# Accepted Manuscript


Stepping community detection algorithm based on label propagation and similarity

Wei Li, Ce Huang, Miao Wang, Xi Chen

PII: S0378-4371(17)30022-5
DOI: http://dx.doi.org/10.1016/j.physa.2017.01.030
Reference: PHYSA 17923

To appear in: *Physica A*

Received date: 27 August 2016
Revised date: 6 December 2016

Please cite this article as: W. Li, C. Huang, M. Wang, X. Chen, Stepping community detection algorithm based on label propagation and similarity, *Physica A* (2017), http://dx.doi.org/10.1016/j.physa.2017.01.030

***Highlights (for review)**



Highlights of our paper are as follows:

A new label propagation-based algorithm was proposed for detecting community structure in networks.

The algorithm proposed in this paper divides networks with a stepping framework.

The algorithm propagates labels based on the similarity between nodes or subnetworks.

A novel evaluation function is defined to get the final optimized partition.

The accuracy in community detection is highly improved.

# Stepping community detection algorithm based on label propagation and similarity

Wei Li[1, 2], Ce Huang[1, 2], Miao Wang[3], Xi Chen*[1, 2]

[1]School of Automation, Huazhong University of Science and Technology, Wuhan, 430074, P. R. China

[2]Image Processing and Intelligent Control Key Laboratory of Education Ministry of China, Wuhan, 430074, P. R. China

[3]China Ship Development and Design Center, Wuhan, 430064, China

**Abstract:** Community or module structure is one of the most common features in complex networks. The label propagation algorithm (LPA) is a near linear time algorithm that is able to detect community structure effectively. Nevertheless, when labeling a node, the LPA adopts the label belonging to the majority of its neighbors, which means that it treats all neighbors equally in spite of their different effects on the node. Another disadvantage of LPA is that the results it generates are not unique. In this paper, we propose a modified LPA called Stepping LPA-S, in which labels are propagated by similarity. Furthermore, our algorithm divides networks using a stepping framework, and uses an evaluation function proposed in this paper to select the final unique partition. We tested this algorithm on several artificial and real-world networks. The results show that Stepping LPA-S can obtain accurate and meaningful community structure without priori information.

**Keywords:** community detection, label propagation, similarity, stepping algorithm, evaluation function

## 1. Introduction

Many complex systems in different fields, including social networks, blogging communities, biological systems, and power grids can be represented by networks comprising nodes and links. A node or vertex represents an entity in the system, and a link or connection is an interaction between a pair of entities. Community or module structure is one of the most common features in real-world networks. At present, there is no specific definition of a community, but it is generally regarded as a group of nodes in which nodes are more densely connected with internal nodes than with those in other communities. Detecting community structure is an important approach in complex network analysis. Research on community structure may help us understand the topology and functions of complex networks, and also can be applied in the real world to solve problems.

In recent years, researchers have proposed various methods for community detection. Newman and Girvan defined a quality function modularity Q and optimized it to detect communities [1][2]. Since then, modularity has been widely used in many methods as an evaluation index of the strength of community structures. Some other optimization algorithms like simulated annealing [3], genetic algorithms [4][5], and extreme optimization [6] have been combined with modularity to partition networks as well. Many methods with different perspectives also perform well: divisive algorithms select links between communities and define the modular structure by removing them from the network [7][8], some algorithms detect communities using the concept of a random walk [9][10], and spectral methods find communities by computing the eigenvectors of a Laplacian matrix [11][12].

A fast community mining method, the label propagation algorithm (LPA), was proposed by Raghavan et al. [13]. It changes each node's label to the most frequent label of its neighbors' label at each iteration, and stops when none of the nodes meets the update criterion. LPA is a simple and unsupervised algorithm without any parameters; furthermore, it does not need any information about the size and number of communities in advance. Besides, its low computational complexity means that it may be suitable for partitioning large networks in real time.

However, because a random factor exists in this algorithm when the number of most frequent labels is larger than one, the partitioned results it generates are not unique. This disadvantage may prevent LPA from being widely used in practice. In addition, LPA may produce a meaningless solution in which all nodes are assigned to one community [13]. In order to solve this problem, Barber et al. [14] proposed a modularity-specialized LPA called LPAm. However, this method may become stuck in a local optimum, leading to inaccurate partitions [14]. LPAm+ is an improved approach to obtain the highest modularity value and can effectively avoid local maxima [15]. Other researchers also have improved and extended different aspects of LPA. Gregory proposed a modified algorithm called the community overlap propagation algorithm (COPRA) [16] to detect overlapping communities. Lou et al. suggested that information in addition to neighbors' labels should be used to analyze the community structure, and proposed a method called LPA-CNP-E [17] that updates a node's label depending on a similarity weighted coherent neighborhood propinquity (CNP). Although LPA has distinct advantages for community detection, there have also been many superior LPA-based methods, and there is still room for partitioning real-world networks more accurately.

In this paper, we propose a non-overlapping community mining algorithm for unweighted and undirected networks called Stepping LPA-S. The main steps of this stepping method proceed as follows. First, node labels are updated depending on a similarity metric S [21] until a stable result is reached. This results in a network consisting of several small subnetworks. Second, we merge the two most similar subnetworks in a random order and then use our proposed objective function to determine which level should be selected as the final solution.

The rest of this paper is organized as follows. In Section 2, we describe the original LPA and our algorithm, Stepping LPA-S. In Section 3, we present the experiments and results. Finally, in Section 4, the conclusion and a discussion of our algorithm are given.

