Physics Letters A ••• (••••) •••–•••

Contents lists available at ScienceDirect

Physics Letters A

www.elsevier.com/locate/pla

Label propagation algorithm for community detection based on node importance and label influence

Xian-Kun Zhang [1], Jing Ren, Chen Song, Jia Jia, Qian Zhang

College of Computer Science and Information Engineering, Tianjin University of Science and Technology, No. 29, 13th Avenue, TEDA, Binhai New Area, Tianjin, 300457, PR China

ARTICLE INFO

Article history:
Received 28 March 2017
Received in revised form 9 June 2017
Accepted 10 June 2017
Available online xxxx
Communicated by C.R. Doering

Keywords:
Large-scale social network
Community detection
Label propagation
Node importance
Label influence

ABSTRACT

Recently, the detection of high-quality community has become a hot spot in the research of social network. Label propagation algorithm (LPA) has been widely concerned since it has the advantages of linear time complexity and is unnecessary to define objective function and the number of community in advance. However, LPA has the shortcomings of uncertainty and randomness in the label propagation process, which affects the accuracy and stability of the community. For large-scale social network, this paper proposes a novel label propagation algorithm for community detection based on node importance and label influence (LPA_NI). The experiments with comparative algorithms on real-world networks and synthetic networks have shown that LPA_NI can significantly improve the quality of community detection and shorten the iteration period. Also, it has better accuracy and stability in the case of similar complexity.

1. Introduction

Social network is a relatively stable social relationship with the interaction result between individual members. Social network is commonly abstracted into a graph, in which nodes represent individual members and edges (or links) are the relationship between individual members [1]. A community is a subgraph containing nodes which are more densely linked to each other than to the rest of the graph or equivalently, a graph has a community structure if the number of links into any subgraph is higher than the number of links between those subgraphs [24]. Given a network diagram, the process of finding its community structure is called community detection. In social network, community detection for social network analysis is important. In recent decades, many social network community detection algorithms have been proposed. According to the solution strategy, algorithms can be divided into optimization methods and heuristic methods [2]. Optimization methods achieve community detection through setting the objective function and iterating to get approximation of the optimal value of the objective function. The representative algorithms of optimization methods include spectral algorithms and modularity maximization algorithms [3–9]. The spectral algorithms [3,4] transform community identification problem into simple quadratic optimization problem, and the approximate optimal partition of the network is obtained by solving the characteristic vectors of Laplacian matrix. Modularity maximization methods are to find the maximum value of the modularity function (Q function) in the network, whose representative algorithms include FN algorithm [5], GA algorithm [6] and EO algorithm [7]. Heuristic methods find the optimal division of communities by setting heuristics rules, and its representative algorithms include GN algorithm [10] and WH algorithm [11].

In addition, there are other effective community detection algorithms. For example, hierarchical clustering algorithms regard the social network as the composition of the multi-layer community, and use the traditional hierarchical clustering algorithm to carry out community detection. Evolutionary computation algorithms regard the process of community detection as the optimization process of the objective function, and use evolutionary computation methods to find the approximate optimal solution to the problem. These algorithms generally have high time and space complexity, and they are not suitable for large-scale social network structure detection. In 2007, Raghavan et al. [12] proposed a fast community detection algorithm LPA based on label propagation. The complexity of LPA is nearly linear time, and the design of the algorithm is simple, so LPA has good efficiency in dealing with large-scale network. Moreover, LPA neither require a predefined objective function optimization nor need community quantity and scale. All of these make LPA receive quite a lot of attention from numerous scholars. However, in the process of updating the node label, LPA has the shortcomings of uncertainty and randomness, which leads

E-mail address: zhxkun@tust.edu.cn (X.-K. Zhang).
[1] Fax: 86 022 60274450.

http://dx.doi.org/10.1016/j.physleta.2017.06.018
0375-9601/

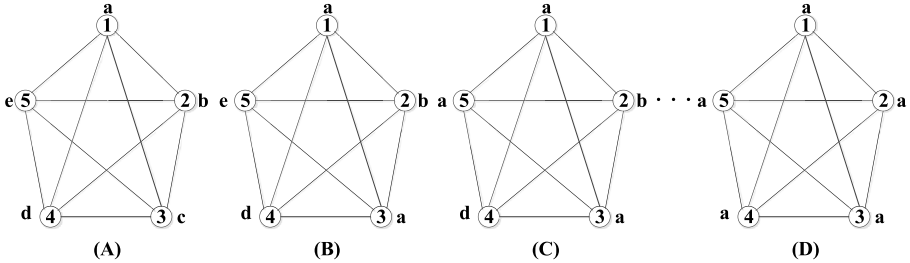

Fig. 1. Label propagation process (Raghavan et al. [12]).

to poor results in accuracy and stability of the algorithm. Literature [19] takes into account the influence of node degree and edge weight during the label updating process. But in large-scale social networks, such as the micro-blog social network, the influence of the nodes' prior attribute on the node importance is not considered.

In this paper, we propose a novel label propagation algorithm for community detection based on node importance and label influence in networks (LPA_NI). Firstly, LPA_NI uses a novel node importance method to evaluate node importance. Secondly, in the iterative process, LPA_NI fixes the node orders of label updating in the descending order of node importance. As the number of multiple labels reaches maximum, LPA_NI calculates the influence of each label and selects the most influential label to update node label. Finally, experiments with comparative algorithms on real-world networks and synthetic networks have shown that LPA_NI can significantly improve the quality of community detection and shorten the iteration period. Also, it has better accuracy and stability in the case of similar complexity.

The rest of the paper is organized as follows. Section 2 introduces the label propagation algorithm and the node importance calculation algorithm based on Bayesian network. In Section 3, this paper introduces the main idea and the detailed process of our algorithm. The experimental results on real-world networks and synthetic networks in Section 4 confirm the effectiveness of the algorithm. The conclusion is given in Section 5.

