Detecting communities from signed network based on local search

Xueyan Liu[1, 2, 3, a], Bo Yang[1, 2, b], Xuehua Zhao[2, 3, c], Yi Yang[4, d]

[1]School of Computer Science and Technology, Jilin University, Changchun 130012, China

[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

[3]School of Digital Media, Shenzhen Institute of Information Technology, Shenzhen 518172, China

[4]China UnionPay, Shanghai 201201, China

[a]email: dyyzlxy@163.com, [b]email: ybo@jlu.edu.cn,

[c]email: lcrlc@sina.com, [d]email: yangyi2@unionpay.com

Keywords: signed networks; community detection; local search

Abstract. Many researchers have begun to study signed networks which are widely existed in real world. In the signed network, the links are labeled the positive or negative sign to represent the active or passive relation between individuals, such as trusted or distrusted relation in social networks. Communities mining is still a great challenge to the domain of signed networks because of negative links. Unlike communities of unsigned networks, positive links mainly occur in the communities and negative links tend to occur between the communities in the signed networks. Nowadays, many methods which are based on global search for signed network community have been raised, and most of these methods require the global information at each iteration. Besides, determining the number of communities is an important problem for current algorithm for the lack of priori knowledge. To address above problems, a novel community detection method based on local information, is proposed for signed networks in this paper. The proposed method mainly includes two steps. In the first step, the number of communities is determined in terms of the centrality of nodes. In the second step, the local objective function is optimized by the local information of nodes, so the global objective function can also be optimized indirectly. Finally, the communities in signed networks are efficiently found. To validate the proposed method, the comparisons are made with other methods in the synthetic and real signed networks. The experimental results indicate that communities in signed networks can be efficiently found by the proposed method.

Introduction

Signed networks whose links can be labeled the positive or negative sign are widely existed in real world. For example, there are friendly or unfriendly relationships in social networks; one can believe or disbelieve others in trust networks; there are cooperative or hostile relationships in the world trade networks. Community structure is the most important topological structure in complex networks [1]. In the unsigned networks, nodes are linked densely with others in the same communities and linked sparsely with other nodes between different communities [2]. In the signed networks, more positive links and less negative links are in the same communities, at the same time, more negative links and less positive links are between different communities [3]. The study of community structure is significant for analyzing the topological structure, function and the changes of the networks.

Nowadays, researchers in various fields are interested in community detection of signed networks, according to balanced theory for this problem they have proposed many optimization algorithms. Doreian and Mrvar have proposed an algorithm (called DM for short) based on optimizing frustration function [4], in other words, they have found community structure to make least positive links between communities and negative links in the same communities. Bansal et al have maximized the agreement function [5] (called AG for short) which is the number of positive links in the same communities and negative links between the different communities or minimized the disagreement function which was coincident with frustration function. Larusso et al have proposed energy function [6] which is similar to agreement function but this algorithm can be used in weighted networks because the weights of the links are considered. However, there are two disadvantages for the current algorithms. Firstly, they expect to know the number of communities in advance, but it is difficult for lack of priori knowledge of networks. Secondly, these methods require global information at each iteration. That means more computing time will be consumed.

To address the above problems, a novel community detection method based on local search (called SLS for short) is proposed for signed networks. In the proposed method, the number of communities can be determined by the information of the network, and community structure of the signed networks can be found by optimizing the local objective function with the local connection information of the nodes. To validate the proposed method, the comparisons are made with other methods in the synthetic and real signed networks. The experimental results indicate that communities in the synthetic and real networks can be efficiently found by the proposed method.

The SLS Algorithm

Define $G = (V, E)$ is a signed network, where V and E denote the sets of nodes and edges respectively. The goal of the SLS algorithm is to find a community partition $\mathrm{C} = (c_1, c_2, \cdots, c_K)$, where c_k denotes a set of nodes which in community k, and the elements in C satisfy $\bigcup_{1\le k\le K} c_k = G$ and $\bigcap_{1\le k\le K} c_k = \emptyset$.

The SLS finds the communities by the two steps as follows: In the first step, the number of communities is determined by the scale-free property of the network and the modified similarity of two nodes. In the second step, the communities is detected by optimizing the local objective function with the local information of nodes. The details of two steps are described as follows:

Step 1: Compute the Number of Communities

In this part, a method for computing the number of communities is given. According to the scale-free property of network, only a few nodes can be seen as “key nodes” with high degree and other nodes are with low degree. Also, “key nodes” are called “central nodes” in their communities because they may be more influential to their neighbor nodes. In Fig.1, red nodes are “central nodes” in their communities.

