LPANNI: Overlapping Community Detection Using Label Propagation in Large-Scale Complex Networks

Meilian Lu, Zhenglin Zhang, Zhihe Qu and Yu Kang

Abstract—Overlapping community structure is a significant feature of large-scale complex networks. Some existing community detection algorithms cannot be applied to large-scale complex networks due to their high time or space complexity. Label propagation algorithms were proposed for detecting communities in large-scale networks because of their linear time complexity, however most of which can only detect non-overlapping communities, or the results are inaccurate and unstable. Aimed at the defects, we proposed an improved overlapping community detection algorithm, LPANNI (Label Propagation Algorithm with Neighbor Node Influence), which detects overlapping community structures by adopting fixed label propagation sequence based on the ascending order of node importance and label update strategy based on neighbor node influence and historical label preferred strategy. Extensive experimental results in both real networks and synthetic networks show that, LPANNI can significantly improve the accuracy and stability of community detection algorithms based on label propagation in large-scale complex networks. Meanwhile, LPANNI can detect overlapping community structures in large-scale complex networks under linear time complexity.

Index Terms—label propagation algorithm, large-scale network, neighbor node influence, overlapping community detection

1 Introduction

LARGE-SCALE complex networks with tens of thousands or millions of nodes have intricate community structures. Detecting community structures in large-scale complex networks helps to analyze group characteristics.

In recent years, researchers have proposed many community detection algorithms [1-4] for detecting non-overlapping communities. However, most of the real networks have overlapping community structures in which one network node may belong to multiple communities. For example, an author in a co-author network may collaborate with several research groups, a user in a social network may join in multiple interest groups. Although some algorithms [5-7] were proposed for detecting overlapping community structures, due to high time or space complexity, most of them cannot be applied to large-scale complex networks. So, overlapping community detection algorithms with low time complexity [8, 9] are of great concern.

Label Propagation Algorithm (LPA) [10] was first proposed by Raghavan et al., which has linear time complexity and was proved to be able to quickly and effectively detect community structures in large-scale complex networks. However, LPA has some shortcomings: 1) it can only detect non-overlapping communities; 2) it suffers from low accuracy in many networks due to considering that the importance of all neighbor nodes are the same when propagating labels; 3) the randomness of label propagation results in instability of community detection; 4) the randomness of label propagation also produces the problem of "label diffusion". That is, a community constantly assimilates some other communities and forms a very big community, which may also decrease the accuracy of community detection. To overcome the shortcomings of LPA, some enhanced algorithms [11-15] were proposed, however these algorithms can only detect non-overlapping communities.

Given that LPA and its enhanced algorithms cannot detect overlapping communities in complex networks, COPRA (Community Overlap PRopagation Algorithm) [16], SLPA (Speaker-Listener Label Propagation Algorithm) [17] and DLPA (Dominant Label Propagation Algorithm) [18] were proposed, which adopt multiple labels and belonging coefficients to identify the possibility that a node belongs to multiple communities. However, these improved methods did not adequately measure the different influence of neighbor nodes, which leads to lower accuracy of community detection. In addition, some parameters in these methods need to be manually set using priori knowledge, which may greatly affect the results of community detection. Finally, these methods still did not well solve the instability problem of community detection algorithms based on label propagation.

Some methods, such as NIBLPA [19], LPA_NI [20], WLPA [21] and CK-LPA [12], proved that considering node importance and label influence can improve the accuracy and stability of community detection algorithms based on label propagation. However, only WLPA was reported to be able to detect overlapping communities.

In order to detect overlapping communities in large-scale networks with low time complexity and improve the accuracy and stability of community detection, we propose an advanced label propagation algorithm, LPANNI (Label Propagation Algorithm with Neighbor Node Influence), which detects overlapping communities by adopting fixed

- *Meilian Lu, Zhenglin Zhang, Zhihe Qu and Yu Kang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. 100876. E-mail: mllu@bupt.edu.cn.*

label propagation sequence based on node importance and label update strategy based on neighbor node influence and historical label preferred strategy.

The main contributions of this paper are summarized as follows: 1) Three new measures, respectively, node importance (NI), similarity between nodes (Sim), neighbor node influence (NNI), is defined to more comprehensively reflect the different influence of neighbor nodes during label propagation and thus increase the accuracy of community detection; 2) A fixed label propagation sequence based on the ascending order of node importance is defined, and a label updating strategy based on neighbor node influence and historical label preferred strategy is proposed, both of which can efficiently avoid the randomness and accelerate the convergence of label propagation, and thus improve the stability of LPA algorithms.

The rest of the paper is organized as follows. Section 2 reviews the related literatures, especially the LPA based algorithms for overlapping community detection which consider node importance or label influence. Section 3 defines several terms associated with node importance and neighbor node influence, a new label update strategy is also defined here. Section 4 describes our proposed LPANNI in detail. Section 5 gives the experimental results and analysis. Finally, Section 6 concludes the paper.

2 Related Works

Community structure is a significant feature of complex networks and is commonly defined as that nodes are much more connected to each other in the same community than they are to the nodes in other communities [1]. Many algorithms were proposed to detect the potential network community structures, such as hierarchical clustering [2], modularity optimization based method [3], density-based method [4], etc. However, they can only detect non-overlapping community structures.

Some algorithms utilizing local expansion and optimization were proposed to detect overlapping community structures, however, these algorithms excessively depend on a local benefit function that characterizes the quality of densely connected node groups. OSLOM (Order Statistics Local Optimization Method) [6] detects overlapping communities by locally optimizing the statistical information of clusters with respect to the random fluctuation using Extreme and Order Statistics. Considering that the overlapping areas between communities are denser than the non-overlapping areas of the communities, Bandyopadhyay et al. [8] proposed the FOCS algorithm, which first detects some initial communities based on the degree of nodes, then iteratively moves a node into a community or generate a new community according to community connectivity and neighborhood connectivity. Chakraborty et al. [9] introduced a vertex based metric, GenPerm, to measure the possibility that a vertex belongs to one or more network components. GenPerm method is used to detect both overlapping and non-overlapping communities by maximizing the GenPerm metric for all vertices in a network. Both FOCS and GenPerm have low time complexity and good community detection effect.

Some fuzzy clustering algorithms were also proposed to detect overlapping communities. Eustace et al. [7] proposed NRATIO (Neighborhood RATIO) algorithm to detect overlapping communities by combining neighborhood ratio matrix and Non-negative Matrix Factorization. However, the space and time complexity of NRATIO is too high to apply to large-scale complex networks.

LPA [10] is a fast non-overlapping community detection algorithm. In which, each node is initially given a unique label, such as node id, then its labels are iteratively and asynchronously updated according to its neighbors' labels. When the iterations end, the nodes with the same label are considered to be in the same community and non-overlapping communities are detected. LPA is an efficient and effective community detection algorithm for large-scale networks because of its linear time complexity and without pre-defined parameters. However, LPA suffers from the instability problem because it updates node labels in random order and random label selection strategy. Although some improved label propagation algorithms [11-15] were proposed to increase the accuracy and stability of LPA, they still cannot detect overlapping communities.

COPRA [16] is the first advanced version of label propagation algorithms for detecting overlapping community structures. In which, each node has up to v labels, and the belonging coefficient of each label identifies the strength that the node is a member of the corresponding community. During the label propagation, the node updates its labels and the corresponding belonging coefficients according to its neighbor nodes' labels. However, COPRA may produce many small and meaningless communities for some networks.

SLPA [17] utilizes label propagation based on speaker-listener rule to detect overlapping communities. In which, each node stores the labels accepted during each iteration in memory. When iterations end, the probability of one label occurring in the memory is considered as the strength that the node belongs to the corresponding community. However, to generate overlapping communities, a post-processing parameter r needs to be pre-defined, and its optimal value is hard determined for most networks. Besides, COPRA also thinks that the importance of all neighbor nodes is the same when updating node labels, which may produce high randomness of label propagation and result in low accuracy and stability.

DLPA [18] introduces confidence to measure the importance of each neighbor node and thus to increase the accuracy of community detection. In addition, DLPA introduces dominant label to decrease the complexity of label propagation and defines an inflation factor in to control the overlapping ratio. However, the confidence uses less information of neighbor nodes and cannot really reflect the difference between the influences of neighbor nodes. Also the quality of the detected communities relies on the value of in. In addition, like COPRA and SLPA, DLPA did not solve the instability problem of LPA that the communities detected by multiple repetition experiments on the same network may be very different.

To improve the accuracy and stability of community detection algorithms based on LPA, some new algorithms, such as NIBLPA, LPA_NI, CK-LPA and WLPA, introduce two important factors, node importance and label influence, to update nodes' labels.

NIBLPA [19] measures node importance by synthetically considering node's k-shell value, node's degree and its neighbors' k-shell value, and updates nodes' labels in descending order of node importance. It also measures label influence by considering the importance and degrees of neighbor nodes, and then selects the label with high influence to update node labels. However, the more important nodes are more likely to be the potential community centers, and their labels are updated first. So the range of label propagation may be limited because the labels of the important nodes are early locked, which may reduce the accuracy of community detection. Besides, although the node importance can be measured using k-shell decomposition, there may be many nodes with the same k-shell value, and the importance of these nodes cannot be well distinguished.

Based on NIBLPA, considering that the difference of node importance between different types of networks, LPA_NI [20] measures node importance using the semantic information of networks. Specifically, aimed at Sina Micro-blog, the priori importance of nodes is learned by expert knowledge from Bayesian network. However, for different networks, the priori importance is different either, this method is not universal.

CK-LPA [12] also measures node importance using kernel weight and node degree. Moreover, nodes' labels are updated in descending order of node importance, and the label with the maximum sum of node weight is selected as new label. However, how to set the size and number of kernels is a problem.

Although the three algorithms above improve the accuracy and stability of LPA, they cannot detect overlapping communities. WLPA [21] was reported to be able to detect overlapping communities and improve the accuracy and stability of LPA by introducing node importance and label influence. Given the different influence of neighbor nodes, WLPA gives each label a weight to distinguish labels' influence and measures node importance using degree centrality. Different from the three algorithms above, WLPA updates node labels in ascending order of node importance to improve the stability. It also updates node label by asynchronous way to further improve the stability. However, when measuring the node importance and label influences, only the direct relationship between adjacent nodes are considered. So in case that a node belongs to more than one community, it tends to choose the community where the node with larger neighbor degree belongs, which may affect the results of overlapping community detection.

