

A Novel Method of Influence Ranking via Node Degree and H-index for Community Detection

Qiang Liu¹(✉), Lu Deng¹, Junxing Zhu¹, Fenglan Li¹,
Bin Zhou^{1,2}, and Peng Zou¹

¹ College of Computer, National University of Defense Technology,
Changsha 410073, Hunan, People's Republic of China
liuqiang1981@nudt.edu.cn

² State Key Laboratory of High Performance Computing,
National University of Defense Technology,
Changsha 410073, Hunan, People's Republic of China

Abstract. Identifying influential nodes is critical to have a better understanding of the network function and the process of information diffusion. Traditional methods of evaluating influential nodes such as degree centrality ignore the location of a node and its neighbors' influence in networks, while this plays an important role in revealing the node's local influence in spreading information. In this paper, we propose a novel method, named DH-index (node Degree and H-index), to measure a node's importance by considering its and neighbors' influence simultaneously. Meanwhile, we put forward a node DH-index based label propagation algorithm (DH_LPA) for community detection. We demonstrate its validity and feasibility on a set of real-world and synthetic networks for our new proposed community detection method.

Keywords: Influence ranking · Information diffusion · Community detection · Label propagation

1 Introduction

In recent years, community detection problem has attracted widespread research from many scholars around the world, and a great number of methods have been proposed. A detailed survey of community detection can be found in [1]. One general problem concerning community detection is that there is still no well-established precise definition of community. In general, a community in a network is described as a group of nodes with dense connections within groups and sparse connections with others. Discovering communities plays an important role in revealing the structure and function characteristics of networks. For instance, in virtual social network such as twitter, it's necessary to detect possible communities of terrorists or reactionary organization so as to avoid any criminal behaviors in real life, which may bring tremendous damage to a country or its people.

Among all the proposed community detection algorithms [2–7], the label propagation algorithm (LPA), proposed by Raghavan [7], has greatly received attention for its near linear time complexity in finding communities in large scale networks.

The LPA utilizes the diffusion of label information of each node to detect communities and does not need any prior knowledge of community structure, such as the number of communities. Nevertheless, the random update order of label information lead to the poor robustness of community detection results. Then, a lot of improved LPA methods were proposed. Barber [8] et al. reformulated the LPA as an equivalent optimization problem, and put forward an improved LPA based on modularity constraints. Leung et al. [9] found that the original LPA may produce large communities due to the fact that some labels can plague a large amount of nodes during the process of label propagation. Then, they proposed an improved LPA based on hop attenuation and node preference so as to avoid finding monster communities. Subelj et al. [10] also presented an improved LPA that combines two unique strategies of community formation, namely, defensive preservation and offensive expansion of communities. Besides, in view of the disadvantage of randomly selecting initial nodes problem of traditional LPA, other methods based on how to select initial nodes for LPA are also proposed. He et al. [11] utilized PageRank to measure node centrality and put forward a node importance based LPA. Sun et al. [12] proposed a centrality-based LPA with specific update order and node preference to uncover communities. However, the PageRank method of [11] is degenerated into degree centrality and does not consider the importance of the node to its neighbors; the centrality-based LPA also uses the degree centrality to computer local density for selecting initial nodes for expansion.

In fact, the initial nodes selection problem for community detection can also be seen as an influence ranking problem. That is because the formation of communities in a network is decided by its important nodes. These nodes are more influential than other nodes, and then other nodes around the influential nodes form communities. Therefore, how to specify a quantitatively exact influence measure is crucial. By far, there are six widely used methods to measure a node's influence, which are degree, closeness, betweenness, eigenvector, katz and core centrality. The disadvantage of the former five methods had been illustrated in Ref. [13]. The utilization of core to measure a node's influence is proposed by Kitsak et al. [14], they deemed that a node's location is more important than the number of its linked neighbors. According to core theory, a node with more linked neighbor nodes on the edge of a network may not be influential compared to a node in the center of a network. Therefore, they advised that the coreness can better measure a node's influence for spreading information than degree centrality. However, calculating coreness needs global topological information of the network, while obtain this information is difficult, especially for the dynamic network whose network structure changes with time passing. Then, the coreness can not better measure a node's influence. Lately, Lǔ et al. [15] extended the concept of the H-index, which was originally used to measure the citation impact of a scholar or a journal, to qualify how important a node is to its network, and showed the H-index can better measure a node's influence in several cases compared to traditional centrality measurements mentioned above. Nonetheless, the H-index only takes into account of the influence of prominent neighbor nodes of a node to measure its influence in the light of the idea that a node is prominent if many other prominent nodes are around it. Then, the H-index ignores the influence of the node itself. Therefore, the H-index calculation of each node in the network will not reflect the node's local influence fairly.