## 2. Algorithm

In this section, we first introduce the original LPA, the similarity metric we adopt, and the modularity function, which together form the basis of our algorithm, and then we introduce our new stepping community mining algorithm, Stepping LPA-S, which analyzes networks in two phases. It propagates node labels depending on the similarity between two nodes or subnetworks in the first and second phases, respectively. Furthermore, we propose an evaluation function DN for determining the best partition.

### 2.1. Original LPA

The LPA was first applied to community detection by Raghavan et al. [13]. The goal of LPA is to divide networks efficiently without any advance information such as the size and number of communities. The steps of the standard LPA are as follows:

(1) Assign a unique initial label to every node.
(2) Generate a random visiting order for all nodes in the network.
(3) Update a node's label in this order to the one owned by the majority of its neighbors. If there is more than one label with maximum frequency, ties are broken randomly.
(4) If every node shares the same label as the majority of its neighbors, the algorithm stops and nodes with the same label are assigned to one community. Otherwise, return to step 2 and repeat the process.

### 2.2. Stepping LPA-S

Before describing the proposed algorithm in detail, we introduce the prominent

quality function, modularity $Q$ [1] first. It compares the real density of intra-module links and the expected density in a random network without any community structure called a null model. The null model used here keeps the degree of each node in the original network, but connects the nodes randomly over the whole network [2].In this circumstance, the expected number of links connecting node $i$ and $j$, $P_{ij}$ can be written as

$$P_{ij} = 2m \cdot \frac{k_i}{2m} \cdot \frac{k_j}{2m} = \frac{k_i k_j}{2m}, \tag{1}$$

where $m$ is the total number of links in the network, and $k_i$ and $k_j$ are the degrees of vertices $i$ and $j$, respectively. With this null model, the mathematical expression of modularity reads

$$\mathrm{Q} = \frac{1}{2m}\sum_{ij}(A_{ij} - P_{ij})\delta\left(C_i, C_j\right) = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta\left(C_i, C_j\right), \tag{2}$$

where $A_{ij}$ is a component in the adjacency matrix, $C_i$ indicates to which community node $i$ belongs, and the value of $\delta$ is one if node $i$ and $j$ are in the same community ($C_i = C_j$) and zero otherwise. A high value of $Q$ means that the partition is markedly different from a random network, thus it has strong community structure.

The basic LPA adopts the label belonging to the majority of its neighbors, which means that it treats all neighbors equally. However, in fact, a node plays different roles in the network depending on its features. In the real world, entities with high similarity tend be gathered in the same group. From this perspective, community detection methods using an appropriate similarity metric may discover some valuable results in practice. Among various similarity [18-20], a metric based on resource allocation was proposed [21]. When node $i$ sends a resource to node $j$, their common neighbors act as the transmitters. Similarity between $i$ and $j$ can be described as the amount of resource $j$ receives. Assuming every transmitter has a unit of resource, this similarity index in an unweighted and undirected network is [21]

$$S_{ij} = \sum_{z \in \Gamma(\mathrm{i}) \cap \Gamma(\mathrm{j})} \frac{1}{k\left(z\right)}, \tag{3}$$

where $\Gamma(i) \cap \Gamma(j)$ denotes the set of common neighbors of nodes $i$ and $j$, and $k(z)$ is the degree of node $z$.

In this study, in order to select labels according to the correlation of the nodes, we propose a modified LPA, in which labels are propagated by similarity $S$, as defined in Eq. (3). We choose the label that leads to the largest modularity when the labels of

the most similar neighbors are not unique, since peaks of modularity turn out to closely link to the expected division in some real systems [1]. This is different from the original LPA. The label propagation procedure using *S* is as follows.

(1) Initialize each node with a unique label.
(2) Calculate the similarity *S* between every pair of nodes according to Eq. 3.
(3) Generate a random visiting order for all nodes in the network.
(4) Update every node's label to that of the neighbor that has the highest value of similarity *S*.
(5) If every node has the same label as its most similar neighbors, the process stops and nodes with same label belong to one subnetwork. Otherwise, repeat steps 3 and 4.

Given the form of *S*, it is reasonable to expect that the number of neighbors with high similarity will be much lower than the degree of a node. In other words, there are only a few choices of label for updating each node. This reality prevents a label from being propagated on a large scale. For example, if nodes *i* and *j* have the same label and the similarity between them and their other neighbors is relatively low, this label is not spread to any other nodes: *i* and *j* hence form a small module. Thus, propagating labels by S results in excessive segmentation.

However, the subnetworks in an over-segmentation result of the first phase are also meaningful. The number of nodes contained in a subnetwork is small, but the similarity among them is high. Hence, we consider partitioning networks with a two-step procedure. In the second phase, a subnetwork is regarded as a super-node. We define the *S*-based similarity $S_{sub}$ between every pair of subnetworks as follows:

$$S_{sub,x,y} = \frac{\sum_{i \in x, j \in y} S_{ij}}{\min(\sum_{i \in x} k(i), \sum_{j \in y} k(j))}, \tag{4}$$

where *x* and *y* denote two separate subnetworks and $k(i)$ is the degree of node *i*. The second phase of the partition algorithm is given in steps 6-9:

(6) Calculate the $S_{sub}$ similarity between every pair of subnetworks according to Eq. 4.
(7) Generate a random visiting order for all subnetworks in the network.
(8) Change labels of the nodes in a subnetwork to that of the most similar subnetwork. Update the similarity matrix.
(9) If there are two communities left in the entire connected network, the process stops and the nodes with same label belong to one community. Otherwise, repeat steps 7-8.