Compared to the algorithms above, the main innovations of our proposed LPANNI algorithm are reflected in the following two aspects: 1) it introduces a new metric, neighbor node influence (NNI), to more legitimately measure the influence of different neighbor nodes when updating node's labels, thus increases the accuracy of overlapping community detection; 2) it combines the sequence of node importance and historical label preferred strategy to update nodes' labels, which significantly decreases the instability of label propagation algorithm.

3. Terms and Label Update Strategy

In LPA, each node has only one label and updates its label iteratively according to its neighbor nodes' labels. When finishing the iterations, the nodes with the same label are partitioned to one community and non-overlapping communities are detected. But in real networks, a node may naturally belong to more than one community. As shown in Fig. 1, node 1 belongs to two adjacent communities.

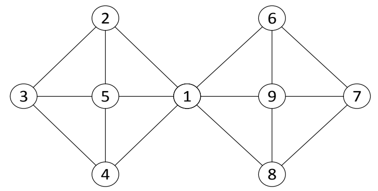


Fig.1. A sample network

As analyzed in Section 2, to detect overlapping community structures, COPRA [16], SLPA [17] and DLPA [18] were proposed. However, these methods don't solve the instability problem of LPA. NIBLPA [19], LPA_NI [20], WLPA [21] and CK-LPA [12] can improve the accuracy and stability of LPA by considering node importance and label influence. However, in terms of node importance, the measure in WLPA is not comprehensive enough to improve the accuracy, the measure of LPA_NI depends on priori knowledge, and the measures of NIBLPA and CK-LPA depend on k-shell, which may not well distinguish the importance of different nodes. Moreover, most of them cannot detect overlapping communities. So we propose LPANNI to detect overlapping community in large-scale networks and improve the accuracy and stability of LPA.

In this section, we first define node importance, similarity between nodes, and neighbor node influence. Then, we elaborate our proposed label update strategy based on neighbor node influences.

3.1 Definitions of Terms

Considering the limitations of the above schemes, in this part, we define three new metrics.

Definition 1. *Node Importance (NI): For a given network* $G=(V,E)$ *, node importance describes the probability that one node is the potential community center.*

One common assumption that one node may be the potential community center is that the node has following features: it connects with more neighbor nodes and these neighbor nodes connect with each other intensively. The more neighbor nodes a node connects with, the larger the degree of the node. And, the closer a node connects with its neighbor nodes, the more triangles may be formed by the node and its neighbor nodes. It means that if a node has a larger degree and forms more triangles with its neighbor nodes, it may be the central node of a potential community and may influence more neighbor nodes.

As shown in Fig. 1, node 5 has more edges connected to itself neighbor nodes than node 3, and node 5 connects with its neighbors more intensively than node 3 does, so node 5 tends to be the center of a potential community.

Based on the assumptions above, we use the degree of node $u \in V$ and the number of the triangles formed by the node and its neighbor nodes to measure its importance.

Meanwhile, in order to avoid bigger difference between the importance values of different nodes and reflect the influence of *NI* on label propagation, the node importance is normalized to $[1/2,1]$. So node importance is defined as:

$$NI(u)=\frac{1}{2}+\frac{1}{2}*\frac{(e_u+k_u)-\min\limits_{i\in V}(e_i+k_i)}{\max\limits_{j\in V}(e_j+k_j)-\min\limits_{i\in V}(e_i-k_i)} \tag{1}$$

Where e_u denotes the number of triangles containing node u , and k_u is the degree of node u .

For weighted network $G=(V,E,w)$, we define:

$$e_u = \sum_{x,y \in Ng(u), e_{xy} \in E} \frac{w(u,x)+w(u,y)+w(x,y)}{3} \quad (2)$$

$$k_u = \sum_{v \in Ng(u)} w(u,v) \quad (3)$$

Where $Ng(u)$ denotes the neighbor node set of node u and $w(x,y)$ denotes the weight of edge e_{xy} between nodes $x,y \in V$.

As we known, for weighted networks, the weights of different edges are different. So we get e_u according to the average weight of the three edges in all the triangles containing node u . Also we get k_u by taking the sum of all the weights of edges connected to node u . Therefore, the definition of NI can also be applied to weighted networks.

Definition 2. *Similarity between nodes (Sim): Similarity between nodes reflects the connection strength between two adjacent nodes, which is used to measure the node influence.*

One of the most commonly used similarity measures is Katz function. However, it cannot effectively distinguish the influence of the degree difference between two nodes on node similarity. Moreover, it has high computing complexity due to utilizing all paths between two nodes. Inspired by Jaccard function, we propose a new similarity function, $\alpha - path$ similarity (Sim), as shown in (4), which is a variation of Jaccard similarity and uses path length threshold α to control the computing complexity and effectively distinguish the influence of the degree difference between two nodes on node similarity.

$$Sim(u,v) = \frac{s(u,v)}{\sqrt{\sum_{x \in Ng(u)} s(u,x) * \sum_{y \in Ng(v)} s(v,y)}} \quad (4)$$

$$s(i,j) = \sum_{|p|=1}^{\alpha} \frac{(A^{|p|})_{ij}}{|p|} \quad (5)$$

Where, p denotes one of the paths connecting node i and j directly or indirectly. $|p|$ denotes the length of p , which varies between 1 and α . $A^{|p|}$ denotes the measure of p .

For unweighted network, $A^{|p|}=1$.

For weighted network, $A^{|p|} = \frac{\sum_{x,y \in p \cap e_{xy} \in E} w_{xy}}{|p|}$.

Definition 3. *Neighbor Node Influence (NNI): The influence of neighbor node v on node u is defined as:*

$$NNI_v(u) = \sqrt{NI(v) * \frac{Sim(v,u)}{\max_{h \in Ng(u)} Sim(h,u)}} \quad (6)$$

For label propagation, we think that the importance of different neighbor nodes are not the same. Instead, we combine NI and Sim between nodes to calculate NNI , and use NNI to measure the different influence of neighbor nodes on the nodes that need to update labels, so that the node labels can be updated more reasonably. The neighbor nodes with bigger NI and Sim will have greater influence on the node needs to update labels.

For example, for node 2 in Fig. 1, according to (6), the influences of its three neighbor nodes are separately $NNI_3(2)=0.43$, $NNI_5(2)=0.55$, $NNI_1(2)=0.51$. Obviously, node 5 has bigger influence on node 2.

3.2 Label Update Strategy

To detect overlapping communities, adopting the idea of COPRA, we use $b(c,u)$ to define the belonging coefficient of node u to community c , that is, the probability that node u belongs to community c . When the iterations for label propagation is finished, each node $u \in V$ will contain a label set L_u , thus overlapping communities are detected.

In order to decrease the complexity of label propagation, like DLPA, only the dominant label with maximum belonging coefficient is propagated in our strategy. Meanwhile, in order to decrease the instability of community detection caused by updating node labels in random order, we sort the nodes in ascending order of node importance, and the node labels are also updated using this ascending order. We think that the bigger the node importance, the bigger the probability that the node is the potential community center. So sorting node in ascending order can make the labels of potential community centers be propagated far enough to occupy the community centers and then the community can plunder the nodes in the boundary area with other communities, thus more accurate community structures can be detected.

For each node in the updating node sequence, we first update its labels and corresponding belonging coefficients, then identify its dominant label according to NNI and historical label preferred strategy.

The detailed label update strategy is defined as follows:

1. Sorting nodes $u \in V$ in ascending order according to their NI into a node sequence $vQueue$.
2. For node $u \in vQueue$, when updating its labels, it receives multiple dominant labels from its neighbor nodes and forms label set L_{Ng}:

$$L_{Ng} = \{l(c_1,b_1), l(c_2,b_2), \ldots l(c_v,b_v)\}, v \in Ng(u) \quad (7)$$

Where $l(c_v,b_v)$ denotes the dominant label of neighbor node v , b_v denotes the belonging coefficient of node v to community c_v .

3. Based on NNI and L_{Ng}, calculate the new belonging coefficient $b^{'}(c,u)$ of node u to community c :

$$b^{'}(c,u) = \frac{\sum_{l(c_v,b_v) \in L_{Ng}, v \in Ng(u), c_v = c} b(c_v,v) * NNI_v(u)}{\sum_{l(c_v,b_v) \in L_{Ng}, v \in Ng(u)} b(c_v,v) * NNI_v(u)} \quad (8)$$

Then the label set $L^{'}$ of node u is generated:

$$L^{'} = \{l(c_1,b_1^{'}), l(c_2,b_2^{'}), \ldots l(c_{|L^{'}|}, b_{|L^{'}|}^{'})\}, \sum_{l(c,b') \in L^{'}} b^{'}(c,u) = 1 \quad (9)$$

Where $|L^{'}|$ is the number of the labels in $L^{'}$.

4. Adaptively delete the useless labels which meet $b^{'}(c,u) < 1/|L^{'}|$, the remaining labels form label set $L^{''}$.
5. Normalize the belonging coefficient of the labels in $L^{''}$:

$$b''_{(c,u)} = \frac{b'_{(c,u)}}{\sum_{l(c,b')\in L''} b'_{(c,u)}}, \quad \sum_{l(c,b'')\in L''} b''_{(c,u)} = 1 \tag{10}$$

Then the normalized L'' is considered as the final label set L_u cached in the updating node u.

6. Identify the label with maximum belonging coefficient in L_u as the dominant label of node u.

Here we introduce a historical label preferred strategy which can further decrease the randomness of label propagation and increase the stability of community detection. That is, if there are multiple labels with maximum belonging coefficients and one of which is the dominant label in the previous iteration, then this dominant label is selected as the dominant label of current iteration, otherwise randomly select one as the dominant label.

