



**HAL**  
open science

## Improved Louvain Method for Directed Networks

Lei Li, Xiangchun He, Guanghui Yan

► **To cite this version:**

Lei Li, Xiangchun He, Guanghui Yan. Improved Louvain Method for Directed Networks. 10th International Conference on Intelligent Information Processing (IIP), Oct 2018, Nanning, China. pp.192-203, 10.1007/978-3-030-00828-4\_20 . hal-02197800

**HAL Id: hal-02197800**

**<https://inria.hal.science/hal-02197800v1>**

Submitted on 30 Jul 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Improved Louvain Method for Directed Networks

Lei Li<sup>1</sup>, Xiangchun He<sup>1</sup>, Guanghui Yan<sup>2\*</sup>

<sup>1</sup> Northwest Normal University, Lanzhou City, -Gansu, -China

<sup>2</sup> Lanzhou Jiaotong University, Lanzhou City, -Gansu, -China  
1029667082@qq.com

**Abstract.** Existing studies about community detection mainly focus on undirected networks. However, research results on detecting community structure in directed networks are less extensive and less systematic. The Louvain Method is one of the best algorithms for community detection in undirected networks. In this study, an algorithm was proposed to detect community structure in mass directed networks. First, the definition for modularity of directed networks based on the community connection matrix was proposed. Second, equations to calculate modularity gain in directed networks were derived. Finally, based on the idea of Louvain Method, an algorithm to detect community in directed networks was proposed. Relevant experiments show that not only does the algorithm have obvious advantages both in run-time and accuracy of community discovery results, but it can also obtain multi-granularity community structure that could reflect the self-similarity characteristics and hierarchical characteristics of complex networks. Experimental results indicate the algorithm is excellent in detecting community structure in mass directed networks.

**Keywords:** Directed Network, Community Detection, Modularity, Modularity Optimization, Louvain Method

## 1 Introduction

Many real systems in the world can be modelled as complex networks. And community structure is common in complex networks [1]. Community structure provides a mesoscale perspective for the study of complex networks, and this characteristic can be used to study the topology and dynamical behavior of networks. Therefore, community detection not only has important theoretical significance, but also has important practical value.

There are two different types of networks: undirected and directed. The research of community detection was initially focused on undirected networks. Currently, there are many different methods to detect community structure in undirected networks [2]. In 2004, Newman et al. [3] proposed *modularity*, which was used originally to measure the accuracy of the results obtained by community detection algorithms and achieved great success in practice. Subsequently, the modularity optimization algorithms which take modularity as the objective function have also become one of the mainstream methods for community detection in undirected networks. Modularity

optimization algorithms mainly include algorithms based on greedy strategy [4-5], extreme value optimization strategy [6], spectral clustering strategy [7] and algorithms which combine multiple strategies [8-9]. The Louvain Method (LM) [8] which integrates greedy strategy and hierarchical clustering strategy has been recognized by many scholars for its low time complexity and high-quality community detection results [10].

In contrast to research on community detection in undirected networks, there are few systematic studies on community detection in directed networks. Currently, there are three main methods of community detection in directed networks:

The first method is to treat the directed network in the same way as an undirected network by ignoring the edge direction of the directed network. However, the direction of the edges in a directed network implies important information and ignoring the direction of the edges to detect community structure will cause inaccurate results [11].

The second method also converts directed networks into undirected networks, but the converted undirected network still contains the information of the direction of the edges. In [12], the authors transformed directed networks into bipartite undirected networks. Then they defined a new modularity for bipartite networks by modifying modularity of undirected networks. Finally, the partition with the largest modularity of bipartite networks is found as the community structure of the networks. This method is inefficient and unsuitable to deal with large-scale networks.

The third method is the modularity optimization method. In 2007, Leicht et al. [11] extended modularity of undirected networks and proposed the modularity of directed networks. Which makes one can design modularity optimization algorithm of directed networks based on the ideas of modularity optimization algorithm of undirected networks. At present, the modularity optimization algorithms of directed networks mainly contain LN algorithm [11] and LLQ algorithm [13]. LN and LLQ can handle directed networks directly, but they both have high time complexity, and the results obtained by these two algorithms also lack accuracy.

