{\sum_{in}+2k_{i,in}}{2m} - (\frac{\sum_{tot}+k_i}{2m})^2] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_i}{2m})^2]$$

The second phase of the algorithm consists in building a new network whose nodes are now the communities found during the first phase. To do so, the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities. Links between nodes of the same community lead to self-loops for this community in the new network. Once this second phase is completed, it is then possible to reapply the first phase of the algorithm to the resulting weighted network and to iterate.

- At first, all the nodes are considers as single community with only one node.
- For node A, the neighbour of A is node B,C,I, they have link between these node,so calculate the ΔQ for three pair of node, we have

$$\Delta Q_{AB} = [\frac{0+2}{2*12} - (\frac{2+3}{2*12})^2] - [-(\frac{2}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0625$$
$$\Delta Q_{AC} = [\frac{0+2}{2*12} - (\frac{3+3}{2*12})^2] - [-(\frac{3}{2*12})^2 - (\frac{3}{2*12})^2] = 0.05208$$
$$\Delta Q_{AI} = [\frac{0+2}{2*12} - (\frac{3+3}{2*12})^2] - [-(\frac{3}{2*12})^2 - (\frac{3}{2*12})^2] = 0.05208$$

- For the node A with unlink edge like node H,D etc.We notice that the ΔQ is negative. So we do not need to consider the unlink node, it would no improve the modularity

$$\Delta Q_{AH} = [\frac{0+0}{2*12} - (\frac{2+3}{2*12})^2] - [-(\frac{2}{2*12})^2 - (\frac{3}{2*12})^2] = -0.0208$$
$$\Delta Q_{AD} = [\frac{0+0}{2*12} - (\frac{3+3}{2*12})^2] - [-(\frac{3}{2*12})^2 - (\frac{3}{2*12})^2] = -0.1667$$

- From above, since the $\Delta Q_{AB} > \Delta Q_{AC} = \Delta Q_{AI}$ The node A is then placed in the community node B. For the new community (AB), the $\sum_i n = 2$ and $\sum_{tot} = 5$
- Now consider the community (AB) and its neighbour node C and I, the node C and node D, the node I and node H for these four pair, we have

$$\Delta Q_{(AB)C} = [\frac{2+4}{2*12} - (\frac{5+3}{2*12})^2] - [\frac{2}{2*12} - (\frac{5}{2*12})^2 - (\frac{3}{2*12})^2] = 0.1146$$
$$\Delta Q_{CD} = [\frac{0+2}{2*12} - (\frac{3+3}{2*12})^2] - [(\frac{3}{2*12})^2 - (\frac{3}{2*12})^2] = 0.05208$$
$$\Delta Q_{(AB)I} = [\frac{2+2}{2*12} - (\frac{5+3}{2*12})^2] - [\frac{2}{2*12} - (\frac{5}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0313$$
$$\Delta Q_{IH} = [\frac{0+2}{2*12} - (\frac{2+3}{2*12})^2] - [-(\frac{2}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0625$$

- Since the $\Delta Q_{(AB)C} > \Delta Q_{(AB)I}$ and $\Delta Q_{(AB)C} > \Delta Q_{CD}$ So the node C can be placed into community (AB).And for $\Delta Q_{(AB)I} < \Delta Q_{IH}$, the I do not place into community (AB).For the new community (ABC), the $\sum_i n = 6$ and $\sum_{tot} = 8$

- Consider the new community B and its neighbour node D and I, and the node D with node E, node I with node H,let see whether it can still improve the modularity if we add the new node.

$$\Delta Q_{(ABC)D} = [\frac{6+2}{2*12} - (\frac{8+3}{2*12})^2] - [\frac{6}{2*12} - (\frac{8}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0017$$
$$\Delta Q_{DE} = [\frac{0+2}{2*12} - (\frac{2+3}{2*12})^2] - [-(\frac{2}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0625$$
$$\Delta Q_{(ABC)I} = [\frac{6+2}{2*12} - (\frac{8+3}{2*12})^2] - [\frac{6}{2*12} - (\frac{8}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0017$$
$$\Delta Q_{IH} = [\frac{0+2}{2*12} - (\frac{2+3}{2*12})^2] - [-(\frac{2}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0625$$

- Since the $\Delta Q_{(ABC)D} < \Delta Q_{DE}$ and $\Delta Q_{(ABC)I} < \Delta Q_{IH}$ so this two node can not be placed in two the community B, so the node A,B,C are the same community.