4. LPANNI Algorithm

4.1 Algorithm Description

Based on the terms and label update strategy defined in Section 3, we propose a new overlapping community detection algorithm based on label propagation, LPANNI. It enhances the accuracy and stability of label propagation algorithms from two aspects. First, it sorts nodes in ascending sequence according to their *NI* and updates node labels according to this sequence rather than random sequence. Second, we think that different neighbor nodes have different influences on the updating nodes, and the influence is not only related to the importance of neighbor nodes but also to the similarities between the updating node and its neighbor nodes. The neighbor nodes with larger *NI* and *Sim* have larger influence on the updating nodes. When the iteration process of label propagation in LPANNI is completed, the nodes with multiple labels are considered as overlapping nodes and overlapping community structures are detected. By sorting nodes according to *NI* and updating node labels according to *NNI*, LPANNI can observably improve the accuracy and stability of detecting overlapping community structures.

LPANNI consists of two phases. In the first phase, it computes the *NI* of all nodes and sorts the nodes in ascending order according to *NI*. In the second phase, it propagates labels by considering both the sequence of *NI* and *NNI* until the algorithm converges, then the overlapping community structure is detected. The common convergence condition is that all the node labels don't change any more. However, this condition may not be met in some networks and the algorithm cannot finish even after hundreds of iterations. Therefore, the convergence condition used in LPANNI is that both the size of label set and the dominant labels of all nodes are stable, or a specified maximum iteration number is reached.

The complete LPANNI is described in Algorithm 1.

For the network in Fig. 1, we first compute *NI*, *Sim* and *NNI*, then sort the nodes in ascending order by their *NI*.

Table 1-A shows the *NI* of all nodes and we can get the ascending order 2→3→4→6→7→8→5→9→1, where for the nodes with same *NI*, we sort them by their node IDs. The ascending order is used for propagating labels.

Table 1-B and 1-C separately show the *Sim* matrix and the *NNI* matrix, where we set the path length threshold $\alpha = 3$. For node 2, the neighbor node influence of its neighbor nodes are separately $NNI_1(2)$=0.51, $NNI_3(2)$=0.43, $NNI_5(2)$=0.55. Obviously, node 5 has the biggest influence on node 2, then node 1 and node 3 are followed. That is, the descending order of node 2's neighbor nodes sorted by *NNI* is 5→1→3. While Table 1-A shows that, the descending order of node 2's neighbor nodes sorted by *NI* is 1→5→3. Table 1-B shows that, the similarities between nodes 1, 5, 3 and node 2 are separately $Sim_1(2)$=0.21, $Sim_5(2)$=0.32, $Sim_3(2)$=0.30, and the descending order is 5→3→1. Thus it can be seen that, *NNI* is related with both *NI* and *Sim*, combining both of which to calculate *NNI* can better measure the different influence of neighbor nodes.

Fig. 2 shows the label propagation of LPANNI in the

Algorithm 1: LPANNI

Input: Network $G=(V,E,w)$, Maximum iteration number T
Output: Label set L_u of node $u\,(u \in V)$

```
/* Phase1: calculate $NI$ and $NNI$, sort nodes */
for all nodes $u \in V$:
    $NI(u) \leftarrow$ Calculate node importance according to (1)
    $Sim(u) \leftarrow$ Calculate node similarity according to (4)
    $NNI(u) \leftarrow$ Calculate neighbor node influence according to (6)
end for
$vQueue \leftarrow$ Sorting all nodes in ascending order
    according to node importance

/* Phase2: label propagation */
$t = 0$
for all nodes $u \in V$
    $L_u \leftarrow \{u,1\}$, $dominant_u = u$
end for
while $t < T$
    for node $u \in vQueue$
        $L_{Ng} \leftarrow \{l(c_1,b_1), l(c_2,b_2), \ldots l(c_v,b_v)\}, v \in Ng(u)$
        $L' \leftarrow$ Update label set of node $u$ according to (8) and (9)
        for label $l$ in $L'$:
            if $b' < 1/|L'|$ then delete $l$
            end if
        end for
        $L'' \leftarrow$ Normalize the remaining labels in $L'$
        according to (10)
        $l_d \leftarrow$ Identify dominant label of node $u$
        according to historical label preference strategy
        $L_u \leftarrow L''$
        $dominant_u \leftarrow l_d$
    end for
    if the size of label set and the dominant labels
    of all nodes don't change
    then break
    end if
    $t \leftarrow t+1$
end while
Output label set $L_u$ of node $u$, and get community structure
```

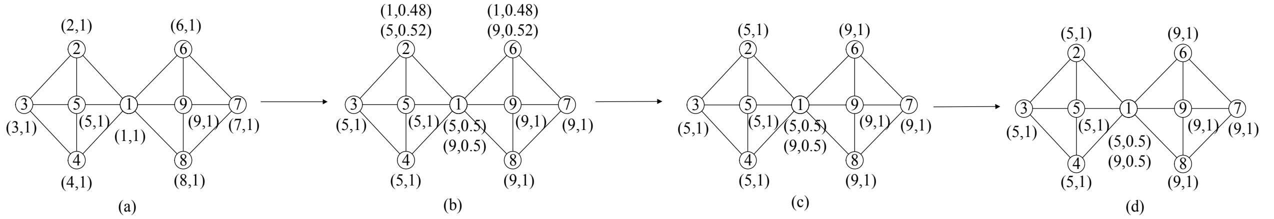


Fig.2. Label propagation process of LPANNI on the sample network (in ascending order of *NI* $2\rightarrow3\rightarrow4\rightarrow6\rightarrow7\rightarrow8\rightarrow5\rightarrow9\rightarrow1$)

sample network.

First, initializing node label use each node's ID, and setting all the belonging coefficients to 1, as shown in Fig. 2(a). Second, calculating NI and sorting the nodes in ascending order by NI , then we can get the nodes' updating sequence $2\rightarrow3\rightarrow4\rightarrow6\rightarrow7\rightarrow8\rightarrow5\rightarrow9\rightarrow1$ in ascending order. Finally, iteratively updating the labels of all nodes, deleting the useless labels with too small belonging coefficients, and updating the dominant labels.

Taking node 2 as an example, the labels of its neighbor nodes 1, 3, 5 are separately (1, 1), (3, 1), (5, 1). According to (6), we can calculate the NI of its neighbor nodes. Then, according to (8), we calculate the new belonging coefficient of each label and get the new label set {(1, 0.34), (3, 0.29), (5, 0.37)}. As the belonging coefficient of label (3, 0.29) meets $0.29 < 1/3$, which is considered as a useless label and is deleted. Then, we normalize the remaining labels according to (10). At last, after finishing updating node 2, the label set of node 2 is {(1, 0.48), (5, 0.52)} and the dominant label is (5, 0.52). In the same way, we can update the labels and the corresponding belonging coefficients of other nodes.

Fig. 2(b) and Fig. 2(c) separately show the results after the first and the second iteration. After the third iteration in Fig. 2(d), the label set and dominant label of each node are stable and the algorithm stops, two overlapping communities are detected, and node 1 is an overlapping node which has equal belonging coefficient to both communities.

4.2 Time Complexity Analysis

Let n be the amount of nodes, m be the amount of edges, k be the average degree of network nodes, T be the maximum iteration number, and α be the maximum path length for measuring Sim .

In the first phase of LPANNI, the time complexity includes those for measuring three metrics and sorting nodes.

1. Time complexity for measuring *NI*. The time complexity of computing nodes' degrees and the number of triangles are $O(k)$ and $O(k^2)$ respectively, the sum is $O(kn+k^2n)$.

2. Time complexity for measuring *Sim*. The time complexity for calculating the similarity between one pair of nodes is $O(k^{\alpha-1})$, thus for m pairs of nodes, it is $O(k^{\alpha-1}*m)$.

3. Time complexity for measuring *NNI*. According to (6), *NNI* is linear to *NI* and *Sim*, so the time complexity for measuring the *NNI* of m neighbor nodes is $O(m)$.

4. Time complexity for sorting nodes in ascending order of NI . We use an existing sorting algorithm to do that, such as radix or bucket sort, so the time complexity is $O(n)$.

In the second phase of LPANNI, the time complexity is determined by the time for iteratively updating nodes' labels. For each iteration, it is $O(kn)$, so the time complexity for T iterations is $O(T*kn)$.

TABLE 1-A
NI OF THE SAMPLE NETWORK

Node ID	1	2	3	4	5	6	7	8	9
NI	1.0	0.5	0.5	0.5	0.8	0.5	0.5	0.5	0.8

TABLE 1-B
NODE SIMILARITY MATRIX OF THE SAMPLE NETWORK

Node ID	1	2	3	4	5	6	7	8	9
1	0	0.21	0	0.21	0.22	0.21	0	0.21	0.22
2	0.21	0	0.30	0	0.32	0	0	0	0
3	0	0.30	0	0.30	0.32	0	0	0	0
4	0.21	0	0.30	0	0.32	0	0	0	0
5	0.22	0.32	0.32	0.32	0	0	0	0	0
6	0.21	0	0	0	0	0	0.30	0	0.32
7	0	0	0	0	0	0.30	0	0.30	0.32
8	0.21	0	0	0	0	0	0.30	0	0.32
9	0.22	0	0	0	0	0.32	0.32	0.32	0

TABLE 1-C
NNI MATRIX OF THE SAMPLE NETWORK

Node ID	1	2	3	4	5	6	7	8	9
1	0	0.51	0	0.51	0.44	0.51	0	0.51	0.44
2	0.29	0	0.40	0	0.37	0	0	0	0
3	0	0.43	0	0.43	0.37	0	0	0	0
4	0.29	0	0.40	0	0.37	0	0	0	0
5	0.37	0.55	0.53	0.55	0	0	0	0	0
6	0.29	0	0	0	0	0	0.40	0	0.37
7	0	0	0	0	0	0.43	0	0.43	0.37
8	0.29	0	0	0	0	0	0.40	0	0.37
9	0.37	0	0	0	0	0.32	0.53	0.55	0

Thus, the overall time complexity of LPANNI is $O((k^{\alpha-1}+1)m+(k^2+(T+1)k+1)n)$. For large-scale networks, $k,\alpha,T \ll \mathrm{m},n$, so $O((k^{\alpha-1}+1)m+(k^2+(T+1)k+1)n) \approx O(hm+kn)$, where h,k are constants and $h,k \ll m,n$. So we can get the conclusion that the time complexity of LPANNI is linear complexity $O(n)$, especially for sparse large-scale networks.