From the second phase, various divisions are produced, and we need to choose one of these as the final partition. We suggest that a good module structure should have a node degree density inside the communities that is much higher than it is outside the communities. Suppose there are $n$ nodes and $n'$ subnetworks in the network, the fraction of inner degree (FID) is then

$$FID = \sum_{x=1}^{n'} \frac{k_{inx}}{k_x - k_{inx}}, \tag{5}$$

and the fraction of inner nodes (FIN) is

$$FIN = \sum_{x=1}^{n'} \frac{n_x}{n - n_x}, \tag{6}$$

where $x$ denotes a subnetwork, $k_{inx}$ is the inner degree of the nodes in $x$, $k_x$ is the total degree of nodes in $x$, and $n_x$ is the number of nodes in $x$. The new evaluation function we propose can be defined as DN = FID/FIN. When there is more than one community in a network, the larger the value of DN, the better the structure. We calculate DN after each label propagation in step 8 and select the partition with the largest DN as the final result.

Note that, because of the random factors in steps 3 and 7, we obtain several different partitions over multiple runs. Thus, we should run Stepping LPA-S for a sufficient number times and then choose the partition that has the highest DN. The number of runs required depends on the scale of the network and is considered reasonable if the similarity of two ultimate partitions measured by normalized mutual information (NMI) [22] (see Section 3.1) is higher than 0.98.

In a word, this method uses similarity S to propagate the labels of all the nodes until a stable result is reached. This intermediate result is excessively segmented. We then repeatedly merge the two most similar subnetworks randomly until there are two communities in the network. After every merging, the value of evaluation function DN is calculated, and the structure with the highest DN is our final result.

## 3. Experiments and results

In this section, we show the results generated by our algorithm for computer-generated networks and four real-world networks in Table 1, whose community structures are known.

### 3.1. NMI

In Section 2.2, we introduce a quality function modularity that evaluates a

partition from the perspective of community structure. We also employ the NMI [22], a popular index in community detection, to measure the degree of similarity of the communities found by the proposed algorithm to the real ones. The NMI of partitions A and B is

$$I(A,B)=\frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B}N_{ij}\log\left(\frac{N_{ij}N}{N_{i\cdot}N_{\cdot j}}\right)}{\sum_{i=1}^{c_A}N_{i\cdot}\log\left(\frac{N_{i\cdot}}{N}\right)+\sum_{j=1}^{c_B}N_{\cdot j}\log\left(\frac{N_{\cdot j}}{N}\right)}, \tag{7}$$

where $N_{ij}$ is the number of nodes in community $i$ of A that appear in community $j$ of B, $c_A$ denotes the number of real communities, and $c_B$ denotes the number of found communities. If the found partition is the same as the real one, NMI reaches its maximal value of 1. If the two partitions are uncorrelated, NMI takes the minimal value 0.

In this study, we use the modularity and NMI as our performance indicators.

### 3.2. Computer-generated networks

In this part, we use two types of popular synthetic networks to test the performance of Stepping LPA-S algorithm. Note that the evaluation indices of LPA are the average values of 100 trials.

#### 3.2.1. Girvan-Newman networks

We used Girvan-Newman (GN) networks [7] to evaluate the performance of our algorithm. Each network contains 128 nodes and four communities of 32 nodes each. Links are placed between node pairs based on probabilities $p_{in}$ and $p_{out}$: $p_{in}$ denotes the probability of generating a link between two nodes that are in the same community and $p_{out}$ denotes the probability of generating a link between two nodes that are in different communities. The average degree $k$ of a node is kept equal to 16, and we can obtain different networks by changing the average number of inter-module links $k_{out}$. Clearly, a lower value of $k_{out}$ leads to a clearer community structure. When $k_{out} > 8$ and the average inner degree is less than eight, the original structure does not meet the definition of community.

The results of LPA and Stepping LPA-S are plotted in Figs. 1 and 2. These figures show that the partitions of Stepping LPA-S have a higher NMI accuracy than the LPA partitions. When the community structure is not evident ($k_{out} \geq 7$), the average modularity of LPA is close or equal to zero. If the value of modularity is zero, all nodes are assigned to one community and the divisions are uninformative. This is a

drawback of LPA [13]. However, when the structure is inconspicuous, Stepping LPA-S can still detect groups that have denser links inside the communities than outside the communities. In addition, in Fig. 2, although the values of NMI for both algorithms decrease as $k_{out}$ increases, the proposed Stepping LPA-S obtains higher accuracy than the LPA. Our algorithm is able to obtain the correct partitions when $k_{out} \leq 4.5$, while the average values of NMI for the LPA are equal to one when $k_{out} \leq 2$. Furthermore, when $k_{out} = 5.5$, these values decline to about 0.6, nevertheless, the same values obtained by Stepping LPA-S stay around 0.9. Thus, in all cases, our method performs better than the basic LPA.

3.2.2. Lancichinetti–Fortunato–Radicchi networks

The Lancichinetti-Fortunato-Radicchi (LFR) benchmark networks [23] make up another standard synthetic data set for testing community detection. They have heterogeneous distributions for node degree and community size, both of which are power law distributions. Several parameters should be set before generating an LFR network: the number of nodes N, average degree $\langle k \rangle$, maximum degree $k_{max}$, minimum community size $c_{min}$, maximum community size $c_{max}$, exponent $\gamma$ for the degree distribution, exponent $\beta$ for the community size distribution, and mixing parameter $\mu$, which represents the fraction of links connecting each node with nodes in its community. In this study, all parameters except $\mu$ were set as follows: N = 1000, $\langle k \rangle = 20$, $k_{max} = 50$, $c_{min} = 20$, $c_{max} = 100$, $\gamma = 2$, and $\beta = 1$. The definition of $\mu$ means that nodes have more inter-community links than intra-community links and the modular structure is fuzzy when $\mu > 0.5$. Thus, the LFR networks used here were generated by setting the mixing parameter $\mu \in [0.1, 0.6]$ in steps of 0.05.