- Now we conside the node D and its neighbour node E and F, we already know $\Delta Q_{(ABC)D} < \Delta Q_{DE}$, so we don't need to consider again. We have

$$\Delta Q_{DE} = [\frac{0+2}{2*12} - (\frac{2+3}{2*12})^2] - [-(\frac{2}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0625$$
$$\Delta Q_{DF} = [\frac{0+2}{2*12} - (\frac{3+3}{2*12})^2] - [-(\frac{3}{2*12})^2 - (\frac{3}{2*12})^2] = 0.05208$$

- From above, since the $\Delta Q_{DE} > \Delta Q_{DF}$ The node D is then placed in the community node E. For the new community node (DE), the $\sum_i n = 2$ and $\sum_{tot} = 5$.

- Now consider the community (DE) and its neighbour node F and community (ABC), the node F and node G, the community (ABC) and node I for these four pair, we have

$$\Delta Q_{(DE)F} = [\frac{2+4}{2*12} - (\frac{5+3}{2*12})^2] - [\frac{2}{2*12} - (\frac{5}{2*12})^2 - (\frac{3}{2*12})^2] = 0.1146$$
$$\Delta Q_{FG} = [\frac{0+2}{2*12} - (\frac{3+3}{2*12})^2] - [(\frac{3}{2*12})^2 - (\frac{3}{2*12})^2] = 0.05208$$
$$\Delta Q_{(DE)(ABC)} = [\frac{2+2}{2*12} - (\frac{5+8}{2*12})^2] - [\frac{2}{2*12} - (\frac{5}{2*12})^2 - (\frac{8}{2*12})^2] = -0.5556$$
$$\Delta Q_{(ABC)I} = [\frac{0+2}{2*12} - (\frac{2+3}{2*12})^2] - [-(\frac{2}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0625$$

- With above result, we can easily find out that node F can be putted into community (DE),For the new community (DEF), the $\sum_i n = 6$ and $\sum_{tot} = 8$.

- Now consider the community (DEF) and the community (ABC) and node G, and node F with H and community B with node I.We have

$$\Delta Q_{(DEF)G} = [\frac{6+2}{2*12} - (\frac{8+3}{2*12})^2] - [\frac{6}{2*12} - (\frac{8}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0017$$
$$\Delta Q_{GH} = [\frac{0+2}{2*12} - (\frac{2+3}{2*12})^2] - [-(\frac{2}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0625$$
$$\Delta Q_{(DEF)(ABC)} = [\frac{2+2}{2*12} - (\frac{8+8}{2*12})^2] - [\frac{2}{2*12} - (\frac{8}{2*12})^2 - (\frac{8}{2*12})^2] = -0.1389$$
$$\Delta Q_{(ABC)I} = [\frac{6+2}{2*12} - (\frac{8+3}{2*12})^2] - [\frac{6}{2*12} - (\frac{8}{2*12})^2 - (\frac{3}{2*12})^2] = 0.0017$$

- Since the $\Delta Q_{(DEF)G} < \Delta Q_{GH}$ and $\Delta Q_{(ABC)I} < \Delta Q_{GH}$ so this two node can not be placed in two the community E, so the node D,E,F are the same community.

- For the rest of node I,H,G, the process just as same as community ABC and community DEF. So in this time, node IGH seem as a community H with $\sum_i n = 6$ and $\sum_{tot} = 8$. There are no more improvement.So we can have the one "pass" new network in figure 5.