5. EXPERIMENTS AND RESULTS

5.1 Experimental Schemes

We evaluate the effects of LPANNI in both synthetic and real networks. All the experiments are conducted on a PC with 2.67 GHz Intel Core i5 CPU and 6 GB RAM.

5.1.1 Baseline methods

Xie et al. [22] summarized the existing overlapping community detection algorithms, proposed a framework to evaluate the ability of those algorithms for detecting overlapping nodes, and comprehensively compared the effects of 14 algorithms in terms of NMI, Omega, F-score, etc. They came out the following conclusions: for low overlapping density networks, SLPA [17], Game [23], OSLOM [6] and COPRA [16] offer better performance than the other tested algorithms; for high overlapping density networks, both SLPA and Game provide better performance. So in this paper, we choose several algorithms that perform better, COPRA, SLPA and Game, as the baseline methods.

Meanwhile, as analyzed in Section 2, considering the advantages of DLPA [18] and WLPA [21] in improving the accuracy of overlapping community detection and the similarity to our algorithm, DLPA and WLPA are also selected as the baseline methods.

5.1.2 Datasets

To adequately evaluate the effects of LPANNI, we use two kinds of datasets. One is four real networks (http://snap.stanford.edu/data/, http://www.personal.umich.edu/~mejn/netdata/) as shown in Table 2. Where n, m and k respectively denote the amount of nodes, the amount of edges, and the average degree of network nodes. However for many real networks, the true community structures are usually unknown, which leads the inability to evaluate the real effects of LPANNI. So we also experiment the algorithms on six different kinds of synthetic networks, which are generated by LFR benchmark networks [24] and their real community structures are known, as shown in Table 3. Some necessary parameters are described in Table 4.

By changing network size n , we can evaluate the effects of LPANNI on different scale networks. By changing parameters mu , om and on to adjust the fuzzy degree of community structures, we can evaluate the effects of LPANNI for detecting the community structures of networks with different fuzzy degree.

5.1.3 Experimental Parameter Settings

In order to obtain the best community results, the parameters of the baseline algorithms with tunable parameters, including COPRA, SLPA and DLPA, need to be adjusted for different dataset. So in following experiments, the optimal parameter values are determined by experiments and the corresponding best results of these algorithms are compared with the results of our algorithm in this paper.

Specifically, we conduct experiments on the networks in Table 2 and Table 3 for each of the above algorithms using the following parameter settings:

1. For COPRA, maximum label number v of each node is taken from range [1, 10].
2. For SLPA, label probability threshold r varies from 0.01 to 0.1 with an interval 0.01 in synthetic networks, while r varies from 0.05 to 0.5 with an interval 0.05 in real networks.
3. For DLPA, inflation parameter in ranges from 1 to 8.

TABLE 2
FOUR REAL NETWORKS

Network Name	N	m	k
Facebook (FB)	4039	88234	43.69
Ca-HepPh (CH)	12006	118489	19.74
Email-Enron (EE)	36692	183831	10.02
Com-Amazon (CA)	334863	925872	5.53

TABLE 3
SIX KINDS OF SYNTHETIC NETWORKS

Network/ Parameters	N	k	max$k$	mu	min$c$	max$c$	on	om
N-1000-mu0.1	1000	10	50	0.1	10	50	100	[2,8]
N-1000-mu0.3	1000	10	50	0.3	10	50	100	[2,8]
N-5000-mu0.1	5000	10	50	0.1	20	100	500	[2,8]
N-5000-mu0.3	5000	10	50	0.3	20	100	500	[2,8]
N-10000-mu0.1	10000	20	100	0.1	20	100	2000	[2,8]
N-10000-mu0.3	10000	20	100	0.3	20	100	2000	[2,8]

TABLE 4
MEANINGS OF PARAMETERS IN SYNTHETIC NETWORKS

Parameter	**Meaning**
n	node number
k	average degree
$\max k$	biggest degree
mu	mixing coefficient (connection probability that
$\min c$	minimum number of nodes in one community
$\max c$	maximum number of nodes in one community
on	the number of overlapping nodes
om	the number of communities each overlapping

TABLE 5
OPTIMAL VALUES OF TUNABLE PARAMETERS OF BASELINE ALGORITHMS IN FOUR REAL NETWORKS

Network	SLPA	COPRA	DLPA
Facebook	r=0.2	v=4	in=8
Ca-HepPh	r=0.5	v=2	in=6
Email-Enron	r=0.45	v=2	in=8
Com-Amazon	r=0.45	v=10	in=5

TABLE 6
OPTIMAL VALUES OF TUNABLE PARAMETERS OF BASELINE ALGORITHMS IN SYNTHETIC NETWORKS

Network	SLPA	COPRA	DLPA
N-1000-mu0.1	r=0.06	v=6	in=2
N-1000-mu0.3	r=0.06	v=6	in=2
N-5000-mu0.1	r=0.05	v=6	in=2
N-5000-mu0.3	r=0.1	v=6	in=2
N-10000-mu0.1	r=0.06	v=6	in=2
N-10000-mu0.3	r=0.1	v=6	in=2

The maximum iteration number T for COPRA, SLPA, DLPA and LPANNI is set to 100. Meanwhile, given that the instability of these algorithms, we separately repeat the experiments for 50 times under each parameter settings and get the average results in term of NMI_max. The determined optimal parameter settings as shown in Table 5 and Table 6.

5.1.4 Evaluation Criteria

For real networks, since the true community structures are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2866424, IEEE Transactions on Knowledge and Data Engineering

generally unknown, we use overlapping modularity Q_{ov} [25] to evaluate the effects of community detection algorithms. Commonly, larger Q_{ov} means better community results. The value of Q_{ov} depends on the number of communities to which each node belongs and the strength of the membership to each community. In order to compare with the comparision algorithms based on the same conditions, we assume that each node has equal strength to all communities it belongs to.

However, for some networks, bigger Q_{ov} may not mean better network partitions [18]. In addition, in case that an algorithm detects many small communities, although the result is similar to the actual situation, the modularity of the detection results is not good due to the resolution limit of modularity. Hence, only using Q_{ov} may not fully evaluate the effects of LPANNI.

So for synthetic networks with known community results, we use NMI [5], NMI_max [26] and Omega [27] to evaluate the effects of the algorithms. All the three metrics can be used to measure the similarity between the experimental results and the ground truth. Their values vary between 0~1 and bigger value means better community detection results.

Both NMI and NMI_max are based on information theory, and NMI was used in many literatures to evaluate the results of community detection. However, NMI may overestimate the similarity of two partitions in some situations. For example, X, Y are two partitions, X contains many communities and Y contains only one community, and the community in Y and one of the communities in X are exactly the same. In this case, the score of NMI is 0.5, while the score of NMI_max is very low. Obviously, NMI_max can reflect the truth more accurately. Omega measures the similarity in a different perspective. It focuses on the number of communities in which a pair of nodes appeared. Whereas there is no clear-cut criterion to determine which metric is best, Fortunato et al. [28] recommended to use criteria based on information theory. So in this paper, we choose NMI_max as the main criterion and the other two as complementary criteria.

It is also important to detect the true overlapping nodes for overlapping community detection algorithms. Like SLPA, we take the identification of overlapping nodes as a binary classification problem, and use $F-score$ to quantify the accuracy of community detection, which is defined in (11):

$$F-score=(2*Precision*Recall)\ /\ (Precision+Recall) \quad (11)$$

Where *Precision* denotes the ratio of the correctly detected overlapping nodes in the total detected overlapping nodes, *Recall* denotes the ratio of the correctly detected overlapping nodes in the true overlapping nodes in the network.

5.2 Selection of Parameter α in LPANNI

In LPANNI, parameter α is used to control the maximum path length for measuring Sim. There is a tradeoff between the accuracy of Sim and time complexity. Bigger α means more topology information is considered to measure the similarities between nodes, and the measure may be more accuracy, however the complexity of the similarity measure is higher. So we first conduct an experiment to check the influence of different α on community detection and determine an appropriate α which leads to better accuracy and lower time complexity.

Fig. 3 shows the influence of α on NMI_max in six types of synthetic networks in Table 3. It can be seen that for different om, with the increasing of α, NMI_max first rises to the top at $\alpha=3$ or $\alpha=4$ followed by small fall on the whole. It shows that, considering more topology information between nodes indeed increase the accuracy of LPANNI. However, considering too long paths may damage the accuracy and largely increase time complexity. Fig. 3 also shows that different networks have different optimal α, while for most networks, the optimal value is 3 or 4. Considering time complexity, in following experiments, we simply set $\alpha=3$.

5.3 Accuracy and Stability in Real Networks

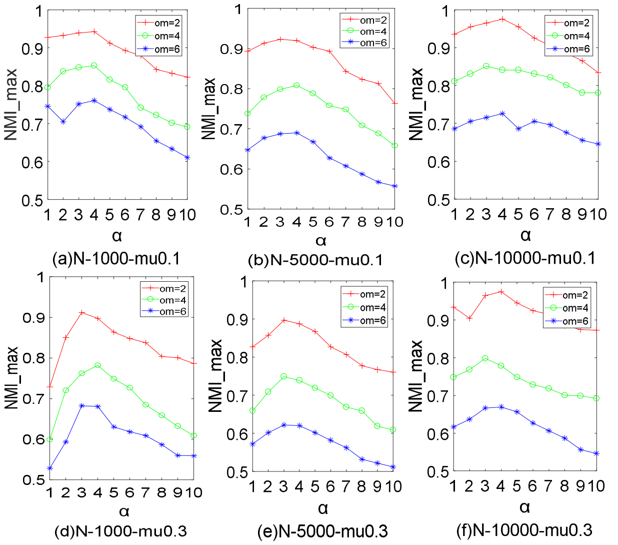
We compared the community detection results of LPANNI with five baseline methods, COPRA, SLPA, DLPA, WLPA, and Game, in four real networks shown in Table 2.