In order to single out influential nodes for community detection, a better measurement of a node's influence is important. In the light of the advantage and disadvantage of H-index in judging influential nodes, we define a DH-index function, which not only consider the prominent neighbor nodes' influence, but also take into account of the node's influence to less prominent neighbors, to measure nodes' influence and ranking them according to this function. Then, a community detection algorithm is proposed based on spreading influential labels of ranking order according to DH-index, named DH-LPA.

The rest of paper is organized as follows. In Sect. 2, we explain and define some fundamental concepts. The Sect. 3 shows the proposed algorithm DH-LPA. In Sect. 4, we give some applications of the DH-LPA algorithm to some synthetic and real-world networks. Section 5 concludes this paper.

2 Preliminaries

2.1 Label Propagation Algorithm (LPA)

Label propagation algorithm [7] is an efficient algorithm for its nearly linear time complexity in detecting communities. According to the theory of the LPA, each node is initialized with a unique label and then let the label spread throughout the network. During the process of propagating label, each node will choose the label which is owned by most of its neighbors. Then, densely connected modules of nodes will reach a consensus on a unique label, and nodes with the same label form a community. The rule of updating community labels can be expressed as follows:

$$C_n = \arg \max_l |N^l(n)| \quad (1)$$

In (1), the $|N^l(n)|$ shows the neighbors of node n which has the label l . If there exists multiple most frequent neighbor labels, a random label will be selected among them. The course of separating label will be iterative until each node does not change its label and has a label that most of their neighbors have. As far as the efficiency of the LPA is concerned, due to its simple computation process and low time complexity, it is very fit for community detection for very large networks. However, random update order leads to the unstable detected results, which hampers its robustness and stability.

2.2 H-index

Due to the disadvantage of traditional centrality methods in measuring the influence of nodes in networks, Lü et al. [15] introduced the H-index concept to quantify how important a node is to its network in 2016. The H-index of a node is defined to be the maximum value h such that there exists at least h neighbors of degree no less than h .

For instance, the Fig. 1 is an example network consisted of 23 nodes and 40 edges [16], the degree of node 1 is 8. However, the H-index of it is 2. Because there exists at least 2 neighbors of degree no less than 2. That is to say, if the Fig. 1 is a citation network, the citation impact of the scholar (node 1) is 2.

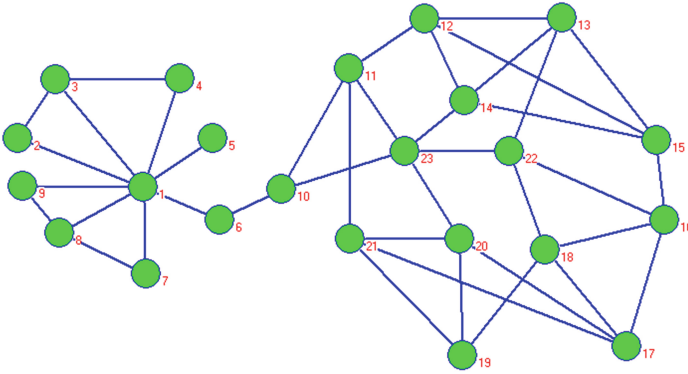


Fig. 1. An example network consisted of 23 nodes and 40 edges [16].