This study proposed an algorithm called ILMDN (Improved Louvain Method for Directed Networks) based on the idea of the LM to detect community structure in mass directed networks. The ILMDN not only has low time complexity and high-precision community detection results, but also can obtain multi-granularity community structure that can reflect the self-similarity characteristics and hierarchical characteristics of complex networks.

## 2 Basic Knowledge

The community structure of a network is a division of its node set. Based on the division, the network can be divided into several subgraphs so that the nodes in the same subgraph are closely connected, but the nodes in different subgraphs are sparsely connected. And each subgraph is called a community of the network.

Modularity is a mathematical definition of community structure. In a directed network, let  $A = \{a_{vw}\}$  is its adjacency matrix.  $k_v^{out} = \sum_w a_{vw}$  is the sum of weights for edges starting from node  $v$ , and simply referred to as the *out-degree* of node  $v$ .

$k_w^{in} = \sum_v a_{vw}$  is the sum of the weights for edges which end with node  $w$ , and simply referred as the *in-degree* of node  $w$ .  $m = \sum_{v,w} a_{vw}$  is the sum of weights for all edges. The modularity of directed networks ( $Q_d$ ) is defined as shown in (1), where  $C_v$  is the community that node  $v$  belongs, and the  $\delta$ -function ( $i, j$ ) is 1 if  $i = j$  and 0 otherwise. In general, the community structure of a network is more apparent when  $Q_d$  is larger. And the value range of  $Q_d$  is in  $[-0.5, 1)$ .

$$Q_d = \frac{1}{m} \sum_{v,w} \left[ \left( a_{vw} - \frac{k_v^{out} \cdot k_w^{in}}{m} \right) \cdot \delta_{C_v, C_w} \right] \quad (1)^{[11]}$$

High values of the modularity correspond to good divisions of a network into communities. Therefore, the modularity optimization algorithm finds the division with the highest modularity as network's community structure making it a feasible community detection technology. The LM is one of the superior modularity optimization algorithms for undirected networks. Its main flow is shown in Fig.1. The algorithm is an iterative algorithm and contains two sub-processes in each iteration: modularity optimization and community aggregation. The LM has been recognized by many scholars and has been continuously improved. Many improvement strategies are proposed to further reduce the running time of LM and improve the accuracy of results obtained by LM [14-17].

It should be noted that modularity has some shortcomings in describing the community structure of the network [18-19]. Despite these deficiencies, modularity and modularity optimization algorithms are feasible in practical application.

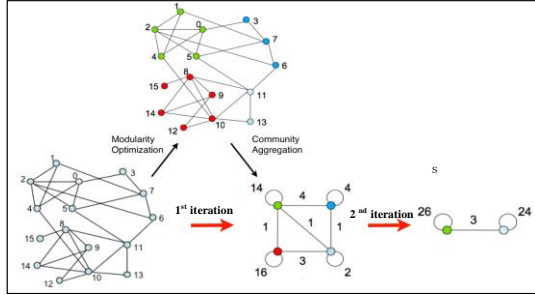


Fig.1. Illustration of the LM algorithm[8]

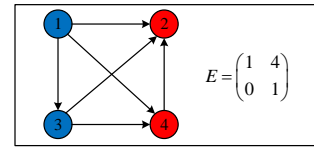


Fig.2. A network and its community connection matrix

### 3 Calculating Modularity Gain in Directed Networks

For the directed network, the gain of the modularity refers to the variation brought to the modularity after the nodes are merged or after the nodes are removed from the community they belong to. In this section, we first derive the definition of directed-network's modularity based on the *community connection matrix* of the network, and then derive the equations to calculate the modularity gain in directed networks.