For algorithms based on label propagation, we compare the average overlapping modularity Q_{ov}*_avg* and the standard deviation Q_{ov}*_std* of 50 repetitions. For algorithms not based on label propagation, given that their results are stable, we conduct their experiments for only once, and then compare the value of Q_{ov}. However, due to the high time and space complexity of Game, we don’t evaluate its performance on large-scale networks.

The experimental results under the corresponding optimal parameter settings (see Table 5) are shown in Table 7, and the optimal Q_{ov} are indicated in bold type.

It can be seen that, as for accuracy, LPANNI obtains the best Q_{ov}*_avg* in the largest network Com_Amazon. Although LPANNI obtains suboptimal Q_{ov}*_avg* in networks Ca-HepPh and Email_Enron, the difference with the optimal value is very small. Among the baseline algorithms, WLPA and SLPA perform well in most of the middle-scale networks, COPRA performs well in Facebook network, and DLPA performs poorly in large-scale networks.

As for stability, LPANNI obtains the smallest Q_{ov}*_std* in most of the real networks, which indicates that the results of LPANNI are very stable. Among the baseline methods,



(a)N-1000-mu0.1 (b)N-5000-mu0.1 (c)N-10000-mu0.1

(d)N-1000-mu0.3 (e)N-5000-mu0.3 (f)N-10000-mu0.3

Fig. 3. Influence of α on NMI_max in Synthetic Networks

TABLE 7
ACCURACY AND STABILITY OF DIFFERENT ALGORITHMS IN FOUR REAL NETWORKS

Networks		Facebook	Ca_HepP	Email_Enron	Com_Amazon
LPANNI	Q_{ov}_avg	0.9088	0.7762	0.7378	**0.8001**
	Q_{ov}_std	0.0032	0.0023	0.0012	0.0004
SLPA	Q_{ov}_avg	0.9296	**0.8102**	0.7302	0.7641
	Q_{ov}_std	0.0112	0.0109	0.0454	0.0006
	r	0.20	0.50	0.45	0.45
COPRA	Q_{ov}_avg	**0.9322**	0.7642	0.6991	0.7898
	Q_{ov}_std	0.0051	0.0040	0.0313	0.0344
	v	4	2	2	10
DLPA	Q_{ov}_avg	0.8835	0.6868	0.7268	0.6626
	Q_{ov}_std	0.0065	0.0155	0.0127	0.0004
	in	8	6	8	5
WLPA	Q_{ov}_avg	0.9270	0.7681	**0.7393**	0.7604
	Q_{ov}_std	0.0024	0.0033	0.0014	0.0002

WLPA is second only to LPANNI, while SLPA, COPRA and DLPA are poor stability in some networks.

So it can be concluded that in terms of Q_{ov}_*avg* and Q_{ov}_*std*, LPANNI and WLPA have higher accuracy and the highest stability in all the four real networks.

5.4 Accuracy and Stability in Synthetic Networks

5.4.1 Accuracy and Stability of Detecting Overlapping Community

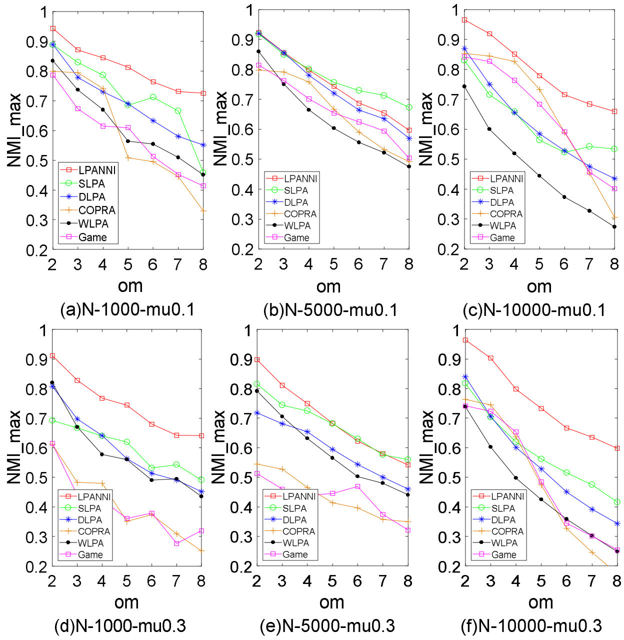
We conduct experiments on six kinds of synthetic networks (Table 3) with different parameters to evaluate the accuracy of LPANNI. For the algorithms with tunable parameters, their optimal values are set according to Table 6.

Fig. 4 shows the NMI_max of different algorithms in the six types of synthetic networks with different *om* . The NMI and Omega of different algorithms on networks with 10,000 nodes are also shown in Fig. 5.

As shown in Fig. 4, LPANNI achieves the highest average NMI_max in most networks, especially performs better than the baseline methods in networks with $mu = 0.3$.

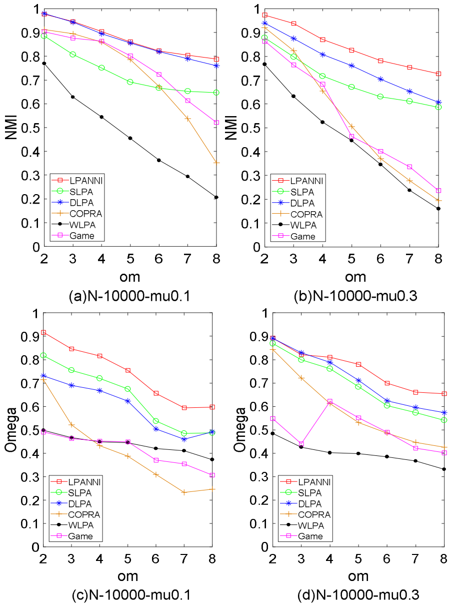
With the increasing of *om*, the overlapping communities in the networks become more intricate, so the NMI_max of all algorithms decline gradually. However, compared with the baseline algorithms, LPANNI shows better performance with keeping up $NMI\_max \geq 0.7$, which is obviously larger than others. Particularly, when fixing network scale and increasing *mu* from 0.1 to 0.3, its NMI_max only slightly decreases, whereas that of COPRA, DLPA and SLPA declines more obviously. This means that LPANNI can detect overlapping communities even in the networks with fuzzy community structures.

As the network scale becomes bigger, the overall NMI_max of LPANNI does not decline obviously, which shows that LPANNI is not sensitive to network scale. It is worth noting that, in networks with 10,000 nodes, the difference between LPANNI and the baseline algorithms is significant. While Game performs slightly better on large-scale networks and worse on middle-scale networks. WLPA performs poorly in synthetic networks although it performs well in real networks.



(a)N-1000-mu0.1 (b)N-5000-mu0.1 (c)N-10000-mu0.1
(d)N-1000-mu0.3 (e)N-5000-mu0.3 (f)N-10000-mu0.3

Fig. 4. NMI_max on six types of synthetic networks with different *om*



(a)N-10000-mu0.1 (b)N-10000-mu0.3
(c)N-10000-mu0.1 (d)N-10000-mu0.3

Fig. 5. NMI and Omega on synthetic networks with10,000 nodes

In conclusion, in terms of NMI_max, LPANNI is appropriate for networks with different scale and different complexities of overlapping community structures, which performs much better than the baseline algorithms in large-scale networks with fuzzy community structures.

Fig. 5 shows that, LPANNI, DLPA and SLPA also perform well in terms of NMI and Omega. However, LPANNI is the best, which is consistent with NMI_Max.

In addition, we compare the NMI_max distribution of different label propagation algorithms to evaluate their stability. For each algorithm, as mentioned before, we repeat 50 times of experiments and get 50 NMI_max. The NMI_max distribution in the six types of synthetic networks are shown in Fig. 6.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2866424, IEEE Transactions on Knowledge and Data Engineering

From Fig. 6, we can see that most of the algorithms behave similarly on networks with same scale but different mixing coefficients, while in networks with same mixing coefficient and different scales, the difference is bigger. It means that network scale has major influence on the stability of the algorithms. In small-scale networks, the label propagation converges rapidly because the network structure is relatively simple, therefore the results of the unstable algorithms for each experiment are quite difference. However, the NMI_max of LPANNI and WLPA fluctuates very little in all the synthetic networks. This illustrates that the stability of LPANNI and WLPA is very good, and stable results can be obtained in networks with different scales and different complexity. Moreover, among the other three label propagation algorithms, DLPA comes second, while COPRA and SLPA have very poor stability.

5.4.2 Accuracy of Detecting Overlapping Nodes

Besides evaluating the accuracy and stability of LPANNI through NMI_max, NMI and Omega, we also compare the accuracy of detecting overlapping nodes using Precision, Recall and F-score. For the algorithms with tunable parameters, the parameter values are set according to Table 6.

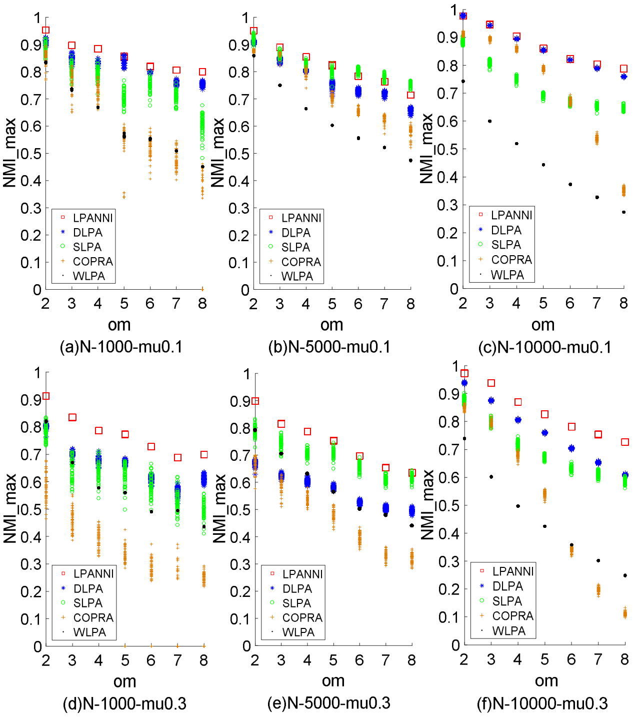
Fig. 7-9 show the F-score, Precision and Recall of different algorithms in the synthetic networks. It can be seen from Fig.7, in most cases, the F-score of LPANNI is the best, and with the increasing of om, its F-score in small and medium scale networks increases slowly, while in large-scale networks, it stays steady on a very high value, even in networks with fuzzier community structures ($mu = 0.3$).