2.3 DH-Index

According to the definition of the H-index above, we can see that it only measures the influence of neighbor nodes, and ignores the influence of the node itself. Therefore, we combine the H-index and node degree to take into account of the influence of the node and its neighbors, and defined a function, named DH-index, to better measure a node influence in networks.

$$DHindex(n) = Hindex(n) \times Degree(n) \tag{2}$$

In (2), the $Hindex(n)$ shows the H-index of a node n , the $Degree(n)$ presents the degree of a node n . According to the DH-index function, we can see that a node’s influence is not only related to the node degree (itself influence), but also is associated with the node H-index (neighbor nodes’ influence). Then, the DH-index is reasonable in measuring the local influence around the node.

For instance, the Table 1 shows values of degree, H-index and DH-index for each node in the network of Fig. 1. From Table 1, we can see that the node 1 has largest degree among all 23 nodes. Then, if we want to choose a node to spread information faster and most broadly over the network, is node 1 a better choice for its biggest degree? Lǔ et al. [15] pointed out a node’s influence should consider its prominent neighbors’ influence, and introduced the H-index to describe a node’s influence. They regarded the bigger the H-index value of a node, the more influential of it. However, the H-index only considers the prominent neighbors’ influence, and does not take into the node’s influence of itself. Take the node 22 and 23 for example. The H-index of node 22 and 23 are 4, and then it is difficult to distinguish which node is more influential. Therefore, we regard that we should simultaneously consider the influence of node itself and its neighbors, and defined the DH-index to measure a node’s influence. Although the node 22 and 23 have the same H-index, we can see that the DH-index of node 23 is bigger than 22, which shows that the node 23 is more influential than node 22.

Table 1. Nodes information in the network

Node	Degree	H-index	DH-index
1	8	2	16
2	2	2	4
3	3	2	6
4	2	2	4
5	1	1	1
6	2	2	4
7	2	2	4
8	3	2	6
9	2	2	4
10	3	2	6
11	4	3	12
12	4	4	16
13	4	4	16
14	4	4	16
15	4	4	16
16	4	4	16
17	4	4	16
18	4	3	12
19	3	3	9
20	4	3	12
21	4	3	12
22	4	4	16
23	5	4	20

3 DH-Index Based Label Propagation Algorithm (DH-LPA)

In order to resolve the limit of the traditional LPA, we put forward a novel DH-index based label propagation algorithm for community detection. Our community detection algorithm DH-LPA includes two phases. In the first phase, we measure and quality the importance of each node. Specially speaking, we rank each node according to the DH-index value in a descending order. Then, the ranking results can better show the importance of each node in networks. The former nodes in the ranking results are more influential compared to the latter ones. In the second phase, based on the obtained update order from the first phase, the nodes in the ranking results will separate their labels to neighbors one by one. Due to the fact that the former nodes have a higher DH-index value than the latter ones, the labels of latter ones will be updated by former nodes. During the course of label propagation, there exists nodes that belong to neighbors of more than one node, and they have been updated by former nodes. For this condition, we should consider the neighbor nodes' influence of this node. If there are more nodes with the same label having high influence, we should change its label with the current ones. Eventually, theses nodes with the same label will form a community.

The details of the DH-index based label propagation algorithm are shown in Algorithm 1.

Algorithm 1 the DH-index based label propagation algorithm(DH-LPA)

Input: Network $G = (V, E)$

Output: Set of Communities C

```

1: Calculating the DH-index value of each node;
2: Ranking the nodes in a descending order according to the DH-index value,
   and getting the sorted nodes list  $L$ 
3: for node in  $L$  do
4:   for neighbor in neighbors(node) do
5:     if neighbor.update==False then
6:       neighbor.label=node.label
7:     else
8:       neighbor.label=choose the most frequent and influential node's label
9:     end if
10:  end for
11: end for
12: Saving nodes with the same label into the same community  $c$  and add it to  $C$ 

```

In Algorithm 1, the step 1 to step 2 are the first phase, and the step 3 to step 12 are the seconding phase. The main idea of the Algorithm 1 is that we first select these nodes with higher DH-index values to spreading their labels to neighbor nodes. That is because these nodes own high local influence compared to their neighbor nodes. For the latter nodes with lower DH-index values, the labels of them has been updated by the former nodes, then it's necessary to taking into account of the neighbor conditions of this node to again update this node's label according to influential nodes around it.