Figure 4: Group network

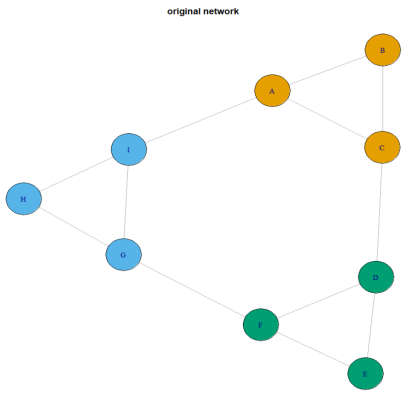
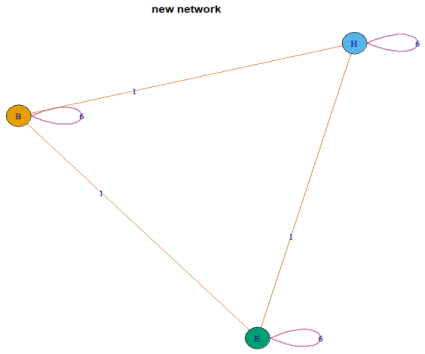


Figure 5: One pass network



- After the one "pass" contraction get the new network, the next step just as above, we set the new community as a single node and calculate each ΔQ

- After calculation, there are no more improvement, so it stop.

$$\Delta Q_{(ABC)(DEF)} = [\frac{6+2}{2*12} - (\frac{8+8}{2*12})^2] - [\frac{6}{2*12} - (\frac{8}{2*12})^2 - (\frac{8}{2*12})^2] = -0.1389$$
$$\Delta Q_{(ABC)(GHI)} = [\frac{6+2}{2*12} - (\frac{8+8}{2*12})^2] - [\frac{6}{2*12} - (\frac{8}{2*12})^2 - (\frac{8}{2*12})^2] = -0.1389$$
$$\Delta Q_{(DEF)(GHI)} = [\frac{6+2}{2*12} - (\frac{8+8}{2*12})^2] - [\frac{6}{2*12} - (\frac{8}{2*12})^2 - (\frac{8}{2*12})^2] = -0.1389$$

5.2.3 Modularity

The modularity of figure 4 is 0.416667.

6 Appendix

```
library(igraph)
g1 <- graph.formula(A-B,A-C, A-D)
V(g1)$group <-c(1,2,2,2)
plot(g1, vertex.color=V(g1)$group,
     vertex.size=20,main='original network')
```

```
modularity(g1, V(g1)$group)

library(UserNetR)
data(hwd)
h1 <- hwd

V(h1)$shape <- ifelse(
    V(h1)$type==TRUE,
    "square","circle")
V(h1)$color <- ifelse(
    V(h1)$type==TRUE,
    "red","lightblue")
h2 <- subgraph.edges(h1,
E(h1)[inc(V(h1)[name %in%
c("The Wolf of Wall Street",
  "Gangs of New York",
  "The Departed")])])
plot(h2, layout = layout_with_kk)

V(h2)$group <- ifelse(
    V(h2)$type==TRUE,
    1,2)
modularity(h2, V(h2)$group)


g2 <- graph.formula(A-B-C-A,I-H-G-I,D-E-F-D,A-I, G-F,C-D)
plot(g2, vertex.color=1,
      vertex.size=20,main='original network')
V(g2)$group <-c(1,1,1,2,2,2,3,3,3)
plot(g2, vertex.color=V(g2)$group,
      vertex.size=20,main='original network')
modularity(g2, V(g2)$group)

g3 <- graph.formula(B-B,E-E,H-H,B-H,B-E,H-E)
g3[from=V(g3), to=V(g3)] <- 1
E(g3)$weight=c(1,1,1,6,6,6)
V(g3)$group=c(1,3,2)
plot(g3,
      vertex.color=V(g3)$group,

      edge.width =ifelse(is.loop(g3),2.5,1.5),

      edge.color=ifelse(is.loop(g3),7, 6),
      edge.label=E(g3)$weight,

      vertex.size=15,main='new network')
modularity(g3, V(g3)$group)
```