As we know, larger mixing parameter mu or the number of memberships om makes the community structures fuzzier, which may lead to detecting overlapping communities becomes more difficulty. So from the results of Fig.7, we can conclude that LPANNI can identify overlapping nodes more accurately, and can reach high accuracy even in large-scale networks with fuzzy community structures.

As for the baseline algorithms, such as DLPA and SLPA, although they may have high F-score, especially for DLPA, its F-score is closer to LPANNI in networks with 10,000 nodes, their F-score drops obviously when mu increases on networks with 1,000 and 5,000 nodes, which indicates that they are sensitive to community structures and cannot accurately detect overlapping nodes in networks with fuzzy community structures.

It is worth noting that the F-score of WLPA is very low, so we can infer that it cannot detect overlapping nodes well. As analyzed in Section 2, WLPA only considers the degree of centrality to measure the node importance, which may affect the results of overlapping community detection. So the reason that WLPA gets "good" results in the real networks may be that the overlapping ratio among communities in the real networks is not very high. While for synthetic networks with certain overlapping ratio, the community structures are very different from those in the real networks, so WLPA performs poorly.

The Precision and Recall of each algorithm in different synthetic networks are also shown in Fig. 8 and Fig. 9, which evaluate the ability of the algorithms to detect overlapping nodes more intuitively. It can be seen that, some algorithms have imbalanced Precision and Recall. There are usually two reasons: over-detection and under-detection. For example, COPRA has high Recall but low Precision, which indicates it has the problem of over-detection. The Precision and Recall of Game is relatively balanced, but neither of the two metrics is high, so its F-score is not high. Whereas the Precision and Recall of LPANNI are high and balance, the larger the scale of networks, the more obvious this feature. This further proves that LPANNI has obvious advantages in detecting overlapping nodes.



(a)N-1000-mu0.1 (b)N-5000-mu0.1 (c)N-10000-mu0.1

(d)N-1000-mu0.3 (e)N-5000-mu0.3 (f)N-10000-mu0.3

Fig. 6. NMI_max distribution on synthetic networks

5.4.3 Comprehensive Ranking of Different Algorithms

To intuitively compare the comprehensive performance of LPANNI and the comparison schemes in terms of multiple criteria, based on the results above, a comprehensive ranking of different algorithms are shown in Table 8.

The ranking in certain network is evaluated as:

$$RS_M(i) = \sum_{j=2}^{8} rank(i,\ O_m^j) \qquad (11)$$

Where, $RS_M(i)$ denotes the ranking of algorithm i in terms of criterion M , $rank(i, O_m^j)$ denotes the ranking of algorithm i in the network with $om = j$.

Similarly, for networks with certain scale and different mu , the ranking is calculated by considering all the rankings in case of different mu . For example, $RS_{NMImax}, N10000*$ in Table 8 is the comprehensive ranking of $NMImax$ in networks with 10,000 nodes, including mu=0.1 and mu=0.3 .

For networks with different scale and different mu , the ranking is calculated by considering all the rankings in case of different mu and different number of nodes. For example, $RS_{NMImax*}$ in Table 8 denotes the comprehensive ranking of $NMImax$ in the networks with 1,000, 5,000 and 10,000 nodes, and the cases of mu=0.1 and mu=0.3 .

Finally, $RS_{NMImax+NMI+Omega}$ denotes the overall ranking in terms of all criteria. It is obviously, compared with the comparison schemes, LPANNI performs best.

5.5 Time Complexity of LPANNI

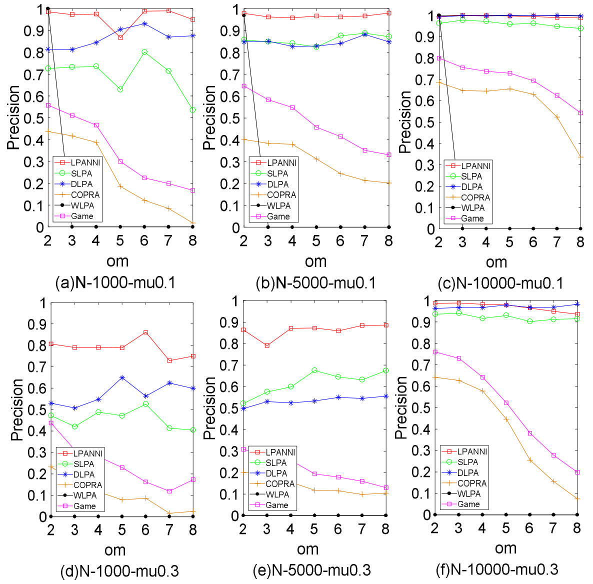
As analyzed in Section 4.2, the time complexity of LPANNI is $O(hm+kn)$, so like COPRA, SLPA and DLPA, for large-scale networks, the time complexity of LPANNI is also linear, especially for sparse large-scale networks.

We compare the time cost of five algorithms, LPANNI, COPRA, SLPA, WLPA and DLPA, in different scale synthetic networks. The network parameters are set as follows: $k=10$, $\max k=50$, $\min c=10$, $\max c=50$, $mu=0.1$, $on/n=0.1$, $om=3$, the node number increases from 1,000 to 100,000.

Fig.10 shows the time cost of the five different algorithms in different scale networks. It can be observed that all the five algorithms have linear time complexity and the difference is the slope. In order to increase the accuracy of node similarity measure, LPANNI considers more local topology information, which slightly increases the time cost. However, as shown in the previous experiments, LPANNI gets better accuracy and best stability compared with the baseline methods, which indicates that LPANNI significantly improve the performance of label propagation algorithms on detecting overlapping community structures.

5.6 Discussion of the Improvement of Node Importance and Label Update Strategy on the Effects of Community Detection

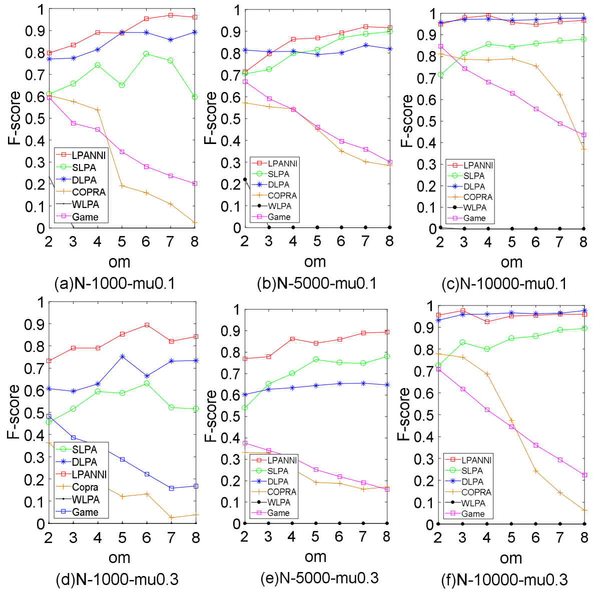
In general, LPANNI improves the label propagation algorithms from two aspects: 1) It defines three metrics, *NI*, *Sim*



(a)N-1000-mu0.1 (b)N-5000-mu0.1 (c)N-10000-mu0.1

(d)N-1000-mu0.3 (e)N-5000-mu0.3 (f)N-10000-mu0.3

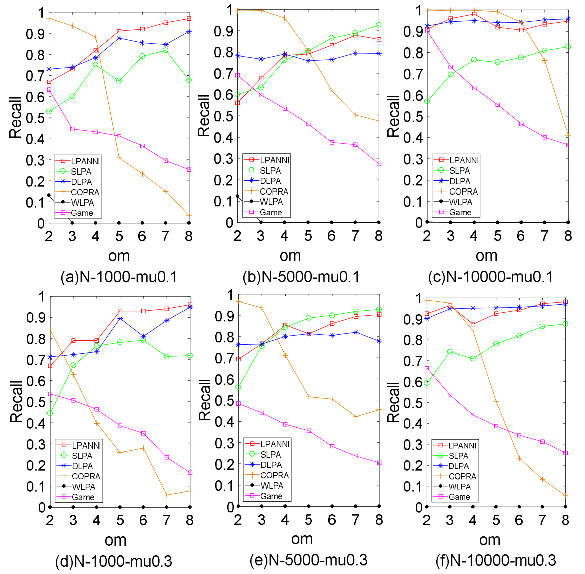
Fig. 8. Precision of different algorithms in six synthetic networks



(a)N-1000-mu0.1 (b)N-5000-mu0.1 (c)N-10000-mu0.1

(d)N-1000-mu0.3 (e)N-5000-mu0.3 (f)N-10000-mu0.3

Fig. 7. F-score of different algorithms in six types of synthetic networks



(a)N-1000-mu0.1 (b)N-5000-mu0.1 (c)N-10000-mu0.1

(d)N-1000-mu0.3 (e)N-5000-mu0.3 (f)N-10000-mu0.3

Fig. 9. Recall of different algorithms in six types of synthetic networks

TABLE 8
COMPREHENSIVE RANKING OF DIFFERENT ALGORITHMS IN SYNTHETIC NETWORKS

Rank	RS_{NMImax} *N10000**	$RS_{NMImax*}$	RS_{NMI} *N10000**	RS_{Omega} *N10000**	$RS_{NMImax+NMI+Omega}$ *N10000**	$RS_{F\text{-}Score}$ *N10000**	$RS_{F\text{-}Score*}$	$RS_{Stability}$ *N10000**	$RS_{Stability*}$
1	LPANNI	LPANNI	LPANNI	LPANNI	LPANNI	DLPA	LPANNI	WLPA	LPANNI
2	COPRA	SLPA	DLPA	SLPA	DLPA	LPANNI	DLPA	LPANNI	WLPA
3	Game	DLPA	SLPA	DLPA	COPRA	SLPA	SLPA	DLPA	DLPA
4	DLPA	COPRA	COPRA	COPRA	SLPA	COPRA	COPRA	COPRA	SLPA
5	SLPA	Game	Game	Game	Game	Game	Game	SLPA	COPRA
6	WLPA	WLPA	WLPA	WLPA	WLPA	WLPA	WLPA	-	-

and *NNI*; 2) It defines a new label update strategy based on *NNI*. As analyzed in Section 2, NIBLPA [19], LPA_NI [20], CK-LPA [12] and WLPA [21] also improved the accuracy and stability of LPA by measuring node importance and label influence. To verify whether the *NI* measure or the label update strategy of LPANNI has more improvement on the effects of community detection and considering the similarity between the above algorithms and LPANNI, we conduct some additional experiments.