Supposing that a directed network is divided into  $k$  communities, the  $k$ -th matrix  $E(e_{ij})$  where  $e_{ij} = \sum_{v,w} a_{vw} \cdot \delta_{C_v,i} \cdot \delta_{C_w,j}$  is defined as its community connection matrix. In Fig. 2, there is a directed network with two communities, and matrix  $\mathbf{E}$  is its community connection matrix. It can be known that  $\gamma_i = \sum_j e_{ij} = \sum_v \delta_{C_v,i} \cdot k_v^{out}$ ,  $\beta_i = \sum_j e_{ji} = \sum_w \delta_{C_w,i} \cdot k_w^{in}$ , and  $\delta_{C_v,C_w} = \sum_i \delta_{C_v,i} \cdot \delta_{C_w,i}$ . So, Equation (1) can be equivalently transformed as (2), which is the new definition of the directed network's modularity.

$$\begin{aligned} Q_d &= \frac{1}{m} \sum_{v,w} \left[ \left( a_{vw} - \frac{k_v^{out} \cdot k_w^{in}}{m} \right) \cdot \delta_{C_v,C_w} \right] = \frac{1}{m} \sum_{v,w} \left[ \left( a_{vw} - \frac{k_v^{out} \cdot k_w^{in}}{m} \right) \cdot \sum_i (\delta_{C_v,i} \cdot \delta_{C_w,i}) \right] \\ &= \frac{1}{m} \sum_i \left[ \sum_{v,w} (a_{vw} \cdot \delta_{C_v,i} \cdot \delta_{C_w,i}) - \frac{1}{m} \sum_v (\delta_{C_v,i} \cdot k_v^{out}) \cdot \sum_w (\delta_{C_w,i} \cdot k_w^{in}) \right] = \frac{1}{m} \sum_i \left( e_{ii} - \frac{1}{m} \cdot \gamma_i \cdot \beta_i \right) \end{aligned} \quad (2)$$

Suppose a directed network contains  $k$  communities and  $\mathbf{A}$  is its community connection matrix. Merging community  $i$  and community  $j$  in the directed network, that is, adding the elements of the  $i$ -th row to the  $j$ -th row and add the elements of the  $i$ -th column to the  $j$ -th column in  $\mathbf{A}$ . After the merging, matrix  $\mathbf{A}$  will become as matrix  $\mathbf{B}$ . Then calculate  $Q_d^A$  which is the value of modularity of the network before merging and  $Q_d^B$  which is the value of modularity of the network after merging by using (2). Finally, by subtracting  $Q_d^B$  from  $Q_d^A$  and the equation to calculate modularity gain after the communities are merged in directed network is derived, as shown in (3).

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} e_{11} & \dots & e_{1,i} & \dots & e_{1,j} & \dots & e_{1n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ e_{i,1} & \dots & e_{i,i} & \dots & e_{i,j} & \dots & e_{i,n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ e_{j,1} & \dots & e_{j,i} & \dots & e_{j,j} & \dots & e_{j,n} \\ \vdots & & \vdots & & \vdots & & \vdots \\ e_{n,1} & \dots & e_{n,i} & \dots & e_{n,j} & \dots & e_{n,n} \end{pmatrix} & \quad \mathbf{B} = \begin{pmatrix} e_{11} & \dots & e_{1,i} + e_{1,j} & \dots & e_{1n} \\ \vdots & & \vdots & & \vdots \\ e_{i,1} + e_{j,1} & \dots & e_{i,i} + e_{i,j} + e_{j,i} + e_{j,j} & \dots & e_{i,n} + e_{j,n} \\ \vdots & & \vdots & & \vdots \\ e_{n,1} & \dots & e_{n,i} + e_{n,j} & \dots & e_{n,n} \end{pmatrix} \\ \Delta Q_d^{merge} &= Q_d^B - Q_d^A = \frac{1}{m} \left( e_{ij} + e_{ji} - \frac{\gamma_i \cdot \beta_j + \gamma_j \cdot \beta_i}{m} \right) \end{aligned} \quad (3)$$