Let's consider the computational complexity of the DH-LPA. Suppose n be the number of nodes and m be the number of edges. According to Algorithm 1, there is two phases: (1) Calculating the DH-index values of each node and sorting them in a descending order; (2) Spreading labels according to nodes' influence. For the first phase, we need to compute the degree and the H-index of each node so as to get the DH-index value, then the time complexity is $O(3n)$. For the second phase, the label propagation process for each node has a time complexity of $O(n)$. Therefore, the total time complexity of the DH-LPA is $O(4n)$. After omitting the constant, the time complexity is $O(n)$.

4 Experiments

In this section, we conduct some experiments on several real world networks and synthetic networks so as to evaluate the performance of our proposed algorithm DH-LPA. Meanwhile, we also compare our algorithm with other well-known algorithms on benchmark network [17, 18] with known community structure. Our algorithm is implemented in Python 2.7. All the experiments were conducted on windows 7 with Intel(R) Core(TM) i5-2520 M processor, 2.5 GHz, 4G RAM.

4.1 Evaluation Metrics: Normalized Mutual Information (NMI) and Modularity

A great many of methods have been proposed for community detection, but it is not clear which method is reliable. In other words, when community partitions are found by an algorithm, a reasonable evaluation criterion should be used to evaluate how accurately the detection algorithm has performed. At present, there are two widely used evaluation methods for testing the efficiency of community detection algorithm. One is the Normalized Mutual Information (NMI), and the other is modularity. For these networks, the real community partitions of which are known, we can use NMI to test the performance of algorithm. If we do not know the real partitions of corresponding network, such as real-world networks, we can use the modularity to check the performance of community detection method. The bigger of NMI and modularity, the better of the partition results are, which can illustrate the efficiency of community detection algorithm.

The NMI, proven by Danon et al. [19], is a reliable criterion in evaluating community partitions. It can evaluate the similarity between the real partitions and the detected ones. Given two partitions A and B of a network in communities. Let C be the confusion matrix whose element C_{ij} is the number of nodes of community i of the partition A that is also in the community j of the partition B . The normalized mutual information $I(A, B)$ is defined as follows:

$$I(A, B) = \frac{-2 \sum_{i=1}^{C_A} \sum_{j=1}^{C_B} C_{ij} \log\left(\frac{C_{ij}N}{C_i C_j}\right)}{\sum_{i=1}^{C_A} C_i \log\left(\frac{C_i}{N}\right) + \sum_{j=1}^{C_B} C_j \log\left(\frac{C_j}{N}\right)} \tag{3}$$

Where C_A (C_B) is the number of groups in the partition A (B), C_i (C_j) is the sum of the elements of C in row i (column j), and N is the number of nodes. If $A = B$, $I(A, B) = 1$; if A and B are completely different, then $I(A, B) = 0$.

Modularity [17] is also a most widely used function for testing efficiency of partitioning communities for a community detection algorithm. Consider an unsigned network denoted as $G = (V, E)$, where V is the vertex set with the number of it is n ; and E is the edge set with the number of it is e . The adjacent matrix of G is A . If V_1 and V_2 are two disjoint subsets of V , then we define $L(V_1, V_2) = \sum_{i \in V_1, j \in V_2} A_{ij}$, $L(V_1, V_1) = \sum_{i \in V_1, j \in V_1} A_{ij}$, and $L(V_1, \overline{V_1}) = \sum_{i \in V_1, j \notin V_1} A_{ij}$, where $\overline{V_1} = V - V_1$. Meanwhile, we also define a partition of a network $G, G_1(V_1, E_1), G_2(V_2, E_2), \dots, G_m(V_m, E_m)$, where V_i and E_i are the aggregation of vertices and edges of G_i for $i = 1, 2, \dots, m$, the modularity Q can be defined as follows:

$$Q = \sum_{i=1}^m \left[\frac{L(V_i, V_i)}{L(V, V)} - \left(\frac{L(V_i, V)}{L(V, V)} \right)^2 \right] \tag{4}$$

According to the above function Q , we can see that the main idea of modularity comes from a comparison between real community partitions structure and network partitions allocated without any regard to the underlying structure. Then, sum over all the partitions differences of this two kinds of network structure.

