Knowledge-Based Systems 187 (2020) 104822

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A three-stage algorithm on community detection in social networks[☆], [☆☆]

Xuemei You, Yinghong Ma [*], Zhiyuan Liu

Business School, Shandong Normal University, Shandong, 250014, PR China

HIGHLIGHTS

- The presented three-stage algorithm detects communities without knowing the number of them beforehand.
- The central nodes' identification completely depends on the node degree and the distance of nodes in networks.
- The number of communities is determined by the size of central nodes' set.
- The three-stage algorithm converges to global optimum because of the integrated local and global structure information in it.

ARTICLE INFO

Article history:
Received 12 August 2018
Received in revised form 17 March 2019
Accepted 27 June 2019
Available online 3 July 2019

Keywords:
Community detection
Three-stage algorithm
Central nodes identification
Label propagation
Communities combination

ABSTRACT

Detecting communities or clusters of networks is a considerable interesting problem in various fields and interdisciplinary subjects in recent years. Tens of hundreds of methods with significant efforts devoted to community detection in networks, while an open problem in all methods is the unknown number of communities in real networks. It is believed that the central node in a community might be highly surrounded by its neighbors and any two centers of the community reside far from each other, and also believed the similarity among nodes in the same community is larger than the others. Therefore, the local and the global structures' information shed important light on community detection. In this work, we present a three-stage algorithm to detect communities based on the local and the global information without giving the number of communities beforehand. The three stages include the central nodes identification, the label propagation and the communities combination. The central nodes are identified according to the distance between them larger than the average; the label propagation is to label nodes with the same colors when they reach to the maximum similarity; the communities combination is to merge two communities into one if the increment of the modularity is positive and maximum when the two communities were combined. Experiments and simulation results both on real world and synthetic networks show that the three-stage algorithm possesses well matched properties compared with seven other widely used algorithms, which indicates that three-stage algorithm can be used to detect community in social networks.

1. Introduction

Many complex systems in different areas can be modeled as networks or graphs in which the function units can be considered as nodes or vertices whose interactions are called edges. An important class of sub-network is communities which are characterized by the subgraphs of densely connected nodes and sparsely connected with other parts of the networks. Communities detection as a fundamental problem in the study of social network not only has important theory research significance in the areas of sociology, biology, electronic commerce, but also has practical applications in network security.

The community structures are closely associated with functions of specific networks, thus identifying such structures yields insights into the functional organization of the network. However, finding communities within an arbitrary network is a computationally difficult task. A growing number of community detection methods have been proposed since the significant work by Girvan and Newman [1]. One popular criteria is to optimize the modularity measure [2], like the Louvain algorithm [3] and the Fastgreedy algorithm [4]. Other methods involve machine

[☆] This work is supported by the National Natural Science Foundation of China (No.71701115, 71471106), Nature Science Foundation of Shandong Province (No.ZR2017MF058) and the Higher School Science and Technology Foundation of Shandong Province (No. J17KA172).

[☆☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.06.030.

[*] Corresponding author.
E-mail address: yinghongma71@163.com (Y. Ma).

https://doi.org/10.1016/j.knosys.2019.06.030
0950-7051/© 2019 Elsevier B.V. All rights reserved.

learning techniques, such as node2vec [5], seeding and semi-supervised learning [6] and low-rank subspace learning [7]. And neural network approaches [8], Bayesian [9] are also applied to community detection, and more recent developed methods see [10–12] and so on.

In general, community detection falls in the scope of clustering [13,14]. A key concept in clustering is using local structure indexes, such as node's similarities [15–17] to detect communities. However, these similarity measures usually do not take into account the global structures of the network, such as the distance among the important nodes. On the other hand, community detection designed with global structure information get the whole network's perspectives, such as Louvain, Fastgreedy, Infomap [18], Eigenvector [19], Label propagation (LPA) [20] and Node2vec. While the global structure information considered in algorithms often decreases the effectiveness.

In the previous methods on community detection, the algorithms based on global information including whole network structure characteristics guarantee the superior at the cost of high complexity. With the rapidly developed of information technology, the online social networks present large scale and dynamic characteristics, which take the community detection into complexity situations. While algorithms employing local structures information might fall into local optimum even though they have low time complexity. It is an interesting problem to balance the global and the local information, the accuracy and time complexity in designing algorithms.