2. Related work

A complex network can be modeled as a graph $G = (V, E)$, where $V = \{v_1, v_2, \cdots, v_n\}$ represents the set of nodes, $E = \{e_1, e_2, \cdots, e_m\}$ represents the edges between the nodes, and n, m represent the number of nodes and edges in the network, respectively. Each edge in E has a pair of nodes in V corresponding. The label of v_i is denoted as c_i, $N(i)$ represents the neighborhood set of v_i and $d(i)$ is the degree of node i.

2.1. Label propagation algorithm for community detection in networks

LPA is a kind of heuristic algorithm, and the main heuristic rules of LPA is to transfer label information constantly between nodes. After several iterations, the node label belonging to the same community will converge. The main idea of LPA is that each node changes its label to the one carried by the largest number of its adjacent nodes (as shown in Fig. 1). The process of LPA can be described as the following steps:

Input: Network $G = (V, E)$, maximum number of iterations $\max Iter$.

Output: Community set $Setc = \{c_1, \cdots, c_k\}$, and k is the number of community.

(1) Initialize the unique label for each node in the network. For a given node x, $c_x(0) = x$.

(2) Set iteration number $t = 1$.

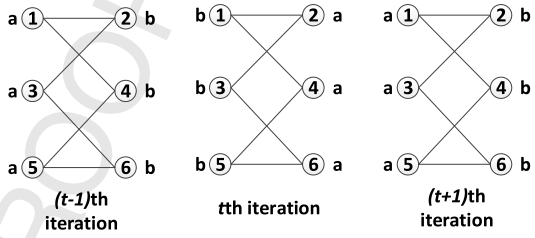

Fig. 2. The oscillation of labels in a bipartite graph (Xing et al. [19]).

(3) Arrange the nodes of the network in random order, and generate an ordered sequence $X = \{x_1, x_2, \cdots, x_n\}$.

(4) For each node $x \in X$, iteratively update the node label according to the formula (1) or (2). In the iterative process, each node changes its label to the one carried by the largest number of its adjacent nodes. As the number of multiple labels reaches maximum, LPA randomly selects one of them to update the node label. Label updating methods can be divided into synchronous updating method and asynchronous updating method. Synchronous updating method is to update the label of node x in the tth iteration based on the label of the adjacent nodes at the $(t-1)$th iteration. The formula is as follows:

$$c_x(t) = f\big(c_{x1}(t-1), \cdots, c_{xk}(t-1)\big),\ x_i \in N(x), \tag{1}$$

where $c_x(t)$ represents the label of node x in the t iteration, and $N(x)$ represents the neighborhood set of node x. Synchronous updating may occur oscillation phenomenon in bipartite or nearly bipartite structure network (as shown in Fig. 2), and asynchronous updating can be a good solution to this problem [12]. Asynchronous updating is to update the label of node x in the tth iteration based on a portion of labels at the tth iteration of its adjacent nodes which have already been updated in the current iteration and another part of labels at the $(t-1)$th iteration which are not yet updated in the current iteration. The formula is as follows:

$$c_x(t) = f\big(c_{x1}(t-1), \cdots, c_{xm}(t-1), c_{x(m+1)}(t), \cdots, c_{xk}(t)\big),\ x_i \in N(x), \tag{2}$$

(5) If iteration number $t = \max Iter$ or the label of each node is the same as that of most of its neighboring nodes, then the nodes with the same label are placed in the same community, and the algorithm ends; otherwise, set $t = t + 1$ and go to step (3).

A label propagation process is shown in Fig. 1. Firstly, LPA initializes all the nodes in the graph, and all the 5 nodes are assigned unique labels: "a", "b", "c", "d" and "e". Secondly, algorithm randomly selects node 3 to update its label, because the neighboring nodes around node 3 are labeled with the maximum value of 1, node 3 randomly selects one of them to update its label as "a". Then, algorithm randomly selects node 5 to update its label, because there are two neighbors of node 5 that share label "a" and only the label "a" has a maximum value of 2, node 5 updates its

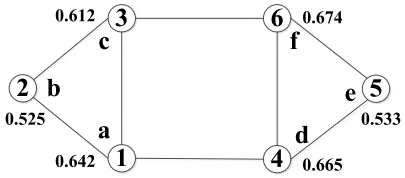

Fig. 3. The social network sample graph.

label as "a". Finally, the labels of the whole nodes have been updated for label "a", the label propagation algorithm end.

Although asynchronous updating method can avoid the oscillation problems of labels, it can be seen from the above steps that the uncertainty and randomness of LPA are mainly from the randomness of update sequence and uncertainty in node labels choosing process in step (4), which affected the accuracy and stability of LPA algorithm. We analyze traditional LPA with the social network sample graph in Fig. 3. There are 2 communities in the network, $\{1, 2, 3\}$ and $\{4, 5, 6\}$. The 6 nodes in the graph are initialized with label "a", "b", "c", "d", "e" and "f", respectively. Assuming that the labels of nodes 1, 2, 3 have been updated to label "a", the labels of nodes 4, 5, 6 are still "d", "e", "f". If we randomly select node 4 to update and label "a" is chosen as its label, then we update node 6, and finally update node 5. The result is that all nodes in network will be updated to label "a", and all nodes will be divided into a community. Conversely, if node 4 is updated to label "f", then update node 5, and finally update node 6. The result of the network will be divided into 2 communities, $\{1, 2, 3\}$ and $\{4, 5, 6\}$, and the results will correspond with the right communities.