Suppose the community which node  $v$  belongs to is  $C_v$  and  $C'_v$  is the community formed after removing node  $v$  and the edge connected to  $v$  from  $C_v$ . The transformation of matrix  $\mathbf{A}$  to matrix  $\mathbf{B}$  corresponds to the merging of nodes, contrarily, the transformation of matrix  $\mathbf{B}$  to matrix  $\mathbf{A}$  corresponds to nodes leaving their community. Therefore, according to the derivation method of (3), the equation to calculate modularity gain after  $v$  removed from  $C_v$  can also be obtained. It is as shown in (4), where  $k_v^{out}$  is the out-degree of node  $v$ ,  $k_v^{in}$  is the in-degree of node  $v$ ,  $\gamma_{C_v}$  is the sum of out-degree of all nodes in  $C_v$ ,  $\beta_{C_v}$  is the sum of in-degree of all nodes in  $C_v$ ,

$e_{v,C'_v}$  is the sum of weights of edges that all start from node  $v$  and end with nodes within  $C'_v$ , and  $e_{C'_v,v}$  is the sum of weights of edges that all start from nodes within  $C'_v$  and end with node  $v$ .

$$\Delta Q_d^{depart} = \frac{1}{m} \left[ \frac{k_v^{out} \cdot (\beta_{C'_v} - k_v^{in}) + k_v^{in} \cdot (\gamma_{C'_v} - k_v^{out})}{m} - e_{v,C'_v} - e_{C'_v,v} \right] \quad (4)$$

Suppose that community  $k$  is the community formed by merging community  $i$  and community  $j$ , then Equations in (5) can be obtained based on the transformation from matrix **A** to matrix **B**.

$$\gamma_k = \gamma_i + \gamma_j, \quad \beta_k = \beta_i + \beta_j, \quad e_{kk} = e_{ii} + e_{jj} + e_{ij} + e_{ji} \quad (5)$$

Suppose that community  $i$  is a sub-graph of community  $j$ , and community  $h$  is the community formed after removing community  $i$  from community  $j$ , then Equations in (6) can be obtained based on the transformation from matrix **B** to matrix **A**.

$$\gamma_h = \gamma_j - \gamma_i, \quad \beta_h = \beta_j - \beta_i, \quad e_{hh} = e_{jj} - e_{ii} - e_{ij} - e_{ji} \quad (6)$$

## 4 Improved Louvain Method for Directed Networks

### 4.1 Symbols and terms

- $G_l = (V_l, E_l, w_l)$ : The directed network generated by the  $l$ -th iteration of ILMDN and is also the input network of the  $(l+1)$ -th iteration.  $V_l$  and  $E_l$  are the node set and edge set of  $G_l$ .  $w_l$  is a mapping from edges to their weights, that is,  $w_l(i, j)$  is the weight of the directed edge starting from node  $i$  and ending with node  $j$ .  $G_0$  is the initial network.

- $C_i^l$ : The community containing node  $i$  in  $G_l$ . It is also the node of  $G_{l+1}$
- $N_l(i) = \{j \mid j \in V_l, i \neq j, (i, j) \in E_l \text{ or } (j, i) \in E_l\}$ : The set of neighborhoods of node  $i$  in  $G_l$ .
- $G = (G_1, G_2, \dots, G_l)$ : The output of ILMDN algorithm. It is the multi-granularity community structure.

### 4.2 The algorithm

The main procedure of ILMDN is shown in **Algorithm 1**, and it is composed of two phases: iteration phase and refinement phase. The iteration phase corresponds to the LM algorithm and contains two sub-phases: modularity optimization (steps 3 to 15) and the community combination (steps 18 to 21). The refinement phase corresponds to the improvement strategy proposed in [14] for undirected networks. This

improved strategy was extended to make it suitable for the directed network.