In Figs. 3 and 4, we show the results for LPA and Stepping LPA-S in terms of modularity and NMI. The trends in variation of both indicators are similar to those obtained on the GN networks. When the mixing parameter $\mu$ is lower than or equal to 0.2, both algorithms perform well and the values of NMI are almost one. When $\mu$ increases, the fraction of inter-community edges increases and the accuracy of both algorithms degrades. However, Stepping LPA-S always obtains partitions closer to the real structures. When $\mu = 0.5$, the NMI of Stepping LPA-S's results stay around 0.9, while the average NMI obtained by LPA declines 0.69. In addition, the modularity is approximately zero. Thus, Stepping LPA-S improves the performance of label propagation.

### 3.3. Real-world networks

We compared the results of LPA, LPA-CNP-E [17], and our proposed Stepping LPA-S method on four real-world networks to evaluate the obtained communities. The results of modularity and NMI are shown in Table 2. The evaluation indices of LPA for each network are the average values of 100 trials.

#### 3.3.1. Zachary's Karate Club network

Zachary's Karate Club network is one of the most commonly used benchmark networks in community mining [24]. This network consists of 34 nodes and 78 links. Each member denotes a node in the network, and a link exists between two nodes if these two members interact consistently outside the activities of the club. However, the administrator (node 1) and instructor (node 33) fell out, and the members were split into two groups with either administrator or instructor. Our final result, consisting of three communities, is shown in Fig. 5. Node 10 has a unique label; thus, there is one community just containing one node. Node 10 and the community centered on node 1 together correspond to a real group, and the community centered on node 33 correspond to the other real group. The value of NMI is higher than 0.9, which affirms the high accuracy of this partition.

#### 3.3.2. Dolphin Social network

The second network used here consists of a dolphin social network that has been investigated by Lusseau [25] over several years. It is composed of 62 dolphins and 159 links, which were built by observational frequent contact. This network is divided into two groups of 20 and 42 nodes. As shown in Fig. 6, the structure discovered by our algorithm contains two communities. Compared with the real division, only node 40 is misclassified. The value of NMI is 0.8888, which is much higher than the average values of LPA and LPA-CNP-E.

#### 3.3.3. US College Football network

The third network we discuss is a college football network, which represents games among Division I of the US College Football League during the 2000 season [7]. In this network, each node denotes a team, and a link between a pair of nodes denotes that these two teams played together. There are 115 nodes and 613 edges in total, and nodes are divided into 12 communities corresponding to the "conferences." Games between teams in the same conference are more frequent than between teams in different conferences. Our community result is shown in Fig. 7, which contains 13

communities. As Fig. 7 reveals, the proposed algorithm detects the obvious community structure in this network, and 105 of the 115 nodes are assigned correctly. In addition, the value of NMI reaches 0.9259. There are two small communities in the final partition, and they are indeed a subset of a real community. Hence, this result also reflects a part of the real relations among football teams.

3.3.4. Political Books network

Another benchmark is a network of books about American politics compiled by Valdis Krebs. This network involves 105 nodes, each of which represents a book about US politics sold on Amazon.com. All the nodes were classified into three groups according to their political inclination in [26]: liberal, conservative, or centrist. If customers frequently bought two books at the same time, the two nodes are connected by an edge. The maximal DN detects four communities, as presented in Fig. 8. From the figure, we can see that the result shows a clear community structure, and our method performs well on this network. The values of the NMI via Stepping LPA-S are higher than the average results of LPA, and the value of NMI is near to that of LPA-CNP-E.

While Stepping LPA-S is a modified algorithm based on LPA, we also compared it with some other common algorithms (Louvain [27], Spectral algorithm [2], Fast algorithm [28], Infomap [29]) in these real-world networks. The results in Table 3 show that the partitions obtained by our method are much more accurate than those of other methods as well. It suggests that our method can generate competitive results in the pool of existing community detect algorithms.

## 4. Conclusions

The original LPA only requires the original connections of the network, and it does not rely on any specific functions. These features make it a practical approach for studying real-world networks that have no information about the modules. However, it only takes the labels of a node's neighbors into account, which means that it treats all neighbors equally and neglects their different influences on the node. In this paper, we suggest that a node should be in the same community as its most similar neighbors. Our algorithm, Stepping LPA-S, is proposed based on this idea. This stepping method first update s node labels depending on *S* in [21] until a stable result is reached. We then merge the two most similar subnetworks obtained in the first step, and use our newly proposed objective function to determine which level to select as the final solution. Experiments were conducted with both

computer-generated networks and several real-world networks with known partitions.