In recent years, many scholars have improved the standard LPA algorithm. Steve et al. [13] extended the standard LPA algorithm, and proposed an algorithm COPRA for detecting overlapping community structure by allowing each node to retain a number of community labels. Barber et al. [14] proposed a constrained label propagation algorithm LPAm, which converted LPA as an optimization problem, and gave the corresponding objective function H, so it could solve the problem of clustering performance of LPA. Leung et al. [15] found that after five iteration of LPA, 95% of the nodes had been correctly gathered. After that, the iteration was mainly to update the nodes labels in the community. Therefore, they improved the updating rules and the iterative rules of LPA. Zhang et al. [16] proposed a LPAc algorithm based on edge clustering coefficient, which improved the label updating process by calculating the edge clustering coefficient. Literature [17] used the method of local cycles to solve the random problem in label updating process. However, these algorithms do not take into account the importance of nodes and the influence of different labels. In large-scale social networks, the priori property of different nodes will determine the node importance, and different node importance will decide the label influence of the node labeled, so the influence of different labels will break the random rules of label propagation process, and it can significantly improve the accuracy and stability of the algorithm. Yao et al. [19] proposed a nodes' k-shell value based label propagation algorithm NIBLPA, which chooses the label with the maximum k-shell value to update the node label. But the algorithm NIBLPA doesn't consider the priori property of nodes in social networks, such as the number of fans of the user nodes, user's micro-blog released number, and user's micro-blog forwarded number in micro-blog social network. Therefore, the accuracy of node importance calculation method needs to be improved.

2.2. Node importance calculation algorithm based on Bayesian network

There are many algorithms to calculate the importance of nodes, such as degree centrality [20], clustering coefficient centrality [21], and betweenness centrality [22], but degree centrality and clustering coefficient centrality can only describe the local information of networks. And the time complexity of betweenness centrality is too high, so it is only suitable for small networks. For large-scale social networks, Zhang et al. [23] proposed a user node importance calculation algorithm based on Bayesian network, which considered the messages in micro-blog social network are high redundancy, fast rate of transmission and high timeliness. Firstly, the algorithm gives a deep analysis of users' behavior patterns in micro-blog social network. Secondly, it uses a Bayesian model to learn prior probability of property of user node, and identifies important users by manual identification. Thirdly, it uses expert knowledge to obtain the prior probability of each property, and studies the attribute value of the node with major influence. Finally, it establishes a model of Bayesian network to describe the relationship between user property and influence, which can normalize all the nodes importance in network. The algorithm can be described as the following steps:

(1) Obtain user's basic information which includes the total number of fans, the total number of micro-blogs, the number of concerns, the number of collections and creation time. Moreover, obtain user's micro-blog information, friend relationship and micro-blog's forwarded relations.

(2) Based on user's basic information, calculate user's accumulative number of forwarded micro-blogs, accumulative number of evaluated micro-blogs, accumulative number of praised micro-blogs, average number of forwarded micro-blogs, average number of evaluated micro-blogs, average number of praised micro-blogs and activity, and classify the micro-blog through the content of the micro-blog information.

(3) Identify the important user node manually from different fields in the micro-blog social network, then obtain the threshold that measures the influence of various properties by comparing the identified important user with the users in the dataset, finally establish user attributes-influence probability formula.

(4) Calculate the total user influence through multiplying the influence of each attribute factor. The formula is as follows:

$$P(Inf) = \prod P(Inf|Attr), \tag{3}$$

where $P(Inf)$ represents the influence of the user, and $P(Inf|Attr)$ represents the influence of each attribute of the user.

(5) Normalize the influence of all the users in the dataset, and the importance of each node is obtained.

3. Proposed method

A lot of improved label propagation algorithms randomly determine label updating order of nodes in label propagation process, because these algorithms didn't take into account that the node importance may impact label updating process, it may lead to the less important nodes affect some more important nodes in turn, and it is called "countercurrent" phenomenon in label propagation process. Besides, many algorithms only use the number of labels to measure label influence in the stage of label selection, and ignore the effect of node influence on label selection. In this section, we propose a novel label propagation algorithm for community detection based on node importance and label influence in networks (LPA_NI). Firstly, LPA_NI uses a novel method for node importance evaluation. Secondly, in the iterative process, LPA_NI updates the nodes in the descending order based on the importance of each node. As the number of multiple labels reaches maximum, LPA_NI calculates the influence of each label and selects the most influential label to update node label.

Please cite this article in press as: X.-K. Zhang et al., Label propagation algorithm for community detection based on node importance and label influence, Phys. Lett. A (2017), http://dx.doi.org/10.1016/j.physleta.2017.06.018

3.1. The basic idea

The importance of nodes is used to measure the influence of nodes in the whole network, and in this paper, we use node importance calculation algorithm based on Bayesian network to normalize the importance of all the nodes in network. In general, the greater the importance of a node has, the greater the impact on the other nodes it has, so the label of this node is more likely to be spread. But node normalized importance based on priori attributes of node is not enough, because there are close links between nodes and nodes in the network, a node linked with more important nodes is more important. Therefore, we proposed a novel node importance calculation method based on the early node importance algorithm, and the formula is as follows:

$$NI(i) = Inf(i) + \alpha * \sum_{j \in N(i)} \frac{Inf(j)}{d(j)}, \tag{4}$$

where $NI(i)$ represents the importance of node i, $Inf(i)$ represents the priori importance of node i, it is from expert knowledge, α represents a tunable parameter ranging from 0 to 1, which measures the extent of the influence of neighboring nodes on node i, $N(i)$ represents the neighborhood set of node i, and $d(j)$ is the degree of node j.

We choose node importance as the measure of node influence, so we improve the performance of stability by fixing the nodes sequence of label updating in the descending order of node importance.