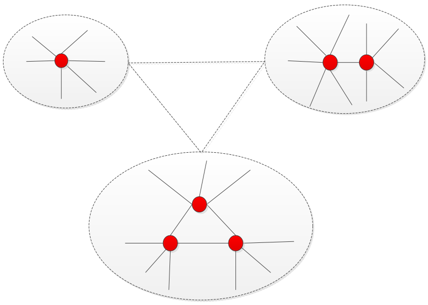


Fig.1 Center nodes in communities

The number of communities can be confirmed by the influential nodes in each community. But there are probably more than one “central node” in a community just like Fig.1. But only one “central node” will be selected in each community. Here, the similarity which is modified from Jaccard coefficient is used to decide whether two “central nodes” are in the same community. If they were, one of them will be randomly selected to represent the community. Finally, the number of selected nodes is also the number of the communities. The similarity is defined as follow：

$$J(i,j) = \frac{|P\Gamma(i) \cap P\Gamma(j)| + |N\Gamma(i) + N\Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|} \tag{1}$$

where $P\Gamma(i)$($P\Gamma(j)$) is the set of nodes which are positively connected with node $i(j)$, and

$N\Gamma(i)(N\Gamma(j))$ is the set of nodes which are negatively connected with node $i(j)$. $\Gamma(i)(\ \Gamma(j)\ )$ is the set of nodes which are connected with node $i(j)$.

Step 2: Detect Community

Then, the method for detecting community is proposed. On the basis of the balance theory for signed networks, Doreian et al have proposed frustration function [4], which is defined as follows:

$$F(C) = \sum_{r=1}^{K}\left(\sum_{i,j\in c_r} A_{ij}^{-} + \sum_{i\in c_r, j\notin c_r} A_{ij}^{+}\right) \tag{2}$$

where K denotes the number of communities, i(or j) is the index of a node, A indicates the adjacent matrix of the network, c_r is the community r. If $A_{ij} > 0$, $A_{ij}^{+} = A_{ij}$; and if $A_{ij} < 0$, $A_{ij}^{-} = -A_{ij}$, that is, $A_{ij} = A_{ij}^{+} - A_{ij}^{i}$. Eq.1 denotes the sum of the number of negative links in the same communities and positive links between the different communities and it exposes the unbalanced level of a signed network. In this paper, it is selected as global objective function. The community detection problem can be transformed to objective function optimizing problem:

$$C^{*} = \arg\min_{C} \sum_{r=1}^{K}\left(\sum_{i,j\in c_r} A_{ij}^{-} + \sum_{i\in c_r, j\notin c_r} A_{ij}^{+}\right) \tag{3}$$

However, it needs global information of all nodes when minimizing the global objective function directly at each iteration. So, the local objective function for node i on signed network is proposed:

$$f(i) = \sum_{1\le j\le n} g(i,j) \tag{4}$$

where, $g(i,j) = \begin{cases} A_{ij}^{-}, & if\ node\ i\ and\ nodes\ j\ is\ in\ the\ same\ community \\ A_{ij}^{+}, & others \end{cases}$.

In accordance with Eq.2 and Eq.4, Eq.5 can be gained:

$$F(C) = \sum_{r=1}^{K}\left(\sum_{i,j\in c_r} A_{ij}^{-} + \sum_{i\in c_r, j\notin c_r} A_{ij}^{+}\right) = \sum_{1\le i\le n} f(i) \tag{5}$$

The process of detecting community is as follows: First, initializing the whole network. Each node randomly gets a community label which ranges from 1 to K, where K is computed in step 1. Randomly selecting a set of nodes and computing the values of these nodes' local function with the community labels. Then, changing these nodes' label to decrease the values of their local faction and updating the values of their neighbor nodes' local function. Next, randomly selecting a neighbor of each node in that set and repeating the procedure above until the number of iterations is up to the limit or the value of the global function does not decrease any more. In the whole procedure, the value of each node's local function decreases continuously by changing its label. Finally, the value of global function is up to minimum and the community structure is gotten.