From Fig. 2, we can see that the values of NMI obtained by both the LPA and Stepping LPA-S in synthetic GN networks decreases as the community structure becomes more inconspicuous, but our method can obtain better and significant divisions than LPA in general. When $0 < k_{out} < 2.5$, which means that the community structure is obvious, both algorithms obtain accurate results, with NMI values equal to 1. When $k_{out}$ is in the range [2.5, 4.5], our method obtains exactly the correct partitions, while the values of NMI obtained by the LPA are 0.9012. When $k_{out}$ increases to 5.5 and the community structure is comparatively indistinct, LPA's NMI values decrease sharply to 0.6748, while our method's NMI values remain larger than 0.8700, which demonstrates the high accuracy of its result. As shown in Figs. 4, when the community structures in LFR networks are relative fuzzy, the algorithm proposed in this study obtains obvious higher accuracy than LPA as well. In order to explain these cases, we first analyze the implication of evaluation function DN. DN represents the normalized relative density of degree: the portion of average intra-module degree relative to the average inter-module degree. A higher DN value indicates that the intra-module nodes connect with each other more tightly. The other advantage roots in the similarity indices we used here. They were proposed from the perspective of resource allocation and can imitate the process of information dissemination in the complex networks simply. Our algorithm takes both the relevance between the nodes (via similarity) and the topology of the whole network (via DN and modularity) into account, while the basic LPA just considers the local information of the nodes. Therefore, Stepping LPA-S performs better than LPA. Furthermore, LPA assigns all nodes to one community in homogeneous networks, but our algorithm can select partitions satisfying the widely accepted definition of community because of its use of the DN and modularity functions and hence avoid the uninformative partitions produced by the LPA. Thus, Stepping LPA-S is able to obtain accurate and meaningful partitions.

We also compare results of LPA, LPA-CNP-E, and Stepping LPA-S on four real-world networks including Zachary's Karate Club, Dolphin Social, US College Football, and Political Books networks. Table 2 shows that Stepping LPA-S performs better than LPA and LPA-CNP-E in terms of NMI.

Our algorithm partitions networks using the stepping process. We take the intermediate result of the Dolphin Social network in Fig. 8 as an example to illustrate the necessity of the second partition. From this figure, we can see that the

intermediate result includes 23 subnetworks. Eight of them belong to one real community and the rest belong to the other, which means that we have over-segmented this network into 23 accurate submodules. It indicates that similarity *S* is suitable for detecting communities via label propagation. In contrast, some labels of nodes whose degree is small are chosen by a few or none of the other nodes. This feature makes it hard for a label to be propagated over a large area, and results in an over-segmented partition. Hence, we adopt a further division, and in the final partition of the Dolphin Social network, only node 40 is misclassified. Node 40 has been assigned into the wrong module in previous studies [2][30], so this result is not only accurate, but also meaningful from the perspective of community definition. In the second phase, we regard every subnetwork as a special node, and use evaluation function DN to divide networks into proper communities. Based on this framework, the accuracy of our algorithm improves the LPA results by more than 42% for Zachary's Karate Club network and more than 64% in the Dolphin Social network. Furthermore, it improves the LPA-CNP-E results by 10% in Zachary's Karate Club network and 22% in the Dolphin Social network compared to LPA-CNP-E. In the other two real-world networks, Stepping LPA-S can also get better results. Relatively poor performance in the latter two networks may result from their larger scale and less-evident real community structure. In the US College Football network, some teams played nearly as many games against teams in other conferences as they did against teams in their own conference. In the Political Books network, there are dense links between the neutral group and other modules, so the topological structure of neutral is not clean. Our results fit the definition of community structure, and can give us a rational understanding of these networks.

Higher values of the three evaluation indicators imply better performance. However, in the Dolphin Social and US College Football networks, our method obtains lower modularity but higher NMI than the other two methods. This phenomenon appears to be due to the definition of modularity and the features of the real-world networks. The basic idea is that intra-module links are denser than expected, and its form is based on a null model. However, perhaps this form of modularity does not fit every real system. In addition, our method obtains the numbers of communities close to the truth while the selection of the final partitions just relies on the evaluation function DN. Thus, Stepping LPA-S can also help predict the number of communities of a network without priori information. In conclusion, DN provides a fresh and practical perspective on module formation.

The runtime of our Stepping LPA-S is mainly composed of two aspects: computing the similarities and label propagation. In the first phase, the calculation of the similarities $S$ between all node pairs requires $n*(n-1)/2$ times where $n$ is the number of nodes, so the computational complexity is $O(n^2)$. In the second phase, since two of $n'$ subnetworks are merged every time until two subnetworks remain, computing $S_{sub}$ takes time $O(n'^2)$. In every iteration, changing labels can be considered as the same process among nodes and super-nodes as the original LPA so that it requires time $O(m_t+n_t)$ ($n_t$ and $m_t$ is the total number of nodes and super-nodes and links among them respectively). When the iteration repeats *iter* times, the computational cost is $O(iter*(m_t+n_t))$. Experimental results indicated that algorithm converges in all used networks in 5 iterations. Since that $n'^2 \ll n^2$ and $iter*(m_t+n_t) \ll n^2$, the overall computational complexity $O(n^2+n'^2+iter*(m_t+n_t))$ can be simplified into $O(n^2)$. Although it does not take linear time any more, Stepping LPA-S is faster than some methods with the complexity of $O(n^3)$ [1, 7, 31] or $O(n^2 \log n)$ [26]. As shown in Table 3, comparing with some classical ones, our proposed algorithm has obvious advantage in accuracy. Therefore, Stepping LPA-S achieves a tradeoff between time cost and accuracy, and this computational complexity is acceptable in practice.