Another factor that affects the stability of LPA is that when the number of multiple labels reaches maximum, the original algorithm randomly selects a label to update the node label. In this paper, we propose a label selection method using the information of the label influence, so we can calculate the influence of each label and selects the most influential label to update node label. The formula is as follows:

$$LI(i, l) = \sum_{j \in N^l(i)} \frac{NI(j)}{d(j)}, \tag{5}$$

where $LI(i, l)$ represents the influence of the label l on the node i, and $N^l(i)$ represents the set of label l around the node i.

As a result, the new label update rule formula is as follows:

$$c_i = \arg \max_{l \in l\max} LI(i, l), \tag{6}$$

where c_i represents the most influential label, and $l\max$ represents the sets of the maximum number of labels.

More specifically, when the number of multiple labels in the adjacent node of the node i reaches maximum, LPA_NI recalculates the influence of the labels that reach the maximum number according to the formula (6), and selects the most influential label to update the node i.

3.2. The steps of LPA_NI algorithm

Based on the novel node importance calculation method and the label influence calculation method, we design a novel label propagation algorithm for community detection based on node importance and label influence (LPA_NI). Specific steps are as follows:

Input: Network $G = (V, E)$, maximum number of iterations $\max Iter$.

Output: Community set $Setc = \{c_1, \cdots, c_k\}$, and k is the number of community.

(1) Initialize the unique label for each node in the network, for a given node x, $c_x(0) = x$.

(2) Calculate the node importance for each node according to the formula (4), sort nodes in descending order of NI, and generate an ordered sequence $X = \{x_1, x_2, \cdots, x_n\}$, where $NI(x_1) \geq NI(x_2) \geq \cdots \geq NI(x_n)$.

(3) Set iteration number $t = 1$.

(4) For each node $x \in X$, iteratively update the node label according to the formula (5) and (6). In the iterative process, each node uses the label with the largest importance of adjacent nodes as the label itself. When the number of multiple labels in the adjacent node of the node i reaches the maximum value, LPA_NI recalculates the influence of the labels that reach the maximum number according to the formula (6), and selects the most influential label to update node i.

(5) If iteration number $t = \max Iter$ or the label of each node is the same as that of most of its neighboring nodes, then the nodes with the same label are placed in the same community, and the algorithm ends; otherwise, set $t = t + 1$ and go to step (4).

LPA_NI algorithm updates the nodes in the descending order based on the importance of each node, so the nodes with larger node importance will influence the nodes with less node importance, which is conducive to reducing the "countercurrent" phenomenon. Furthermore, LPA_NI considers the effect of label influence on label selection strategy, so it has higher accuracy in label updating process. Thereby, LPA_NI improves the accuracy and stability of the algorithm.

We implement LPA_NI on the social network sample graph in Fig. 3 with $\alpha = 1$. The decimals and letters outside the nodes are the node importance and node labels. Using our node importance calculation method on the graph, the new importance of node 1–6 are 1.3302, 0.943, 1.3132, 1.3702, 0.9793, 1.3662, respectively, so the node updating sequence is fixed as 4-6-1-3-5-2. The label propagation process is shown in Fig. 4.

Firstly, we update the label of node 6. We use a set of tuples $(l, n, LI(l))$ to describe the adjacent node information of node 6, where l represents the label, n represents the number of the label l, and $LI(l)$ represents the influence of label l. As shown in Fig. 4(a), node 6 has 3 adjacent nodes, and the adjacent nodes have different labels, so the adjacent node information of node 6 is $\{(c, 1, 0.4377), (d, 1, 0.4567), (e, 1, 0.4897)\}$. So we choose label "e" as the new label of node 6.

Then, we update the label of node 4. After the label updating of node 6, the adjacent node information of node 4 is $\{(e, 2), (a, 1)\}$. Due to the maximum value of adjacent node of node 4 is only 1, LPA_NI no longer calculates the label influence, and selects label "e" as the new label of node 4 as shown in Fig. 4(b). The next label propagation of node 1 and node 3 are consistent with node 6 and node 4, so the labels of node 1 and node 3 are updated as new label "b", and the specific process is no longer demonstrated.

Finally, the labels of adjacent node information of node 5 and node 2 are their own labels, so they are not updated and, as shown in Fig. 4(c). The labels of all nodes are the same as that of most of their neighboring nodes, and LPA_NI ends.

In the label propagation process sample of LPA_NI, the algorithm can get the results of the two communities through an iteration, which fully consistent with the correct community division. Since there is no randomness, the LPA_NI algorithm has good stability and accuracy.

3.3. Time complexity analysis

Suppose $n = |V|$ and $m = |E|$, and we firstly analyze the time complexity of the algorithm.

(1) The time complexity of initialization for all nodes in step 1: $O(n)$.

(2) The time complexity of calculating the node importance of all nodes in step 2: $O(n)$, and the time complexity of ranking the nodes in descending order of NI based on the fast sorting algo-

Please cite this article in press as: X.-K. Zhang et al., Label propagation algorithm for community detection based on node importance and label influence, Phys. Lett. A (2017), http://dx.doi.org/10.1016/j.physleta.2017.06.018