---

**Algorithm 1: ILMDN**


---

**Input:** Initial network  $G_0$  .

**Output:** Multi-granularity community structure  $G$  .

**Local variables:**  $l$  denotes the iteration times of iteration phase.  $t$  denotes the traverse times on the node set of input network in each iteration. The initial values of  $l$  and  $t$  both are 0. *Increase* identifies whether there is node movement in a traverse, and its initial value is *true*.

```

1 While ( true )
2    $l \leftarrow l + 1$  ;
3   For each  $i$  of  $V_{l-1}$  do
4      $V_l.Add(i)$  ;
5   While increase do
6     increase  $\leftarrow$  false ;
7      $t \leftarrow t + 1$  ;
8     For each  $i$  of  $V_{l-1}$  do
9       Equation (4) is used to calculate  $\Delta Q_d^{depart}$  that is the gain of  $Q_d$  obtained by removing node  $i$  from its community. Equation (3) is used to calculate gains of  $Q_d$  obtained by merging node  $i$  with each of its neighboring communities, and the maximum gain named as  $\Delta Q_d^{max\ Enter}$  is selected among them. Finally, let the NO. of the neighboring community corresponding to  $\Delta Q_d^{max\ Enter}$  is MaxCid;
10      if ( $\Delta Q_d^{max\ Enter} \leq 0$  &&  $0 < \Delta Q_d^{depart}$ )
11        increase  $\leftarrow$  true , make node  $i$  leave its community and become a community alone;
12      else if ( $\Delta Q_d^{max\ Enter} + \Delta Q_d^{depart} > 0$ )
13        increase  $\leftarrow$  true , make node  $i$  leave its community and merge into community whose NO. is MaxCid ;
14      else
15        make  $i$  stay in its community;
16    if ( $t == 1$ )
17      The iteration phase is over, and the refinement phase is executed;
18    For each  $i$  of  $V_{l-1}$  do
19       $w_l(C_i^{l-1}, C_i^{l-1}) \leftarrow w_l(C_i^{l-1}, C_i^{l-1}) + w_{l-1}(i, i)$  ;
20    For each  $j$  of  $N_{l-1}(i)$  do
21       $w_l(C_i^{l-1}, C_j^{l-1}) \leftarrow w_l(C_i^{l-1}, C_j^{l-1}) + w_{l-1}(i, j)$  ;
22     $G.Add(G_l)$  ;

```

---

In the  $l$ -th iteration of iteration phase, first assign a different community to each node of  $G_{l-1}$  in steps 3 ~ 4. So, the number of communities equals the number of nodes in initial. Then traverse  $V_{l-1}$  several times until there is no change on communities of every nodes in steps 5~15. In each traversal, there are three cases of commu-

nity ownership for node  $i$ . The first case corresponds to steps 10~11, in which removing node  $i$  from its community can increase the value of  $Q_d$ , however if node  $i$  is ultimately merged into any neighboring community, the value of  $Q_d$  will be reduced. The second case corresponds to steps 12~13, in which removing node  $i$  from its community and then merging it with *MaxCid* community can increase the value of  $Q_d$  the most. The third case corresponds to steps 14~15, in which removing node  $i$  from its community would decrease the value of  $Q_d$ , and then merging node  $i$  with any neighboring community would reduce the value of  $Q_d$  further. In steps 18 ~ 21, the edge set  $E_l$  and the mapping  $w_l$  are generated by traversing  $G_{l-1}$  one time. It should be noticed that there is no node movement at the last traversal on nodes set of input network in each iteration. Therefore, if a certain iteration has only traversed the node set of the input network for one time, the iteration phase is terminated, and then goes to perform the refinement phase (step 16 ~ 17). The main flow of refinement phase is shown as **Function 1**. In the refinement phase, an iterative process is actually executed.

---

**Function 1: Refinement Phase**

---

1. Obtain each node's community after the iteration phase based on  $G_0, G_1, \dots,$  and  $G_{l-1}$ ;
  2. Select  $G_0$  as input to implement modularity optimization sub-phases once;
  3. Implement community merged sub-phases based on  $E_0$  once;
  4.  $G.Add(G_l)$ ;
- 

### 4.3 The time complexity and space complexity of ILMDN

Let  $l$  denote the iteration times of iteration phase and  $t$  denote the traverse times on the node set of input network in each iteration. Numerous experiments show that  $l$  and  $t$  are constants independent of the scale of input network [8,14-17], and their values are small. In addition, the variables needed when calculating the modularity gain can be updated in real time according to Equations in (5) and (6), so the time consumed to calculate the modularity gain is a constant time  $O(1)$ . Therefore, the time complexity and spatial complexity of ILMDN are both  $O(n+m)$  for a network with  $n$  nodes and  $m$  edges.