Overall, Stepping LPA-S, which considers both the similarity between node pairs and the global features via modularity and DN is a promising method for community detection. However, unweighted and non-overlapping network can not represent all complex systems. Thus, in the future, how to apply the proposed algorithm in weighted networks and networks with overlapping community structure is an important task.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (61473131, 71571081), 973 Project of China (2013CB329506), and the Fundamental Research Funds for the Central Universities of China.

## References

[1] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2004) 026113.

[2] M.E.J. Newman. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74 (2006), 036104.

[3] R. Guimerà, M. Sales-Pardo, L.A.N. Amaral, Modularity from fluctuations in

random graphs and complex networks, Phys. Rev. E 70 (2004), 025101.

[4] R. Shang, J. Bai, L. Jiao, C. Jin, Community detection based on modularity and an improved genetic algorithm, Physica A 392 (2013) 1215–1231.

[5] M. Tasgin, A. Herdagdelen, H. Bingol, Community Detection in Complex Networks Using Genetic Algorithms. 2006. arXiv: 0711.0491.

[6] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Phys. Rev. E 72 (2005) 027104.

[7] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (2002) 7821–6.

[8] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks., Proc. Natl. Acad. Sci. USA 101 (2004) 2658–2663.

[9] H. Zhou, Distance, dissimilarity index, and network community structure, Phys. Rev. E 67 (2003) 61901.

[10] H. Zhou, R. Lipowsky, Network Brownian Motion: A New Method to Measure Vertex-Vertex Proximity and to Identify Communities and Subcommunities, in: Comput. Sci. - ICCS 2004, 2004: pp. 1062–1069.

[11] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 888–905.

[12] A.Y. Ng, M.I. Jordan, Y. Weiss, On Spectral Clustering: Analysis and an Algorithm, Adv. Neural Inf. Process. Syst. 14. (2002) 849–856.

[13] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (2007) 036106.

[14] M.J. Barber, J.W. Clark, Detecting network communities by propagating labels under constraints, Phys. Rev. E 80 (2009) 026129.

[15] X. Liu, T. Murata, Advanced modularity-specialized label propagation algorithm for detecting communities in networks, Physica A 389 (2010) 1493–1500.

[16] S. Gregory, Finding overlapping communities in networks by label propagation, New J. Phys. 12 (2010) 103018.

[17] H. Lou, S. Li, Y. Zhao, Detecting community structure using label propagation with weighted coherent neighborhood propinquity, Physica A 392 (2013) 3095–3105. .

[18] B. Yan, S. Gregory, Detecting community structure in networks using edge prediction methods, J. Stat. Mech. 09 (2012).

[19] Q.J. Jiao, Y. Huang, H. Bin Shen, Community mining with new node similarity

by incorporating both global and local topological knowledge in a constrained random walk, Physica A 424 (2015) 363–371.

[20] J. Wu, Y. Hou, Y. Jiao, Y. Li, X. Li, L. Jiao, Density shrinking algorithm for community detection with path based similarity, Physica A 433 (2015) 218–228.

[21] T. Zhou, L. Lü, Y.C. Zhang, Predicting missing links via local information, Eur. Phys. J. B. 71 (2009) 623–630.

[22] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech. 09 (2005).

[23] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (2008) 046110.

[24] W.W. Zachary, An Information Flow Model for Conflict and Fission in Small Groups, J. Anthropol. Res. 33 (1977) 452–473.

[25] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, S.M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: Can geographic isolation explain this unique trait?, Behav. Ecol. Sociobiol. 54 (2003) 396–405.

[26] M.E.J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA 103 (2006) 8577–8582.

[27] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks. J. Stat. Mech. (2008) 10008.

[28] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E. 69 (2004) 066133.

[29] M. Rosvall, C.T. Bergstrom, Maps of random walks on complex networks reveal community structure., Proc. Natl. Acad. Sci. USA 105 (2008) 1118–23.

[30] S. Zhang, H. Zhao, Community identification in networks with unbalanced structure, Phys. Rev. E 85 (2012) 066114.

[31] H. Lu, H. Wei, Detection of community structure in networks based on community coefficients, Physica A 391 (2012) 6156–6164.

**Tables**

| Datasets | Nodes | Links | Communities | Description |
|---|---|---|---|---|
| **Karate** | 34 | 78 | 2 | Zachary's Karate Club |
| **Dolphins** | 62 | 159 | 2 | Dolphin Social Network |
| **Football** | 115 | 613 | 12 | US College Football Network |
| **Polbooks** | 105 | 441 | 3 | Books about US Politics |

Table 1. Datasets used in the experiments

| Data set | Quality measure | LPA | LPA-CNP-E [17] | Stepping LPA-S |
|---|---|---|---|---|
| **Karate** | modularity | 0.3573 | 0.3027 | 0.3715 |
| | NMI | 0.6493 | 0.8370 | 0.9241 |
| **Dolphins** | modularity | 0.4868 | 0.4633 | 0.3787 |
| | NMI | 0.5402 | 0.7313 | 0.8888 |
| **Football** | modularity | 0.5897 | 0.6006 | 0.5754 |
| | NMI | 0.8928 | 0.9098 | 0.9259 |
| **Polbooks** | modularity | 0.5117 | 0.4511 | 0.4967 |
| | NMI | 0.5242 | 0.5710 | 0.5712 |