Description of the SLS

The SLS algorithm includes two steps for community detection on signed networks. In the first step, computing the number of communities based on "center nodes". In the second step, optimizing local objective function by the local information of each node. TABLE 1 shows the procedure of the two steps.

Complexity Analysis

Suppose the number of nodes and communities are n and K respectively, the average degree of the network is d and the number of the candidate nodes is h. The first step is shown in Table 1.1, computing the degrees of all nodes takes $O(nd)$ in line 03-05. Then, the time complex of sorting the nodes based on the degrees is $O(n\log n)$. Selecting center nodes from candidate nodes takes $O(h^2)$ in line11-22. So, the total time complex in step 1 is $O(n\log n)$. The second step is shown in Table 1.2, initializing the nodes with labels takes $O(n)$ in line 04-06. Updating labels of nodes takes $O(kd)$ in line 09, and re-computing the local function values of neighbor nodes takes

$O(kd^2)$ in line11-13. And selecting a neighbor node takes $O(d)$ in line 14. The total time complex in step two is $O(c \cdot kd^2)$, where c denotes the iterations in step 2. On the average condition, the total time complex in two steps is $O(c \cdot kd^2)$. On the worst condition, the each node's degree is $n-1$, so the time complex is $O(c \cdot kn^2)$.

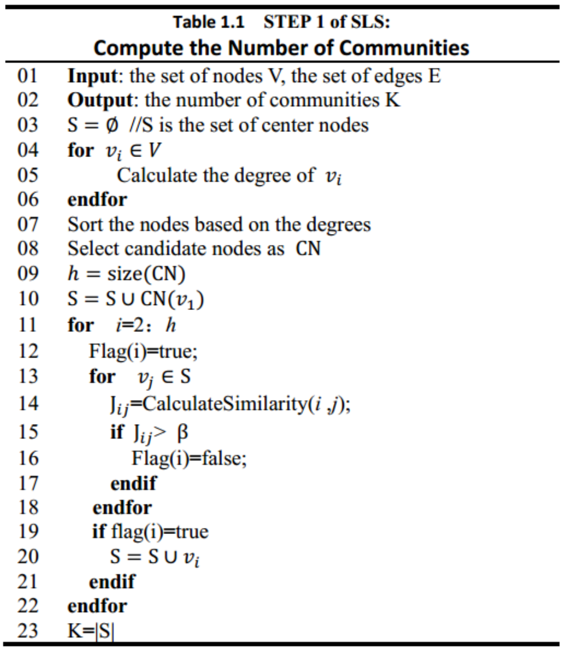


Table 1.1 STEP 1 of SLS:
Compute the Number of Communities

```
01  Input: the set of nodes V, the set of edges E
02  Output: the number of communities K
03  S = ∅  //S is the set of center nodes
04  for $v_i \in V$
05          Calculate the degree of $v_i$
06  endfor
07  Sort the nodes based on the degrees
08  Select candidate nodes as CN
09  $h$ = size(CN)
10  S = S ∪ CN($v_1$)
11  for  $i$=2 : $h$
12     Flag(i)=true;
13     for  $v_j \in$ S
14        $J_{ij}$=CalculateSimilarity($i$ ,$j$);
15        if $J_{ij}$> β
16           Flag(i)=false;
17        endif
18     endfor
19     if flag(i)=true
20        S = S ∪ $v_i$
21     endif
22  endfor
23  K=|S|
```