Considering LPA_NI is similar to NIBLPA, and it needs priori importance to measure the node importance, which is not easy to get for most networks, we combine NIBLPA, CK-LPA, WLAP and LPANNI to set up the following four comparison schemes and evaluate their effects:

1. Algorithm 2: adopting the node importance of WLPA and the label update strategy of LPANNI.
2. Algorithm 3: adopting the node importance of CK-LPA and the label update strategy of LPANNI.
3. Algorithm 4: adopting the node importance of NIBLPA and the label update strategy of LPANNI.
4. Algorithm 5: adopting the node importance of LPANNI and the label update strategy of WLPA.

Given that CK-LPA and NIBLPA are for non-overlapping community detection, we don't combine their label update strategies with the node importance of LPANNI.

5.6.1 Accuracy and Stability in Real Networks

We compare the average Q_{ov} (Q_{ov}_avg) and the standard deviation of Q_{ov} (Q_{ov}_std) of the four combination algorithms and LPANNI. The results under the corresponding optimal parameters are shown in Table 9, and the optimal Q_{ov} are indicated in bold type.

Table 9 shows that, for large-scale real networks, Email-Enron and Com-Amazon, LPANNI obtains the optimal Q_{ov}_avg and the smallest Q_{ov}_std. Considering the node importance measures of the comparison schemes, only NIBLPA gets the optimal Q_{ov}_avg in Ca_HepPh, and all of the comparisons have little difference with LPANNI. While the community detection results of Algorithm 5, which uses WLPA's label update strategy, are pretty good. This means that the label update strategy of WLPA also seems good for real networks.

So we can draw the following conclusions in real networks: node importance measure has relatively small influence on the effects of community detection; label update strategy has greater influence on the results; LPANNI performs best in large-scale real networks.

5.6.2 Accuracy and Stability in Synthetic Networks

Similar to the previous experiments, we use NMI_max, NMI and Omega to evaluate the accuracy of the four combination schemes and LPANNI for detecting overlapping community in synthetic networks. Fig.11 shows NMI_max of different combinations in the six types of synthetic networks (Table 3) with different *om* . The NMI and Omega on networks with 10,000 nodes are shown in Fig. 12.

From Fig. 11 and Fig. 12, it can be seen that, the NMI_max of Algorithms 2, 3 and 4, which respectively combine the node importance of WLPA, CK-LPA, NIBLPA and the label update strategy of LPANNI, are similar in most cases, and they are similar with LPANNI. The NMI of LPANNI in large-scale networks with 10,000 nodes is slightly better than the three combinations, while the Omega of LPANNI is significantly better than the three combinations in most cases.

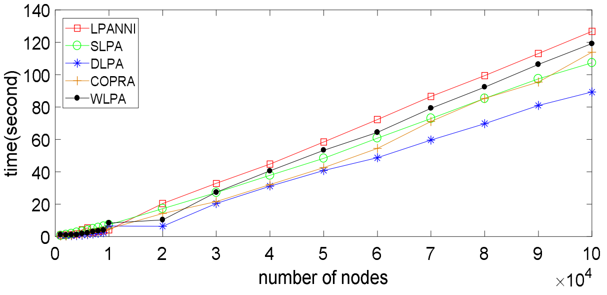


Fig. 10. Time cost of different algorithms under different network nodes

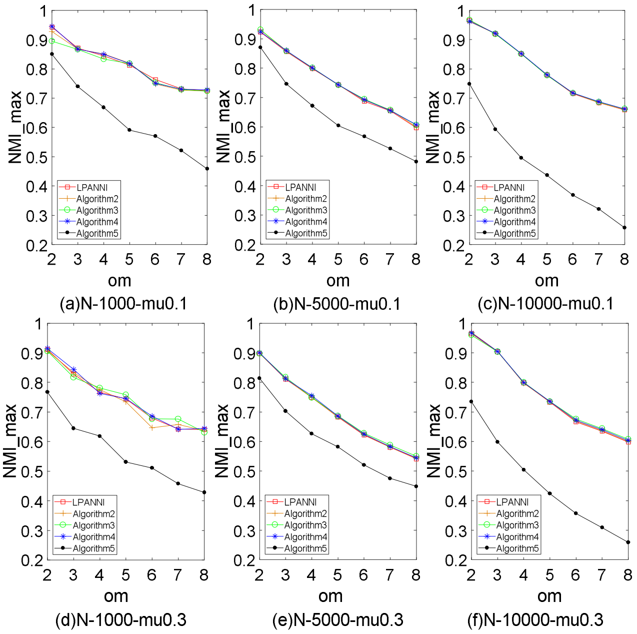
TABLE 9
ACCURACY AND STABILITY OF DIFFERENT COMBINATION SCHEMES IN FOUR REAL NETWORKS

Networks		Facebook	Ca_HepPh	Email_Enron	Com_Amazon
LPANNI	Q_{ov}_avg	0.9088	0.7762	**0.7378**	**0.8001**
	Q_{ov}_std	0.0032	0.0023	0.0012	0.0004
Algorithm 2	Q_{ov}_avg	0.9151	0.7704	0.7362	0.6693
	Q_{ov}_std	0.0056	0.0021	0.0046	0.0030
Algorithm 3	Q_{ov}_avg	0.9154	0.7465	0.6309	0.6654
	Q_{ov}_std	0.0071	0.0018	0.0597	0.0025
Algorithm 4	Q_{ov}_avg	0.9149	**0.7850**	0.7368	0.6964
	Q_{ov}_std	0.0042	0.0039	0.0018	0.0003
Algorithm 5	Q_{ov}_avg	**0.9723**	0.7798	**0.7378**	0.7608
	Q_{ov}_std	0.0024	0.0019	0.0058	0.0005

For algorithm 5, which combines the label update strategy of WLPA and the node importance of LPANNI, the results is similar to WLPA, that is, it performs poorly in terms of NMI_max, NMI and Omega in synthetic networks although it performs well in real networks. While Algorithm 2, which adopts the node importance of WLPA and the label update strategy of LPANNI, also performs well in synthetic networks. So we can infer that it may be the label update strategy of WLAP that leads to WLPA's poor ability for detecting overlapping nodes.

Actually, as analyzed in Section 2 again, in case that a node belongs to more than one community, WLPA tends to choose the community where the node with larger neighbor degree belongs, which may affect the result of overlapping community detection. So, adopting the label update strategy of WLPA, the higher the node overlapping ratio, the worse the results of overlapping community detection may be. The experimental results of the combination schemes further prove that the label update strategy based on NNI in LPANNI is much better than that in WLPA.

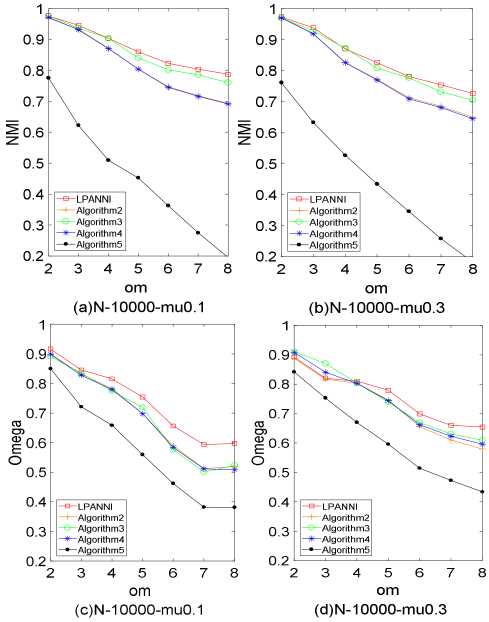
The reason why WLPA performs poorly in synthetic networks while performs well in real networks maybe that its label update strategy can hardly detect overlapping nodes, and the node overlapping ratio in the real networks is low. So the measure of label influence in WLPA may need to be further improved, while the corresponding measure of label influence in LPANNI is proved well.



(a)N-1000-mu0.1 (b)N-5000-mu0.1 (c)N-10000-mu0.1

(d)N-1000-mu0.3 (e)N-5000-mu0.3 (f)N-10000-mu0.3

Fig. 11. NMI_max on synthetic networks



(a)N-10000-mu0.1 (b)N-10000-mu0.3

(c)N-10000-mu0.1 (d)N-10000-mu0.3

Fig. 12. NMI and Omega on synthetic networks with 10,000 nodes

With respect to the stability of the above algorithms, we run each algorithm for 50 times in each experimental scenario, then calculate their variance of NMI_max. Table 10 gives the stabilities of all the combination schemes. It can be seen that all of the stabilities are very good, the difference between different combinations seems slight. However, LPANNI is most stable in most cases. So we conclude that, the stability is mainly determined by label update strategy rather than node importance measure.