Table 2. Quality of the results of LPA, LPA-CNP-E, and Stepping LPA-S for the four real-world networks

| Data set | Quality measure | Louvain | Spectral algorithm | Fast algorithm | Infomap | Stepping LPA-S |
|---|---|---|---|---|---|---|
| **Karate** | modularity | 0.4151 | 0.4188 | 0.3807 | 0.4151 | 0.3715 |
| | NMI | 0.7071 | 0.5866 | 0.6925 | 0.7071 | 0.9241 |
| **Dolphins** | modularity | 0.5196 | 0.5265 | 0.4955 | 0.5204 | 0.3787 |
| | NMI | 0.4743 | 0.5792 | 0.5727 | 0.5632 | 0.8888 |
| **Football** | modularity | 0.6043 | 0.6009 | 0.5682 | 0.5634 | 0.5754 |
| | NMI | 0.8850 | 0.8848 | 0.7436 | 0.9214 | 0.9259 |
| **Polbooks** | modularity | 0.5268 | 0.5260 | 0.5020 | 0.5127 | 0.4967 |
| | NMI | 0.4187 | 0.4160 | 0.4396 | 0.4677 | 0.5712 |

Table 3. Quality of the results of Louvain, Spectral algorithm, Fast algorithm, Infomap, and Stepping LPA-S for the four real-world networks

**Figures**

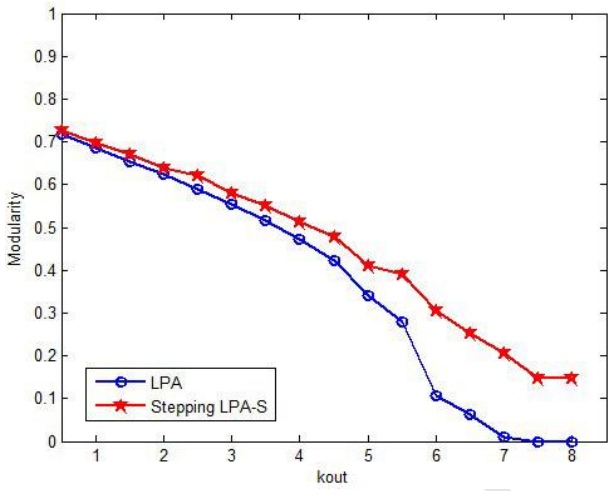


Fig. 1. Modularity of the results obtained by LPA and proposed Stepping LPA-S methods on GN networks.

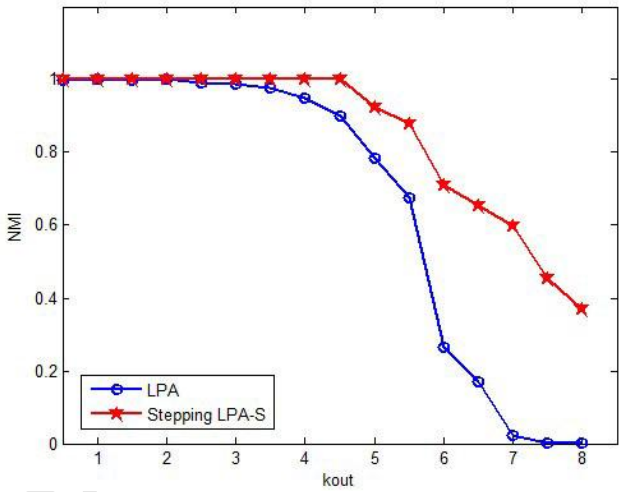


Fig. 2. NMI of the results obtained by LPA and the proposed Stepping LPA-S methods on GN networks.

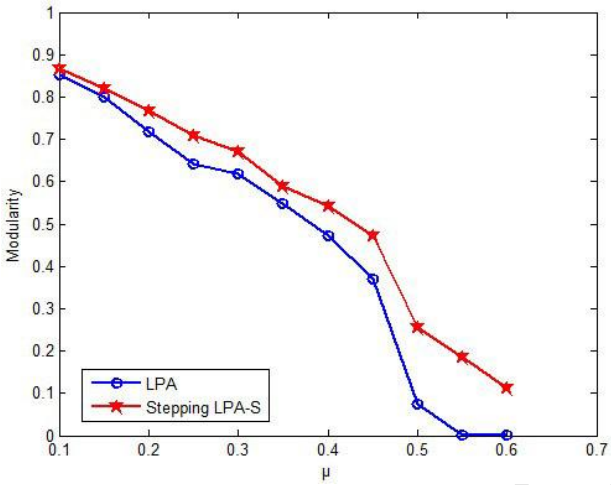


Fig. 3. Modularity of the results obtained by LPA and the proposed Stepping LPA-S methods on LFR networks.

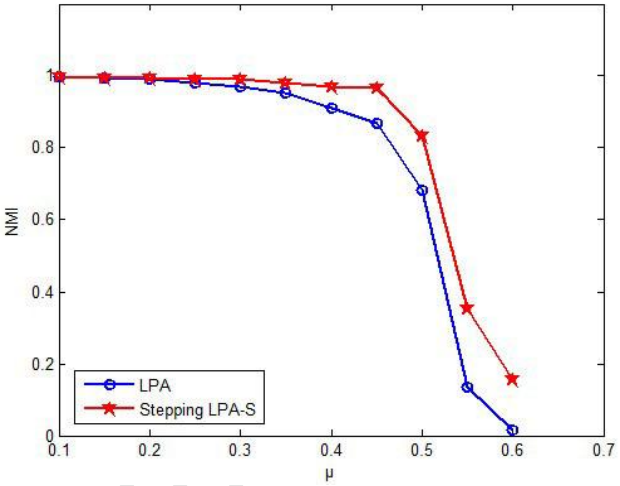


Fig. 4. NMI of the results obtained by LPA and the proposed Stepping LPA-S methods on LFR networks.