## 5 Experimental Comparison and Analysis

To verify the ILMDN's performance, based on the six directed networks in Table 1, contrast experiments between ILMDN, LN and LLQ were performed.

First, ILMDN has a deficiency that it sensitives to the input sequence of initial-network's nodes, that is, different node input sequences will result in different community structure. However, the difference between the different community structures is insignificant, and the most of the results when input sequences of nodes are random



are close to or the best community structure of the network. For example, when ILMDN was ran 1000 times on the Directed LM Network, three different results shown in Fig. 3 were obtained. To simplify the display, two directed edges between two nodes in the network were replaced by an undirected edge in the figure. The times that three results appeared from left to right are respectively 2, 84 and 914. There is little difference between the three results and the rightmost result, which appeared the most, is the best community structure of the network.

**Table 1.** The 6 directed networks

Name	Introduction	Nodes	Edges
Directed LM Network	It is generated by converting every undirected edge in LM Network shown in Fig.1 as two directed edges that are in the opposite direction.	16	56
Directed Karate Clubs Network	It is generated by converting every undirected edge in Karate Club Network [20] as two directed edges that are in the opposite direction.	34	156
Wikipedia Vote Network	The network is constructed by SNAP using Wikipedia user voting data [21].	7115	103689
Email Communication Network	The network is constructed by SNAP using Enron's Email Dataset [21].	265214	420045
Bank Customer Transaction Network	The network is generated using customer transaction records of a commercial bank in the first quarter of 2015. Nodes of the network are customer's accounts, and the weight of edge $A \rightarrow B$ is the cumulative times of transactions from the account A to the account B.	278565	313520
Wikipedia Talk Network	The network is constructed by SNAP using Wikipedia page edit data [21].	2394385	5021410

Table 2 shows the performances of LN, LLQ and ILMDN for community detection in the six directed networks. For each algorithm/network, the table displays the modularity that is achieved and the computation time  $T$  (in seconds). Because ILMDN is sensitive to the input sequence of nodes, ILMDN was ran repeatedly on each network in experiment: run ILMDN 1000 times on the first three networks and 300 times on the latter three networks. Then for each network, the maximum, minimum and average computation time ( $T_{\max}, T_{\min}, T_{ave}$ , respectively) consumed by ILMDN were counted. The maximum, minimum, and average values of modularity ( $Q_d^{\max}, Q_d^{\min}, Q_d^{ave}$ , respectively) achieved by ILMDN were also counted. In the table, the time "0.0" indicates that the algorithm consumes less than one millisecond when running on a smaller network, and "—" indicates that the algorithm failed to detect the community structure of the network within one hour when the network scale was large. As can be seen from the table, the computation time of ILMDN was far less than that of LN and LLQ. Even for networks with more than two million nodes, ILMDN detected its community structure in about a minute. In terms of modularity, first, for each network, the maximum modularity achieved by ILMDN was greater than or equal to the modularity which LN and LLQ achieved. Second, because the result that ILMDN obtained on the network when input sequence of nodes is random is close to or the

best community structure of the network, the average values of modularity ILMDN achieved at each network was close to the maximum value. Finally, on some networks, the smallest modularity ILMDN achieved was still greater than the modularity LLQ obtained. In summary, compared with LN and LLQ, ILMDN has obvious advantages in terms of computation time and accuracy of community discovery results.

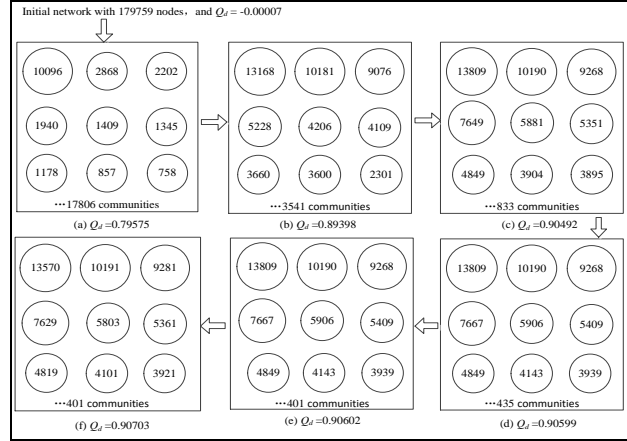
**Table 2.** Summary of numerical results