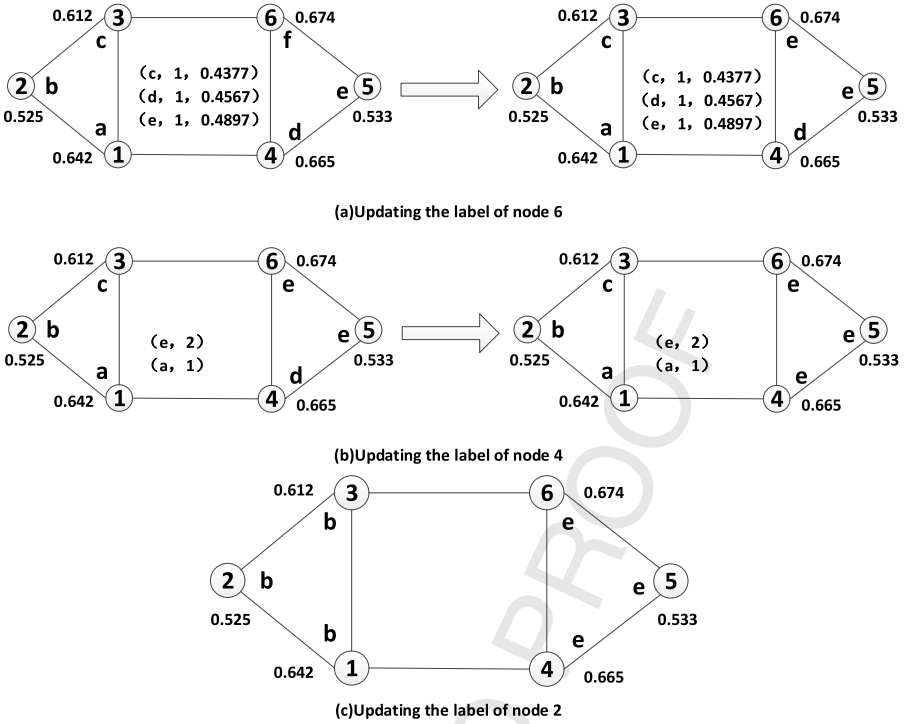

Fig. 4. Label propagation process of LPA_NI.

rithm in step 2: $O(n\log_2(n))$, so the whole time complexity of step 2 is $O(n\log_2(n)+n)$.

(3) The time complexity of normal label updating: $O(m)$, and the time complexity of recalculating the labels according to the formula (5) if necessary: $O(m)$.

(4) The time complexity of assigning the nodes with the same labels to a community in step 5: $O(n)$.

Step 4 is iterative, and maximum number of iterations is max*Iter*. The time complexity of the whole algorithm is $3\times O(n)+2\times \max Iter\times O(m)+O(n\log_2(n))$.

Then we analyze the space complexity of the algorithm.

(1) The space complexity of using adjacency list to store the node and edge information in step 1: $O(n+m)$.

(2) The space complexity of ranking the nodes based on the fast sorting algorithm in step 2: $O(n\log_2(n))$.

(3) The space complexity of assigning the nodes with the same labels to a community in step 5: $O(n)$.

So the space complexity of the whole algorithm is $2\times O(n)+O(m)+O(n\log_2(n))$.

4. Experiments and analysis

In order to verify the performance of the proposed LPA_NI algorithm, this paper uses real-world networks and synthetic networks to carry out the community detection experiment.

We present the performance of LPA_NI on several real-world complex networks with the absent of ground-truth communities: Zachary's karate club [25], Football network [26], Dolphins network [28] and the Sina micro-blog dataset which used in literature [17]. And we present the performance of LPA_NI on Girvan–Newman benchmark and LFR benchmark [29].

We compare the performance of LPA_NI with LPA and NIBLPA. Where NIBLPA is an improved LPA algorithm choosing the label with the maximum k-shell value to assign the nodes. In order to overcome the influence of the randomness of the experiments, all the algorithms are processed 50 times and the average value is used as the results, and the maximum number of iterations of the algorithm is set to 100 times. All the algorithms are implemented in Java and tested in JDK8.0 platform, and simulations are carried out in a notebook PC with Inter (R) Core (TM) CPU i5-4200 @ 2.50 GHz and 12 GB memory under Windows 10 OS.

4.1. Experiment dataset

The details of the Sina micro-blog dataset are as follows:

(1) 63641 Sina Micro-blog users' information, including uid, nickname, name, location, homepage url, gender, the number of fans, the number of concerns, the number of micro-blogs, the number of collections and creation time;

(2) 84168 micro-blogs' information about 12 topics from 2014-05-03 to 2014-05-11, including mid, release time, the micro-blog content, source, the number of forwarded micro-blog, the number of evaluated micro-blog, the number of praised micro-blog, published user uid and topic of micro-blog;

(3) 12 topics includes Meizu, XiaoMi, Rockets, Jeremy Lin, Hengda, Korean, haze, house's price, My Old Classmate, corrupt officials and transgenic;

(4) 1391718 users' relationships;

(5) 27759 forwarded micro-blog's relations.

62827 users' information contained in the dataset can restore the true Sina micro-blog network, and 12 themes of micro-blog information can restore the flow of information in micro-blog social network. Users' relationships and forwarded micro-blog's relations are enough to restore the real topology of the micro-blog Sina network.

Because the original dataset cannot meet the experimental requirements, it is necessary to preprocess the Sina micro-blog dataset. Specific operations are as follows:

(1) If and only if the user nodes are mutual concerned, there is an edge between the nodes, otherwise there is no edge between the nodes;

(2) Remove the discrete small community (the total number of nodes in community is less than 5), and obtain a network graph consisting of 1220 points and 5023 edges.

LFR benchmark is presented by Lancichinetti et al. The LFR benchmark generates static networks with built-in community structure. The configuration of generated networks depends on various user-specified parameters. Number of nodes is N, k specifies the average degree of nodes and $k_{\max}$ is the upper bound on degrees of nodes. The mixing parameter mu is the fraction of edges that a node has to the nodes outside of its community. There, as we decrease μ we obtain a clear set of communities with fewer number of inter-edges. Later, the authors adapted the LFR benchmark to generate overlapping communities. Parameter O_n specifies the number of overlapping nodes and O_m controls the number of membership of overlapping nodes. In this paper, we set $N = 1000$, $k = 15$, $k_{\max} = 50$, $\mu = 0.3$ to generate LFR benchmark and set $N = 128$, $k = 16$, $k_{\max} = 16$, $\mu = 0.1$ to generate Girvan–Newman benchmark to present the performance of LPA_NI.