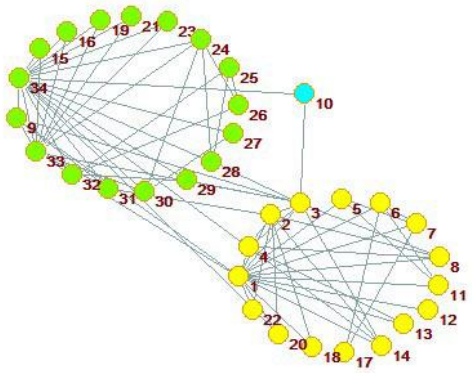

Fig. 5. Detected community structure in the Zachary's Karate Club network using the

proposed method.

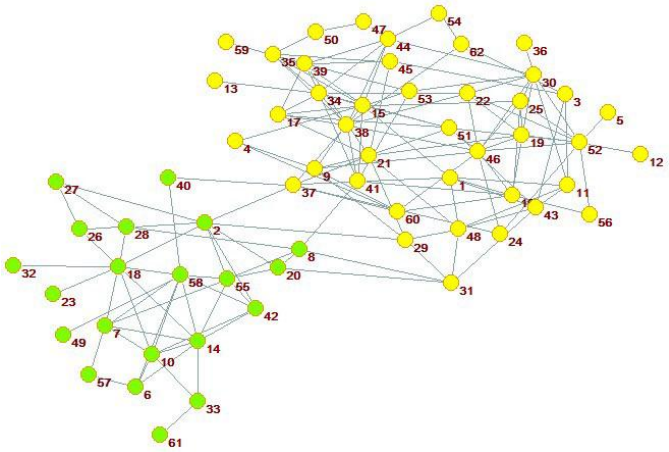

Fig. 6. Detected community structure in the Dolphin Social network using the proposed method.

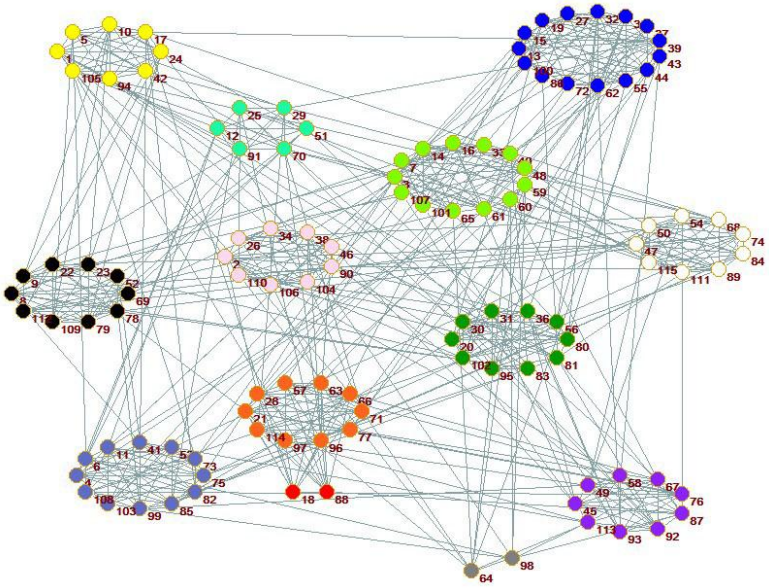

Fig. 7. Detected community structure in the US College Football network using the proposed method.

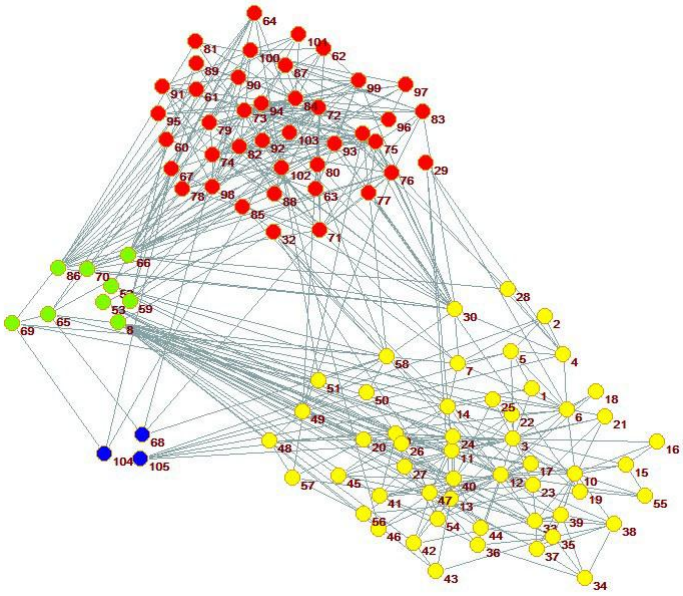

Fig. 8. Detected community structure in the Political Books network using the proposed method.

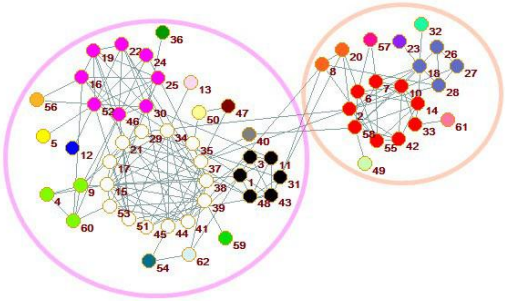

Fig. 9 Intermediate results for the Dolphin Social network. Nodes in different colors represent different subsets, and the groups enclosed in the big circles correspond to the two real communities.