Networks	LN		LLQ		ILMDN		
	$Q_d$	$T$	$Q_d$	$T$	$Q_d^{\max}$ , $Q_d^{\min}$ , $Q_d^{ave}$	$T_{\max}$ , $T_{\min}$ , $T_{ave}$	
Directed LM Network	0.392	0.0	0.392	0.0	0.392, 0.344, 0.390	0.0, 0.0, 0.0	
Directed Karate Clubs Network	0.420	0.2	0.381	0.0	0.420, 0.392, 0.418	0.0, 0.0, 0.0	
Wikipedia Vote Network	0.419	13.1	0.332	7.4	0.434, 0.414, 0.428	1.4, 0.4, 0.7	
Email Communication Network	—		0.727	2827.0	0.782, 0.769, 0.777	4.0, 2.4, 3.1	
Bank Customer Transaction Network	—		0.926	317.5	0.928, 0.926, 0.927	5.2, 3.7, 4.3	
Wikipedia Talk Network	—		—		0.571, 0.561, 0.566	71.3, 45.4, 57.5	

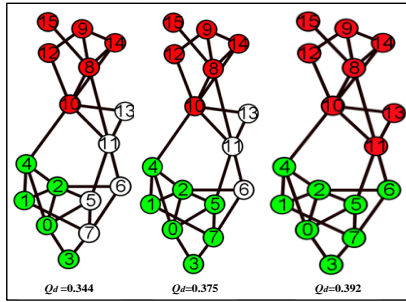
ILMDN was also compared to the LM. It is found that the community structure ILMDN detected in Directed LM Network is the same as the community structure the LM mined in LM Network, and the results are the same for Directed Karate Clubs Network and Karate Clubs Network. The result further verifies the accuracy of ILMDN.

In addition, the ILMDN can provide a multi-granularity community structure based on the intermediate partitions found at each iterative. The multi-granularity community structure of the largest isolated subgraph of Bank Customer Transaction Network is shown in Fig.4. There are 37,979 isolated subgraphs in Bank Customer Transaction Network, while the largest isolated subgraph contains 179,759 nodes and the other isolated subgraphs are smaller (the second largest isolated subgraph contains only 121 nodes). In order to display the results more clearly, only the multi-granularity community structure of the largest isolated subgraph was shown in Fig.4. There was a total of 6 iterations when ILMDN was ran on the largest isolated subgraph. The community structures detected from the first iteration to the fifth iteration are shown from Fig.4(a) to Fig.4(e). Note that the result of the sixth iteration is the same as the fifth iteration and is not shown. Fig.4(f) shows the community structure detected after the refinement phase. Each chart in Fig.4 shows the modularity of the result, the total number of communities and the top nine communities with the most nodes in the corresponding results. In every chart, the circles indicate the communities, and the numbers in the circles indicate the numbers of nodes in the community. And Fig.5 shows the internal structure of a community generated by the second iteration. There are five communities in the figure, which are all generated by the first iteration. Multi-granularity community structure embodies the self-similarity and hierar-

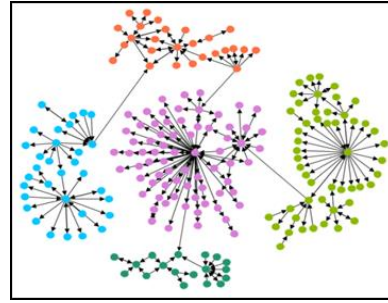
chical characteristics of complex networks, which provides important information for subsequent research on the network.



**Fig. 4.** The multi-granularity community structure of the largest isolated subgraph of Bank Customer Transaction Network



**Fig. 3.** Different community structures detected by ILMDN in Directed LM Network



**Fig. 5.** The internal structure of a community in Fig.4 (b).

As can also be seen from Fig.4, the modularity gain obtained by the first iteration is significantly higher than that of other iterative obtained. The community structure detected after the refinement phase had the same community number as the community structure detected after the iteration phase, but the community ownership of some nodes changed, so that the modularity that ILMDN achieved finally was improved.