4.2 Test on Real-World Networks

Test on Zachary’s Karate Club Network. Zachary’s karate club network [20] was generated by Zachary, who studied the friendship of 34 members of a karate club over a period of 2 years. In the course of research, he found a disagreement developed between the administrator and the instructor of karate club. Eventually, the club was divided into two groups almost of the same size. This network consists of 34 nodes and 78 edges.

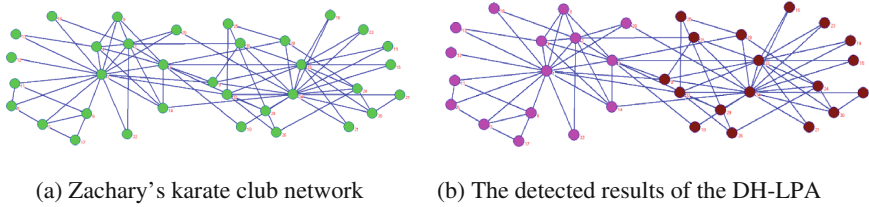


Fig. 2. Zachary’s karate club network and the detected results of the DH-LPA

Figure 2(b) shows the detected community partitions of the DH-LPA. We can see that two partitions are found, which is equal to the real partitions of the network. In Fig. 2(b), The value of modularity is 0.3715, and the NMI is 1, which illustrates the efficiency of DH-LPA on this network.

Test on Bottleneck Dolphin Network. Bottleneck dolphin network [21] describes a network of 62 bottlenose dolphins living in Doubtful Sound, New Zealand, was compiled by Lusseau after studying their behavior for 7 years. A tie between two dolphins was established by their statistically significant frequent association. The network split naturally into two large groups where the number of ties was 159.

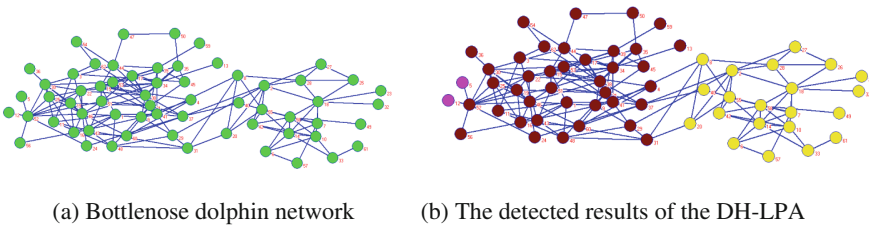
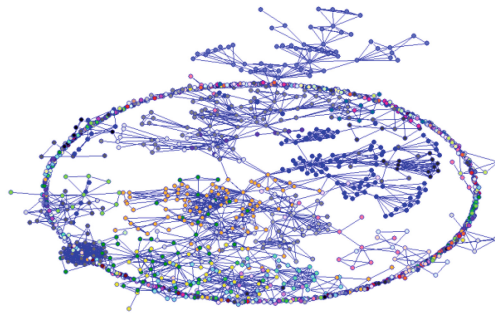


Fig. 3. Bottleneck dolphin network and the detected results of the DH-LPA

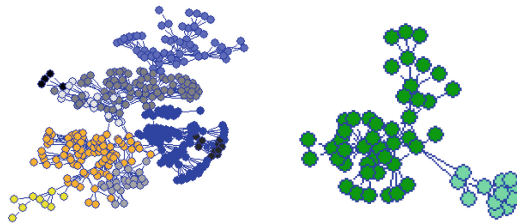
From Fig. 3(b), we can see that the DH-LPA found 3 communities, which is a little different from the real partitions of the network. The value of modularity is 0.3749, and the NMI is 0.8069, which is very close to real partitions. During the course of label propagation, we found that the former updated labels may be again relabeled by other

influential and frequent nodes' labels, such as the node 52. At the previous label propagation process, the node 52, 5 and 12 own the same label, but latter the label of the node 52 was relabeled by influential nodes, so we see this condition in Fig. 3(b).