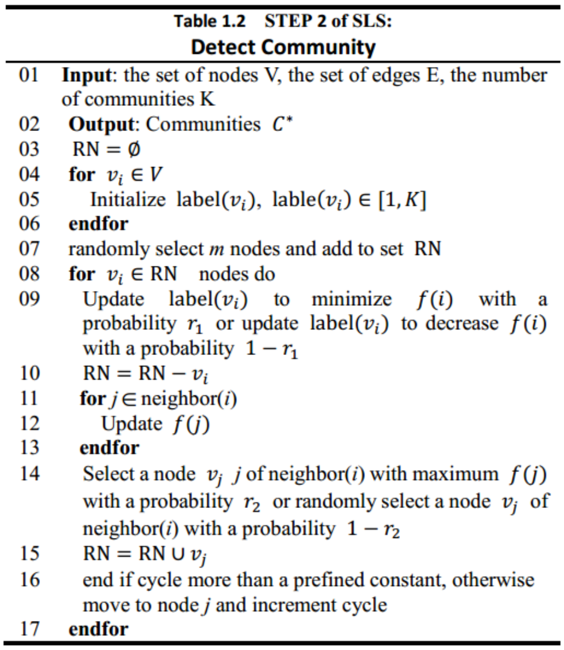


Table 1.2 STEP 2 of SLS:
Detect Community

```
01  Input: the set of nodes V, the set of edges E, the number
    of communities K
02  Output: Communities $C^*$
03  RN = ∅
04  for $v_i \in V$
05     Initialize label($v_i$), lable($v_i$) ∈ [1, $K$]
06  endfor
07  randomly select $m$ nodes and add to set RN
08  for $v_i \in$ RN   nodes do
09     Update label($v_i$) to minimize $f(i)$ with a
       probability $r_1$ or update label($v_i$) to decrease $f(i)$
       with a probability $1 - r_1$
10     RN = RN − $v_i$
11     for $j \in$ neighbor($i$)
12        Update $f(j)$
13     endfor
14     Select a node $v_j$ $j$ of neighbor($i$) with maximum $f(j)$
       with a probability $r_2$ or randomly select a node $v_j$ of
       neighbor($i$) with a probability $1 - r_2$
15     RN = RN ∪ $v_j$
16     end if cycle more than a prefined constant, otherwise
       move to node $j$ and increment cycle
17  endfor
```

Table 1 SLS Algorithm

Results and Discussions

In this section, the performance of SLS will be validated. Because SLS algorithm are based on objective function optimization according to balanced theory, DM [4] and AG [5] which are both based on the same theory as SLS are selected. The difference between SLS and other algorithms is that SLS needs local information but the other two algorithms need global information in each iteration.

The methods are tested in synthetic and real-word networks. The synthetic networks are generated by $\mathrm{SLFR} = (n, k_{avg}, k_{max}, \gamma, \beta, s_{min}, s_{max}, \mu, p-, p+)$ [7]. Here, we set the number of nodes n to 100, the average degree k_{avg} and the maximal degree k_{max} to 10 and 20 respectively, the node degree distribution exponent γ and the community size distribution exponent β to 2 and 1 respectively, the minimal community size s_{min} and the maximal community size s_{max} to 15 and 30 respectively, the mixture parameter μ to [0.1, 0.3, 0.5], the proportion of negative links in the communities $p-$ and the positive links between the communities $p+$ to range from 0 to 1 by steps 0.2. The real-world datasets are two widely-used networks which have been regarded as benchmarks. Gahuku Gama subtribes network [8] describes the political relationships of 16 Gahuku-Gama subtribes. Slovene parliamentary party network which describes the political relationship of 10 parties [9].

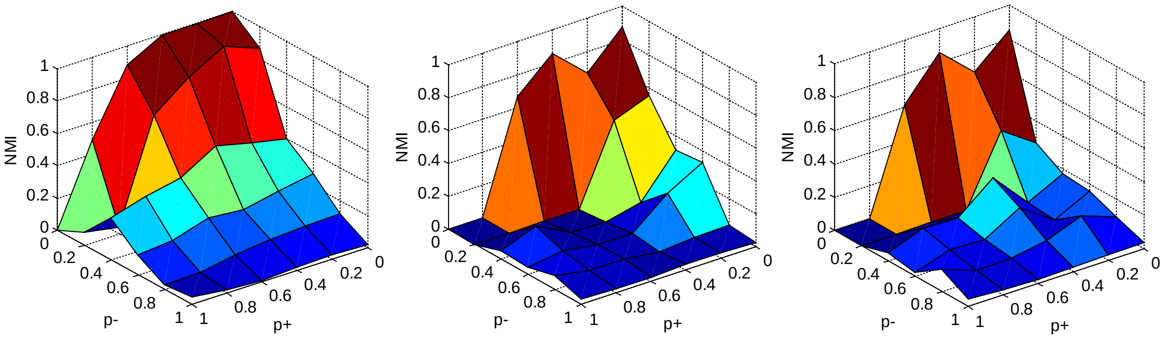
Here, Normalized Mutual Information (called NMI for short) [10] is adopted to estimate the performances of algorithms because the ground truth of above datasets is known. The definition of NMI is as Eq.6,

$$NMI(A,B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} m_{ij}\, log\left(\frac{m_{ij}n}{m_{i\cdot}m_{\cdot j}}\right)}{\sum_{i=1}^{c_A} m_{i\cdot}\, log\left(\frac{m_{i\cdot}}{n}\right) + \sum_{j=1}^{c_B} m_{\cdot j}\, log\left(\frac{m_{\cdot j}}{n}\right)} \quad (6)$$

where A and B are the real and detected community partition respectively and n denotes the number of nodes in the network. M is a confusion matrix of A and B, m_{ij} denotes the count of nodes both in real community i and detected community j.