## 6 Conclusion and Discussion

In this paper, we proposed an algorithm named ILMDN for detecting community structure in directed networks. ILMDN not only has linear time complexity, but also has higher accuracy of community detection results. And ILMDN can obtain multi-granularity community structure which contains important information for subsequent

research on the networks. Future research will be focused on solving ILMDN's sensitivity to node input sequence, as well as detecting overlapped community in directed networks.

## References

- [1] Girvan M, Newman M E J. Community structure in social and biological networks [J]. Proceedings of the National Academy of Sciences of the United States of America, 2002, 99(12):7821-7826.
- [2] Liu DY, Jin D, He DG, Community Mining in Complex Networks [j]. Journal of Computer Research and Development, 2013, 50(10):2140-2154.
- [3] Newman M E, Girvan M. Finding and evaluating community Structure in networks [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2004, 69 (2 Pt 2):026113-026113.
- [4] Newman M E J. Fast algorithm for detecting community structure in networks [J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2004, 69(6):066133-066133.
- [5] Clauset A, Newman M E J, Moore C. Finding community structure in very large networks.[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2004, 70(2):066111.
- [6] Duch J, Arenas A. Community Detection in Complex Networks Using Extremal Optimization[J]. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2005, 72(2 Pt 2):027104.
- [7] Newman M E J. Modularity and community structure in networks[J]. Proceedings of the National Academy of Sciences, 2006, 103(23):8577-82.
- [8] Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks [J]. Journal of Statistical Mechanics Theory & Experiment, 2008, 2008(10):155-168.
- [9] Liu X, Murata T. Advanced modularity-specialized label propagation algorithm for detecting communities in networks [J]. Physica A Statistical Mechanics & Its Applications, 2009, 389(7):1493-1500.
- [10] Lancichinetti A, Fortunato S. Community detection algorithms: A comparative [J]. 2009.Physical Review W,2009,80(5):056117
- [11] Leicht E A, Newman M E J. Community structure in directed networks.[J]. Physical Review Letters, 2007, 100(11):118703.
- [12] Guimerà R, Salespardo M, Amaral L A. Module identification in bipartite and directed networks[J]. Physical Review E, 2007, 76(3 Pt 2):036102.
- [13] Liu Y, Liu Q, Qin Z. Community Detecting and Feature Analysis in Real Directed Weighted Social Networks[J]. Journal of Networks, 2013, 8(6).
- [14] Gach O, Hao J K. Improving the Louvain Algorithm for Community Detection with Modularity Maximization[M]// Artificial Evolution. 2013:145-156.
- [15] Bhowmick S, Srinivasan S. A Template for Parallelizing the Louvain Method for Modularity Maximization [M]// Dynamics On and Of Complex Networks. New York: Springer, 2013:111-124.
- [16] LIU Y, KANG X H,Gao H ,et al. A Community Detecting Method Based on the Node Intimacy and Degree in Social Network [J]. Journal of Computer Research and Development, 2015, 52(10):2363-2372.
- [17] De Meo P, Ferrara E, Fiumara G, et al. Generalized Louvain method for community detection in large networks[C]//International Conference on Intelligent Systems Design &Applications. Piscataway: Institute of Electrical and Electronics Engineers, 2011:88-93.
- [18] Fortunato S, Barthélemy M. Resolution limit in community detection. Proceedings of the National Academy of Sciences of the United States of America, 2007, 104(1):36-41.
- [19] Kim Y, Son S W, Jeong H. Finding communities in directed networks. Physical Review E Statistical Nonlinear & Soft Matter Physics, 2010, 81(2):016-031.
- [20] Zachary W W. An Information Flow Model for Conflict and Fission in Small Groups[J]. Journal of Anthropological Research, 1977, 33(Volume 33, Number 4):452-473.
- [21] Jure Leskovec and Andrej Krevl. SNAP Datasets: Large Network Dataset Collection. <http://snap.stanford.edu/data>, last accessed 2017/11/6.