Test on NetScience Network. NetScience network contains a co-authorship network of scientists working on network theory and experiment, as compiled by M. Newman in May 2006. The network was compiled from the bibliographies of two review articles on networks [22, 23], with a few additional references added by hand, which contains 1461 nodes and 2742 edges in total. This network is weighted, but we handle it as an unweighted one in our experiments.



(a) The complete detection partitions



(b) Bigger communities A and B found by the DH-LPA

Fig. 4. The detected results of the DH-LPA on NetScience network

As far as the NetScience network is concerned, the DH-LPA is also competent in detecting better communities. On this network, our algorithm obtains 277 community partitions with a big modularity value of 0.9541, which illustrates that our algorithm has found communities with strong structures. The complete detection results of the DH-LPA can be seen in Fig. 4(a). Figure 4(b) presents two bigger community partitions found by the DH-LPA.

4.3 Test on Synthetic Networks

In this section, we first test accuracy of the DH-LPA on synthetic benchmark network with a known community structure, so as to illustrate that our proposed algorithms can figure out the real community partitions. We use the Lancichinetti-Fortunato-Radicchi (LFR) benchmark networks proposed by Lancichinetti et al. [24] to evaluate the performance of the DH-LPA. By tuning the parameters of the networks, different benchmark network can be generated. This kind of generated networks is defined as $LFR(N, k, \max k, \mu, \min c, \max c)$. Where N is the number of nodes in network, k is the average degree of nodes, $\max k$ is the maximum degree of the nodes, μ is the mixing parameter, $\min c$ is the minimum for the community sizes, and $\max c$ is the maximum for the community sizes.

Table 2. The NMI values comparison of different algorithms on LFR Networks

No	$LFR(N, k, \max k, \mu, \min c, \max c)$	Nodes	Edges	NMI		
				DH-LPA	Fast Newman [17]	Danon [18]
1	(128,16,16,0.1,32,32)	128	1024	1.0000	1.0000	1.0000
2	(256,16,16,0.1,32,32)	256	2048	1.0000	0.9549	0.9549
3	(512,10,16,0.2,10,50)	512	2532	0.9182	0.6518	0.8262
4	(1000,20,50,0.2,40,50)	1000	10067	0.8286	0.8153	0.6153
5	(2000,20,50,0.2,50,60)	2000	20660	0.8828	0.7864	0.8566
6	(4000,30,50,0.2,50,100)	4000	59679	0.9042	0.7731	0.9257

We generated 6 different LFR benchmark networks according to Ref. [24]. In order to better illustrate the performance of different algorithms, we set different parameters for benchmark networks. The number of edges for each networks is presented according to LFR code, which contains bilateral edges. In Table 2, from network 1 to 6, the number of nodes is increased, and other parameters are also changed to improve the complexity of the benchmark network. From Table 2, we can see that DH-LPA obtains better results compared to Fast Newman and Danon algorithm in most conditions. For the network 6, the Danon method gets a higher NMI value than DH-LPA and Fast Newman. However, in our experiments, we find the DH-LPA can quickly generate community partition results for its nearly linear time complexity. In larger networks, such as the network 6, the Fast Newman and Danon algorithm run for nearly 50 min and then generated results, which illustrates their high time complexity.

5 Conclusion

In this paper, we put forward a community detection algorithm DH-LPA based on DH-index, and test the effectiveness of DH-LPA on 3 real world networks and the artificial benchmark networks. Meanwhile, we compare it with other algorithms in these networks. The experiment results show that the DH-LPA is very effective in

community detection problems. In our future work, we will pay attention to improve the efficiency of DH-LPA so as to make it work on dynamic networks, which is a very interesting work.