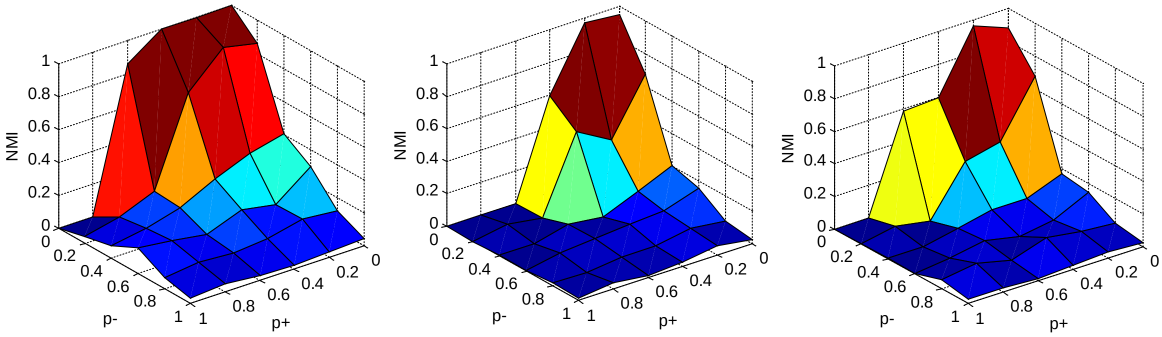
The value of Eq.6 is higher, the result of the algorithm is better. All the nodes are in correct communities when NMI is 1, and all the nodes are in one community when the NMI is 0.

First, we test the algorithms in synthetic networks. In Fig.2 to Fig.4, we can see the results of the algorithms. X-axis and Y-axis denote the noise between and in communities respectively and Z-axis denotes the value of NMI. As we can see from Fig.2 to Fig.4, NMI is lower when mixture parameter is higher. In each Figure, the NMI decreases when the noise increases and these algorithms are more sensitive to negative noise than positive noise. SLS performs better than DM and AG on all the generated datasets.