4.2. Evaluation criteria

The most popular measure to compare the similarity between the delivered community structure and the ground-truth communities is Normalized Mutual Information (NMI). We have used an implementation of NMI measure made available by McDaid et al., for sets of overlapping communities [29]. Further, we use modularity value of obtained community structures when the ground-truth community structure is unknown.

Modularity is currently widely used in measuring the performance of network detection algorithms that proposed by Newman and Girvan in literature [24]. Moreover, while the underlying class labels of the real network are unknown, we can only adopt the modularity as the evaluation criteria on real networks. For a dataset with no overlapping communities, the modularity is defined as:

$$Q = \frac{1}{2m}\sum_{i,j\in V}\left(A_{ij} - \frac{d(i)d(j)}{2m}\right) \times \delta(c_i, c_j), \tag{7}$$

where Q represents the modularity, m represents the number of edges in the network, and A is the adjacency matrix of the network, if node i and node j are directly connected, $A_{ij} = 1$; elsewise, $A_{ij} = 0$. c_i and c_j, denote the label of node i and node j, respectively, if $c_i = c_j$, then $\delta(c_i, c_j) = 1$, else $\delta(c_i, c_j) = 0$.

4.3. Experiment result

In this section, the real dataset is used to test the effectiveness of LPA_NI comparing with traditional LPA and NIBLPA, and we test the stability of the algorithms by analyzing the fluctuation range of the whole results.

In order to compare the effectiveness of the algorithms, we use the average modularity Q to measure the quality of divided communities, so the greater the Q value is, the more accurate the results of community detection are, and the smaller the fluctuation range of Q value is, the more stable the community detection results are. NIBLPA and LPA_NI contain only one parameter α. And the parameter α is used to measure the extent of the influence of neighboring nodes to node i, so we let $\alpha = \{0, 0.1, 0.2 \cdots, 1\}$. More specifically, if $\alpha = 0$, the importance of the adjacent nodes of the node i has no influence, so the importance of node i is only its own importance. And if $\alpha = 1$, the importance of adjacent nodes is accumulated in multiples of 1 to obtain the new importance of node i. The experimental results are shown in Fig. 5.

As it can be seen in Fig. 5, for the real dataset used in this paper, the variation trend of the Q value of LPA_NI is like bell shaped curve with the gradual increase of parameters α, and when $\alpha = 0.4$, the maximum modularity $Q = 0.6995$ is achieved. More specifically, when $\alpha = 0.4$, the influence of adjacent nodes on the node i is the most appropriate, and the new node importance in improving the community structure detection has the best effect. The Q value of NIBLPA changes little under different parameter α, and the maximum modularity $Q = 0.6197$ is obtained when the parameters $\alpha = 0.5$ and $\alpha = 0.6$. Experimental results show that the modularity of LPA_NI is higher than NIBLPA when $\alpha = 0.4$. And with the importance of adjacency nodes to the influence of node i gradually increased, the quality of community detection of LPA_NI firstly increased, then gradually decreased. Therefore, the impact of an appropriate amount of the adjacent node importance will significantly improve the accuracy of LPA_NI, and LPA_NI can get an optimal result of community detection.

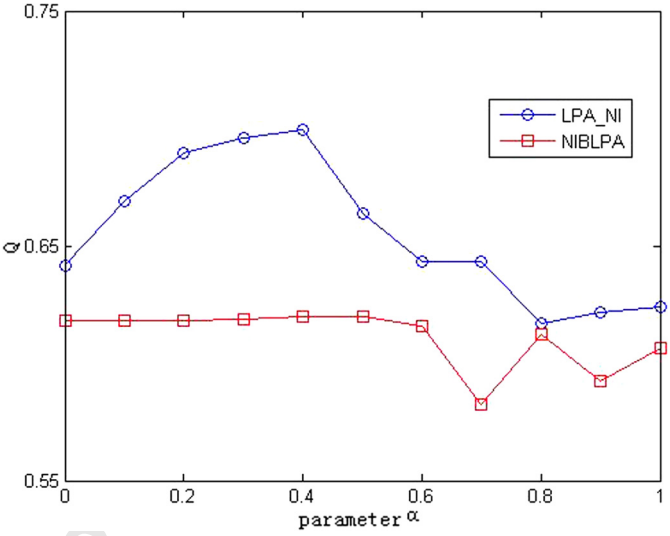

Fig. 5. The comparison of Q under different tunable parameter α.

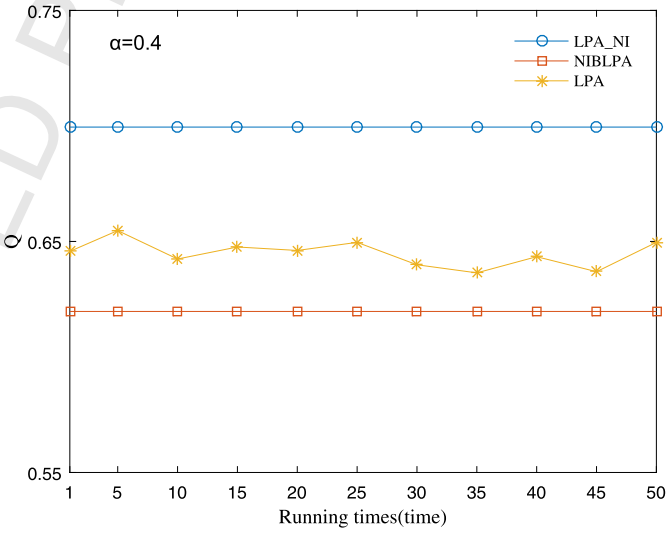

Fig. 6. 50 repeated experiments under parameter $\alpha = 0.4$.