Acknowledgments. This work was supported by 973 Program of China (Grant No. 2013CB329601, 2013CB329604, 2013CB329606).

References

1. Fortunato, S.: Community detection in graphs. *Phys. Rep.* **486**, 75–174 (2010)
2. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U.S.A.* **99**(12), 7821–7826 (2002)
3. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Phys. Rev. E* **70**, 066111 (2004)
4. Xu, B., Deng, L., Jia, Y., Zhou, B., Han, Y.: Overlapping community detection on dynamic social network. In: *Proceedings of the 6th International Symposium on Computational Intelligence and Design*, Hangzhou, China, vol. 2, pp. 321–326 (2013)
5. Liu, Q., Zhou, B., Li, S., Li, A.P., Zou, P., Jia, Y.: Community detection utilizing a novel multi-swarm fruit fly optimization algorithm with hill-climbing strategy. *Arab. J. Sci. Eng.* **41**, 807–828 (2016)
6. Lin, W., Kong, X., Yu, P.S., Wu, Q., Jia, Y., Li, C.: Community detection in incomplete information networks. In: *21st International Conference on World Wide Web*, pp. 341–350. ACM Press (2012)
7. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**, 036106 (2007)
8. Barber, M.J., Clark, J.W.: Detecting network communities by propagating labels under constraints. *Phys. Rev. E* **80**, 026129 (2009)
9. Leung, I.X., Hui, P., Lio, P., Crowcroft, J.: Towards real-time community detection in large networks. *Phys. Rev. E* **79**, 066107 (2009)
10. Subelj, L., Bajec, M.: Unfolding communities in large complex networks: combining defensive and offensive label propagation for core extraction. *Phys. Rev. E* **83**, 036103 (2011)
11. He, M., Leng, M., Li, F., Yao, Y., Chen, X.: A node importance based label propagation approach for community detection. In: Sun, F., Li, T., Li, H. (eds.) *ISKE2012. Advances in Intelligent Systems and Computing*, vol. 214, pp. 249–257. Springer, Heidelberg (2014)
12. Sun, H., Liu, J., Huang, J., Wang, G., Yang, Z., Song, Q., Jia, X.: CenLP: a centrality-based label propagation algorithm for community detection in networks. *Phys. A Stat. Mech. Appl.* **436**, 767–780 (2015)
13. Wang, W., Street, W.N.: A novel algorithm for community detection and influence ranking in social networks. In: *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 555–560. IEEE Press, New York, August 2014
14. Kitsak, M., Gallos, L.K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H.E., Makse, H.A.: Identification of influential spreaders in complex networks. *Nat. phys.* **6**, 888–893 (2010)
15. Lü, L., Zhou, T., Zhang, Q.M., Stanley, H.E.: The H-index of a network node and its relation to degree and coreness. *Nat. Commun.* **7**, 10168 (2016)
16. Chen, D., Lü, L., Shang, M.S., Zhang, Y.C., Zhou, T.: Identifying influential nodes in complex networks. *Phys. A Stat. Mech. Appl.* **391**, 1777–1787 (2012)

17. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **69**(6), 066133 (2004)
18. Danon, L., Díaz-Guilera, A., Arenas, A.: The effect of size heterogeneity on community identification in complex networks. *J. Stat. Mech. Theory Exp.* **11**, P11010 (2006)
19. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. *J. Stat. Mech. Theory Exp.* **9**, P09008 (2005)
20. Zachary, W.W.: An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **33**, 452–473 (1977)
21. Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behav. Ecol. Sociobiol.* **54**, 396–405 (2003)
22. Newman, M.E.J.: The structure and function of complex networks. *SIAM Rev.* **45**, 167–256 (2003)
23. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: Structure and dynamics. *Phys. Rep.* **424**, 175–308 (2006)
24. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**, 046110 (2008)