TABLE 10
STABILITY (VARIANCE OF NMI_MAX) OF DIFFERENT COMBINATION SCHEMES IN SYNTHETIC NETWORKS

Algorithms	N1000 -mu0.1	N1000 -mu0.3	N5000 -mu0.1	N5000 -mu0.3	N10000 -mu0.1	N10000 -mu0.3
LPANNI	$5.44\text{x}10^{-17}$	$1.64\text{x}10^{-4}$	$1.66\text{x}10^{-16}$	$2.85\text{x}10^{-4}$	$1.37\text{x}10^{-17}$	$1.78\text{x}10^{-16}$
Algorithm2	$3.81\text{x}10^{-16}$	$5.99\text{x}10^{-4}$	$3.52\text{x}10^{-5}$	$4.26\text{x}10^{-4}$	$3.46\text{x}10^{-6}$	$4.95\text{x}10^{-4}$
Algorithm3	$2.85\text{x}10^{-16}$	$1.22\text{x}10^{-3}$	$2.04\text{x}10^{-16}$	$3.73\text{x}10^{-4}$	$2.38\text{x}10^{-16}$	$3.60\text{x}10^{-5}$
Algorithm 4	$3.00\text{x}10^{-4}$	$4.87\text{x}10^{-4}$	$2.04\text{x}10^{-16}$	$1.75\text{x}10^{-4}$	$1.84\text{x}10^{-5}$	$3.08\text{x}10^{-5}$
Algorithm 5	$2.74\text{x}10^{-4}$	$1.34\text{x}10^{-4}$	$1.20\text{x}10^{-4}$	$1.47\text{x}10^{-5}$	$2.85\text{x}10^{-16}$	$1.19\text{x}10^{-16}$

Taken the results of real networks and synthetic networks together, we can draw the following conclusions: 1) Using the node importance of LPANNI yields the best performance for most of cases; 2) The label update strategy of LPANNI is the most efficient; 3) Compared with node importance (*NI*), label update strategy has dominant influence on the improvements achieved by LPANNI.

However, it should be additionally noted that LPANNI uses node importance in a different way from other schemes, although we separate node importance from label update strategy to evaluate which part has greater influence on the performance improvements achieved by LPANNI. For most of the label update strategies in comparison methods, node importance is only used to measure the label weight of the neighbor nodes. In LPANNI, on the one hand, the update order of node labels is based on the ascending order of node importance, that is, node importance is used to determine the label update order. On the other hand, LPANNI considers both the importance of neighbor nodes and the similarities between the updating node and its neighbor nodes, and combines both as *NNI* to measure the weight of the labels coming from neighbors. So the improvement of LPANNI to the existing label propagation algorithms mainly benefits from the label update strategy, which is based on *NNI* and the label update order in the ascending order of node importance.

6. CONCLUSION

We proposed a novel overlapping community detection algorithm based on label propagation, LPANNI. It can be used to detect community structures in large-scale complex networks due to linear time complexity. Through measuring the influence of neighbor nodes using node importance and similarity between node pairs to improve the label update strategy, LPANNI significantly increases the accuracy for detecting overlapping community structures. Meanwhile, by updating node labels in ascending order of node importance and historical label preferred strategy, LPANNI observably solves the random problem of label propagation algorithms, and thus greatly increase the stability of community detection results. It is worth noting that, LPANNI use parameter α to control the topology information between node pairs to measure node similarity, while bigger α will increase the time cost. Based on experiments, $\alpha = 3$ is appropriate for most networks.

However, LPANNI can only be applied to homogeneous networks. When extending LPANNI to heterogeneous networks, we may face some new challenges that the different node types and edge types should be considered to reasonably measure the NI, similarity between nodes and NNI. Our following work is to extend LPANNI to detect overlapping community structures in heterogeneous networks.

Acknowledgment

The work was supported in part by the National Science Foundation of China grants 61471060.

References

[1] M. E. J. Newman and M. Girvan, "Finding and Evaluating Community Structure in Networks," *Physical Review E*, 69, 026113 (2004), doi: 10.1103/PhysRevE.69.026113

[2] H. L. Yan, J. Xiang, X. Y. Zhang and J. F. Fan, "Community Detection Using Global and Local Structural Information," *Pramana - J Phys*, (2013) 80 (1), pp. 173-185, doi: 10.1007/s12043-012-0359-5

[3] V. D. Blondel, J. L. Guillaume, R. Lambiotte and E. Lefebvre, "Fast Unfolding of Community Hierarchies in Large Networks," *J Stat Mech*, 2008, abs/0803.0476

[4] J. Huang, H. Sun, Q. Song, H. Deng and J. Han, "Revealing Density-Based Clustering Structure from the Core-Connected Tree of a Network," *IEEE Transactions on Knowledge & Data Engineering*, 2013, 25(8), pp. 1876-1889, doi: 10.1109/TKDE.2012.100A.

[5] Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the Overlapping and Hierarchical Community Structure of Complex Networks," *New Journal of Physics*, 2008, 11(3), pp. 19-44, doi: 10.1088/1367-2630/11/3/033015

[6] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding Statistically Significant Communities in Networks," *Plos One*, 2011, 6(4), pp. 336-338, doi:10.1371/journal.pone.0018961

[7] J. Eustace, X. Wang, and Y. Cui, "Overlapping Community Detection Using Neighborhood Ratio Matrix," *Physica A Statistical Mechanics & Its Applications*, 2015, 421, pp. 510-521, doi: 10.1016/j.physa.2014.11.039

[8] S. Bandyopadhyay, G. Chowdhary, D. Sengupta. "FOCS: Fast Overlapped Community Search," *IEEE Transactions on Knowledge & Data Engineering*, 2015, 27(11): 2974-2985

[9] T. Chakraborty, S. Kumar, N. Ganguly, et al. "GenPerm: A Unified Method for Detecting Non-Overlapping and Overlapping Communities," *IEEE Transactions on Knowledge & Data Engineering*, 2016, 28(8): 2101-2114

[10] U. Raghavan, R. Albert, and S. Kumara, "Near Linear Time Algorithm to Detect Community Structures in Large-scale Networks," *Physical Review E*, 2007, 76(3), pp. 036106

[11] H. Lou, S. Li, and Y. Zhao, "Detecting Community Structure Using Label Propagation with Weighted Coherent Neighborhood Propinquity," *Physica A Statistical Mechanics & Its Applications*, 2013, 392(14), pp. 3095–3105, doi: 10.1016/j.physa.2013.03.014

[12] Z. Lin, X. Zheng, N. Xin, and D. Chen, "CK-LPA: Efficient Community Detection Algorithm Based on Label Propagation with Community Kernel," *Physica A Statistical Mechanics & Its Applications*, 2014, 416(C), pp. 386-399, doi: 10.1016/j.physa.2014.09.023

[13] H. Sun, J. Huang, X. Zhong, K. Liu, J. Zou, and Q. Song, "Label Propagation with α-degree Neighborhood Impact for Network Community Detection," *Computational Intelligence & Neuroscience*, 2014, vol. 2014, pp. 130689-130689, doi: 10.1155/2014/130689

[14] H. Sun, J. Liu, J. Huang, G. Wang, and Q. Song, "CenLP: A Centrality-based Label Propagation Algorithm for Community Detection in Networks," *Physica A Statistical Mechanics & Its Applications*, 2015, 436, pp. 767-780, doi: 10.1016/j.physa.2015.05.080

[15] J. Xie, B K. Szymanski. "LabelRank: A stabilized label propagation algorithm for community detection in networks," Network Science Workshop. IEEE, 2013: 138-143

[16] S. Gregory, "Finding overlapping communities in networks by label propagation," *New Journal of Physics*, 2010, 12(10), pp. 2011-2024, doi: 10.1088/1367-2630/12/10/103018

[17] J. Xie, B. K. Szymanski, and X. Liu, "SLPA: Uncovering Overlapping Communities in Social Networks via a Speaker-Listener Interaction Dynamic Process," *2011 11th IEEE International Conference on Data Mining Workshops*, IEEE Computer Society, 2011, pp. 344-349

[18] H. Sun, J. Huang, Y. Tian, Q. Song and H. Liu, "Detecting Overlapping Communities in Networks via Dominant Label Propagation," *Chinese Physics B*, 2015, 24(1):551-559, doi: 10.1088/1674-1056/24/1/018703

[19] Xing Y, Meng F, Zhou Y, et al. "A node influence based label propagation algorithm for community detection in networks". *The Scientific World Journal*, 2014, 2014(5): 627581

[20] Zhang X K, Ren J, Song C, et al. "Label propagation algorithm for community detection based on node importance and label influence". *Physics Letters A*, 2017, 381

[21] Tong C, Niu J, Wen J, et al. "Weighted label propagation algorithm for overlapping community detection". *IEEE International Conference on Communications*. IEEE, 2015: 1238-1243

[22] Xie J, Kelley S, Szymanski B K. "Overlapping community detection in networks: The state-of-the-art and comparative study". *ACM Computing Surveys, vol. 45, no. 4.* ACM, 2013, p.43(35)

[23] Chen W, Liu Z, Sun X, et al. "A game-theoretic framework to identify overlapping communities in social networks". *Data Mining & Knowledge Discovery*, 2010, 21(2): 224-240

[24] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark Graphs for Testing Community Detection Algorithms," *Physical Review E*, 2008, 78(4), pp. 561-570, doi: 10.1103/PhysRevE.78.046110

[25] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Catania, "Extending the Definition of Modularity to Directed Graphs with Overlapping Communities," *Statistical Mechanics: Theory and Experiment*, 2009, pp. 03024, doi: 10.1088/1742-5468/2009/03/P03024

[26] A. F. Mcdaid, D. Greene, N. Hurley. "Normalized Mutual Information to Evaluate Overlapping Community Finding Algorithms", *Computer Science*, 2011.

[27] L. M. Collins, C. W. Dent. "Omega: A General Formulation of the Rand Index of Cluster Recovery Suitable for Non-disjoint Solutions", *Multivariate Behavioral Research*, 1988, 23(2):231

[28] S. Fortunato, D. Hric. "Community Detection in Networks: A User Guide", *Physics Reports*, 2016, 659:1-44

Meilian Lu is an associate professor in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. She received her Ph.D degree in Communication and Information System from Beijing University of Posts and Telecommunications in 2012. Her research interests are data mining and large-scale complex network analysis.

Zhenglin Zhang is a graduate student in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. His research interests are data mining and community detection in large-scale complex network.

Zhihe Qu is a graduate student in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Her research interests are data mining and dynamic community detection in large-scale complex network.

Yu Kang is a graduate student in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Her research interests are data mining and location promotion in location-based social network.