In order to analyze the stability of LPA_NI, NIBLPA and LPA, this paper analyzes the 11 experimental results of 50 repeated experiments with parameter $\alpha = 0.4$, the variation trend of each algorithm modularity value Q is shown in Fig. 6.

It can be seen from Fig. 6 that Q values of LPA_NI and NIBLPA are 0.6995 and 0.6197 in the 50 repeated tests. Because LPA_NI and NIBLPA improve the randomness of the update sequence and the label selection process, they have good stability. However, LPA is less stable because of the randomness of the algorithm. In general, LPA_NI can get stable results than LPA and effective results than LPA and NIBLPA.

Please cite this article in press as: X.-K. Zhang et al., Label propagation algorithm for community detection based on node importance and label influence, Phys. Lett. A (2017), http://dx.doi.org/10.1016/j.physleta.2017.06.018

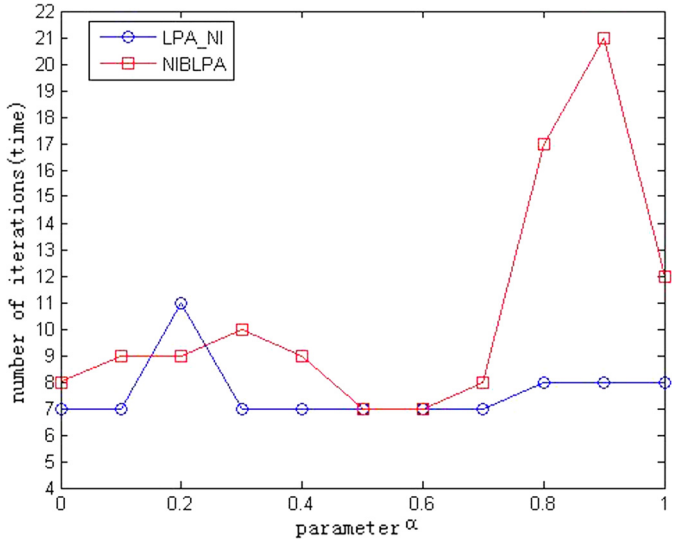

Fig. 7. The number of iterations under different tunable parameter α.

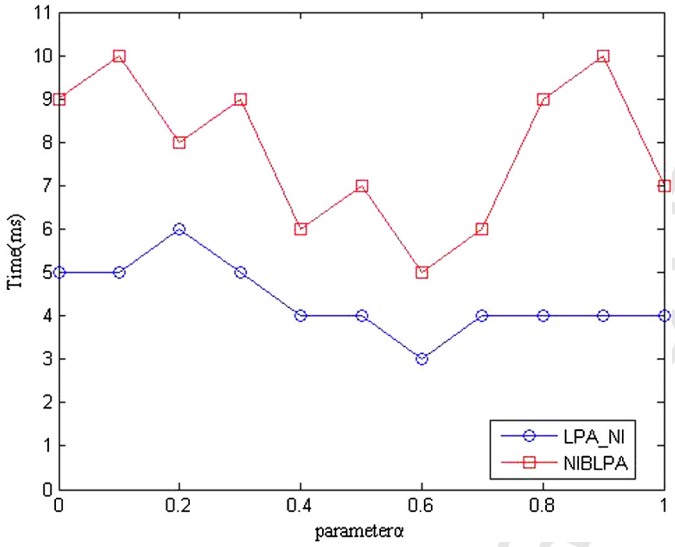

Fig. 8. The running time under different tunable parameter α.

In order to analyze the efficiency of LPA_NI, NIBLPA and LPA, Figs. 7 and 8 show the number of iteration and running time of the algorithm, respectively.

It can be seen from Fig. 7 that as the parameter α increases, the number of iterations of the LPA_NI are stable in 7 times. And NIBLPA has a larger fluctuation on the number of iterations. In the early period, the number of iterations is around 8 times, then the number will surge to 17, 21, 13 times. Experimental results show that the number of iterations of the LPA_NI is significantly less than NIBLPA, and the running time is obviously better than other algorithms. Therefore, the node importance calculated by k-shell value can not fully describe the node influence in the large-scale social network so when $\alpha \geq 0.8$, the number of iterations and running time of the algorithm will be significantly increased. LPA_NI using the priori knowledge to calculate the node importance is more accurate, so the number of iterations and running time of LPA_NI will be less, and the stability of LPA_NI is better than the other algorithms.

Then, we compare the performance of LPA_NI, NIBLPA and LPA on Zachary's karate club, Football network, Dolphins network. The experimental results are shown in Table 1. It's easy to see LPA_NI achieve higher Q than other algorithms on different networks.

Table 1
Q obtained by LPA_NI, NIBLPA and LPA on different real-world networks.

Networks	Q_{LPA_NI}	Q_{NIBLPA}	Q_{LPA}
Karate	0.4151	0.3944	0.3590
Football	0.5805	0.5762	0.5899
Dolphins	0.5143	0.5184	0.4797

Table 2
NMI obtained by LPA_NI, NIBLPA and LPA on different synthetic networks.

Networks	NMI_{LPA_NI}	NMI_{NIBLPA}	NMI_{LPA}
LFR	1	1	1
Girvan–Newman	0.9989	0.9989	0.9989

We also show the performance of LPA_NI on LFR and Girvan–Newman benchmark in Table 2. We compare our method with LPA and NIBLPA. It's easy to see LPA_NI achieve higher NMI on different networks.