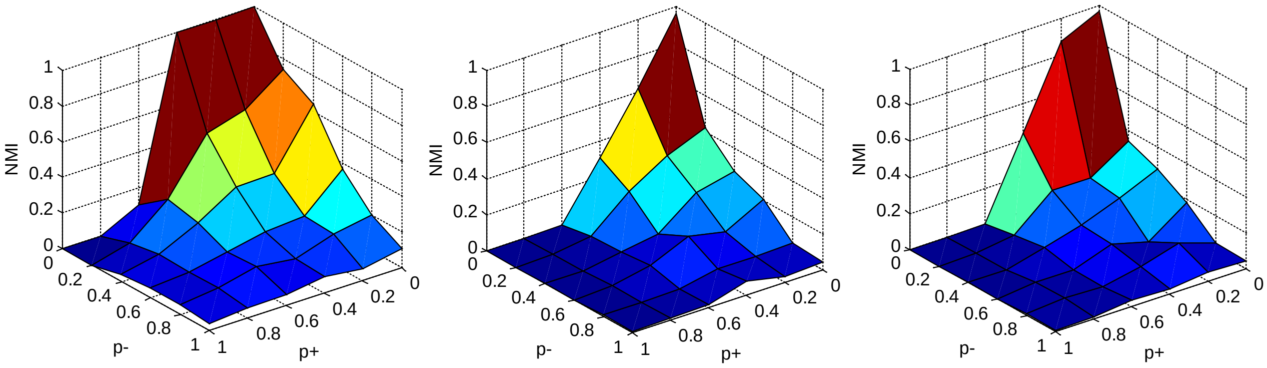
(a) SLS (b) DM (c) AG

Fig.2 The performances of algorithms when u=0.1



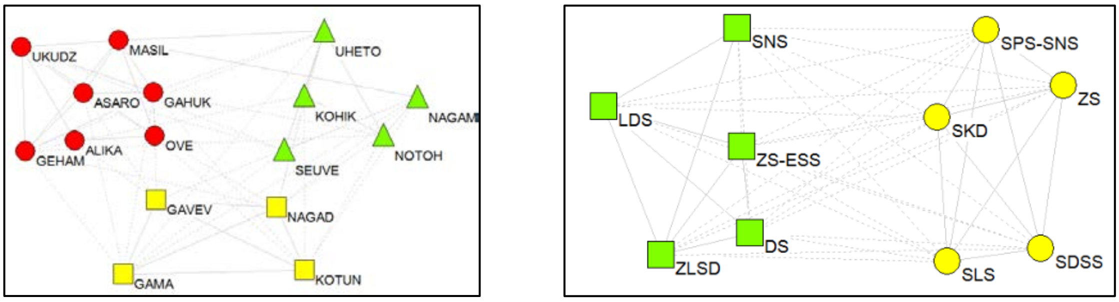
(a) SLS (b) DM (c) AG

Fig.3 The performances of algorithms when u=0.3



(a) SLS (b) DM (c) AG

Fig.4 The performances of algorithms when u=0.5



(a) Gahuku Gama subtribes network (b) Slovene parliamentary party network

Fig.5 The result of SSL on real world networks

Then we test the proposed algorithm on real world networks. In Fig.5, the shapes denote the real communities, and the color of the nodes denotes the detected communities. As we can see, SLS can efficiently find communities correctly in Gahuku Gama subtribes network and Slovene parliamentary party network.

Conclusion

This paper has proposed a community detection algorithm which is based on local search according to balanced theory for signed network. By the scale-free property of the network and the modified similarity of two nodes, the number of communities is determined before detecting. Then, the local objective function is optimized with the local information of each node. With the two steps mentioned above, the communities of signed networks is detected. Compared with other methods, communities in the synthetic and real signed networks can be efficiently found by the proposed method. In the future work, we will improve the algorithm to be efficient for large scale signed networks.

Acknowledgement

This work is funded by the National Natural Science Foundation of China under grants 61373053, 61133011, 61572226, 61300146 and 61272291, Jilin Province Natural Science Foundation under grant 20150101052JC, Guangdong Natural Science Foundation under grant 2016A030310072 and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education Foundation of Jilin University under grant 93K172016K19.

References

[1] Michelle Girvan, M. E. J. Newman. Community structure in social and biological networks[J]. Proceedings of the National Academy of Sciences, 2002, 99(12): 7821-7826.

[2] M. E. J. Newman. Communities, modules and large-scale structure in networks[J]. Nature Physics, 2012, 8(1): 25-31.

[3] Bo Yang, William K. Cheung, and Jiming Liu. Community mining from signed social networks[J]. Knowledge and Data Engineering, IEEE Transactions on, 2007, 19(10): 1333-1348.

[4] Patrick Doreian, Andrej Mrvar. A partitioning approach to structural balance [J]. Social networks, 1996, 18(2): 149-168.

[5] Nikhil Bansal, Avrim Blum and Shuchi Chawla. Correlation clustering[J]. Machine Learning, 2004, 56(1-3): 89-113.

[6] Nicholas Larusso, Petko Bogdanov, Ambuj K. Singh. Identifying communities with coherent and opposing views. In: Proc. of the 15th Annual Graduate Student Workshop in Computing. Santa Barbara: UCSB, 2010. 31−32..

[7]Qing Cai, Maoguo Gong, Shasha Ruan, et al. Network Structural Balance Based on Evolutionary Multiobjective Optimization: A Two-Step Approach[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(6): 903-916.

[8] K. E. Read. Cultures of the central highlands, New Guinea[J].Southwestern Journal of Anthropology, 1954, 10(1): 1–43.

[9] Samo Kropivnik and Andrej Mrvar. An Analysis of the Slovene Parliamentary Parties Network. Developments Statistics and Methodology,A. Ferligoj, A. Kramberger eds. 1996, 209–216.

[10] Ludmila I. Kuncheva and Stefan T. Hadjitodorov. Using diversity in cluster ensembles. Systems, man and cybernetics, 2004 IEEE international conference on. 2004, 2: 1214–1219.