In summary, on the one hand, LPA_NI takes into account that the node importance may impact label updating process, so it will let the more important nodes affect some less important nodes, and it can reduce the "countercurrent" phenomenon in label propagation process. On the other hand, LPA_NI takes into account that the effect of label influence on label selection. Therefore, it can improve the accuracy in label selection process. As it can be seen, LPA_NI can significantly improve the quality of community detection and shorten the iteration period. Also, it has better accuracy and stability in the case of similar complexity.

5. Conclusion

This paper presents a label propagation algorithm for community detection based on node importance and label influence. The algorithm firstly uses a novel method for node importance evaluation, and fixes the node orders of label updating in the descending order of node importance. During each label updating process, when the number of multiple labels reaches the maximum value, this paper introduces the label influence into the formula of label updating to improve the stability. This algorithm maintains the advantages of the original LPA algorithm, and it can get stable and efficient results by avoiding the randomness in label propagation process. By experiments on real dataset, we demonstrate that the proposed algorithm has better performance than some of the current representative algorithms.

Uncited references

[18] [27]

References

[1] Youfang Lin, Tianyu Wang, Rui Tang, Yuanwei Zhou, Houkuan Huang, An effective model and algorithm for community detection in social networks, J. Comput. Res. Dev. 49 (2) (2012) 337–345.
[2] Dayou Liu, Di Jin, Dongxiao He, Jing Huang, Jianning Yang, Bo Yang, Community mining in complex networks, J. Comput. Res. Dev. 50 (10) (2013) 2140–2154.
[3] M. Shiga, I. Takigawa, H. Mamitsuka, A spectral approach to clustering numerical vectors as nodes in a networks, Pattern Recognit. 44 (2) (2011) 236–251.
[4] Liang Huang, Ruixuan Li, Hong Chen, Xiwu Gu, Kunmei Wen, Yuhua Li, Detecting network communities using regularized spectral clustering algorithm, Artif. Intell. Rev. 41 (4) (2014) 579–594.
[5] R. Guimera, M. Sales-Pardo, L.A. Amaral, Modularity from fluctuations in random graphs and complex networks, Phys. Rev. E 70 (2) (2004) 188.
[6] R. Guimera, Functional cartography of complex metabolic networks, Nature 433 (7028) (2005) 895–900.
[7] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Phys. Rev. E 72 (2) (2005) 986.
[8] Zhan Bu, Chengcui Zhang, Zhengyou Xia, Jiandong Wang, A fast parallel modularity optimization algorithm (FPMQA) for community detection in online social network, Knowl.-Based Syst. 50 (3) (2013) 246–259.

Please cite this article in press as: X.-K. Zhang et al., Label propagation algorithm for community detection based on node importance and label influence, Phys. Lett. A (2017), http://dx.doi.org/10.1016/j.physleta.2017.06.018

[9] S. Zhang, H. Zhao, Normalized modularity optimization method for community identification with degree adjustment, Phys. Rev. E 88 (5–1) (2013) 471.
[10] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (12) (2002) 7821–7826.
[11] F. Wu, B.A. Huberman, Finding communities in linear time: a physics approach, Eur. Phys. J. B-Condens. Matter 38 (2) (2004) 331–338.
[12] U.N. Raghavan, R. Albert, S. Kumara, Near linear-time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (3) (2007) 036106.
[13] G. Steve, Finding overlapping communities in networks by label propagation, New J. Phys. 12 (10) (2010) 2011.
[14] M.J. Barber, J.W. Clark, Detecting network communities by propagating labels under constraints, Phys. Rev. E 80 (2) (2009) 026129.
[15] I.X. Leung, P. Hui, P. Lio, J. Crowcroft, Towards real time community detection in large networks, Phys. Rev. E 79 (6) (2009) 066107.
[16] XianKun Zhang, Xue Tian, YaNan Li, Chen Song, Label propagation algorithm based on edge clustering coefficient for community detection in complex networks, Int. J. Mod. Phys. B 28 (30) (2014).
[17] XianKun Zhang, Song Fei, Chen Song, Xue Tian, YangYue Ao, Label propagation algorithm based on local cycles for community detection, Int. J. Mod. Phys. B 29 (5) (2015).
[18] J. Xie, B.K. Szymanski, LabelRank: a stabilized label propagation algorithm for community detection in networks, Netw. Sci. Workshop (2013) 138–143.
[19] Y. Xing, F. Meng, Y. Zhou, M. Zhu, M. Shi, G. Sun, A node influence based label propagation algorithm for community detection in networks, Sci. World J. 2014 (5) (2014) 627581.
[20] J. Sohn, D. Kang, H. Park, B.G. Joo, I.J. Chung, An Improved Social Network Analysis Method for Social Networks, Springer, Netherlands, 2014, pp. 115–123.
[21] L. Šubelj, M. Bajec, Group detection in complex networks: an algorithm and comparison of the state of the art, Phys. A, Stat. Mech. Appl. 397 (3) (2014) 144–156.
[22] O. Green, D.A. Bader, Faster betweenness centrality based on data structure experimentation, Proc. Comput. Sci. 18 (1) (2014) 399–408.
[23] Xian-Kun Zhang, Jia Jia, Chen Song, Ranking the importance of user node in micro-blog social network, Comput. Eng. Des. (2016).
[24] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[25] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. (1977) 452–473.
[26] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.
[27] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (4) (2008) 046110.
[28] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, P. Spyridonos, Community detection in social media performance and application considerations, Data Min. Knowl. Discov. 24 (3) (2012) 515–554.
[29] A.F. Mcdaid, D. Greene, N. Hurley, Normalized mutual information to evaluate overlapping community finding algorithms, Comput. Sci. (2011).

Please cite this article in press as: X.-K. Zhang et al., Label propagation algorithm for community detection based on node importance and label influence, Phys. Lett. A (2017), http://dx.doi.org/10.1016/j.physleta.2017.06.018