In this paper, we take the local and global information together and propose a three-stage(TS) algorithm. The basic idea of TS algorithm includes three stages: identifying central nodes, label propagation and communities combination. Identifying the community centers and assigning proper community labels based on local density and relative distance are the global strategy in this stage; Diffuse the labels until all nodes are labeled according to the maximum similarity in the second stage; And merge the communities one by one until the modularity is not increased in the third stage. Simulations on real and synthetic data prove the accuracy and efficiency of TS algorithm. The application of TS algorithm is also settled by its basic idea, it can be applied to real business recommendation and precision marketing. And it is also suitable for friend recommendation in social networks based the small-world phenomenon. The arrangement of this paper is as follows. In Section 2, the proposed three-stage algorithm is presented in detail step by step. In Section 3, experimental results both on real world networks and synthetic networks show the efficiency and accuracy of three-stage algorithm compared with seven other classical methods. In Section 4, discussion on the influence of different similar indexes for TS algorithm is given. And finally is the conclusion and the further research.

2. Three-stage Algorithm

The networks in this paper are undirected and unweighted graph, denoted by $G = (V, E)$, where V and E are the vertex set and edge set respectively. Each vertex in G represents an element in the data set, and each edge is a relationship between a pair of elements. $n = |V|$ is the number of nodes and $m = |E|$ is the number of edges in network respectively. The network structure is represented as an adjacency matrix $A = (a_{ij})_{n\times n}$, where $a_{ij} = 1$ if an edge exists between nodes i and j; Otherwise $a_{ij} = 0$.

The main steps in the algorithm includes the central nodes identification, the label propagation and the communities combination. We describe the three-stage algorithm in detail in the following text.

2.1. The first stage: Central nodes identification

In the first stage, the key intuition is that the central node in a community might have highly surrounded by neighbors in this community, while neighbors of the central node may not connect tightly with each other. The number of the node's neighbors is named degree. Larger degree means that the node has more neighbors and therefore it has a high local degree. The node with larger degree is more likely to be a community center. Fig. 1(a) shows an example of a synthetic network and the nodes are ranked according to degree centrality, bigger circle with larger node degree.

On the other hand, the proverb "If two men ride on a horse, one must ride behind". In this view, the distance among central nodes might not be very close. So, we assume that the distance of two central nodes is not less than the average distance of the network. Hence, we detect the central nodes by their degree together with the distance of nodes.

Therefore, in the first stage, rank nodes with degree, and then choose the central nodes according to the average distance of nodes.

The average distance of graph G is denoted by D, $D = \frac{2}{n(n-1)} \sum_{u,v\in V} d(u, v)$, where $d(u, v)$ is the distance defined as the shortest path between u and v.

In this stage, rank all nodes in V via their degrees. Denote the node rank by $\{v_1, v_2, \dots, v_n\}$ when $d(v_1) \geq d(v_2) \geq \cdots \geq d(v_n)$. Calculate the average distance D of G, set $C_0 = \{v_1\}$ be the initial central nodes set.

For $v_j \notin C_0$, if

$$d(v, v_j) \geq D, \text{ for any } v \in C_0. \tag{1}$$

Then update C_0 by $C_0 \cup \{v_j\}$. Go on this operation until all nodes in $u \in V - C_0$ satisfy $d(v, u) < D$. At this time, the central node set is detected. Denote the central node set by $C_0 = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}$.

A synthetic network shows how the algorithm works for detecting communities, Fig. 1(a), is described step by step as above. Fig. 1(b) shows 6 community centers nodes $C_0 = \{50, 44, 37, 30, 26, 11\}$ are got by this stage.

2.2. The second stage: Label propagation

After identifying the central nodes at the first stage, we will partition the remained nodes such that the nodes in the same part have the most similarity.

Here, we take Sϕrensen similarity as the measurement, which is the comprehensively best compared to other similarity indexes mentioned in discussion section, and Sϕrensen index is a statistic used for comparing the similarity of two samples. Here we take the neighbors and the degree as the statistic to measure the similarity of two nodes:

$Sim(u, v) = \frac{2|N(u)\cap N(v)|}{d(u)+d(v)}$, where $N(u)$ is the neighbor set of u.

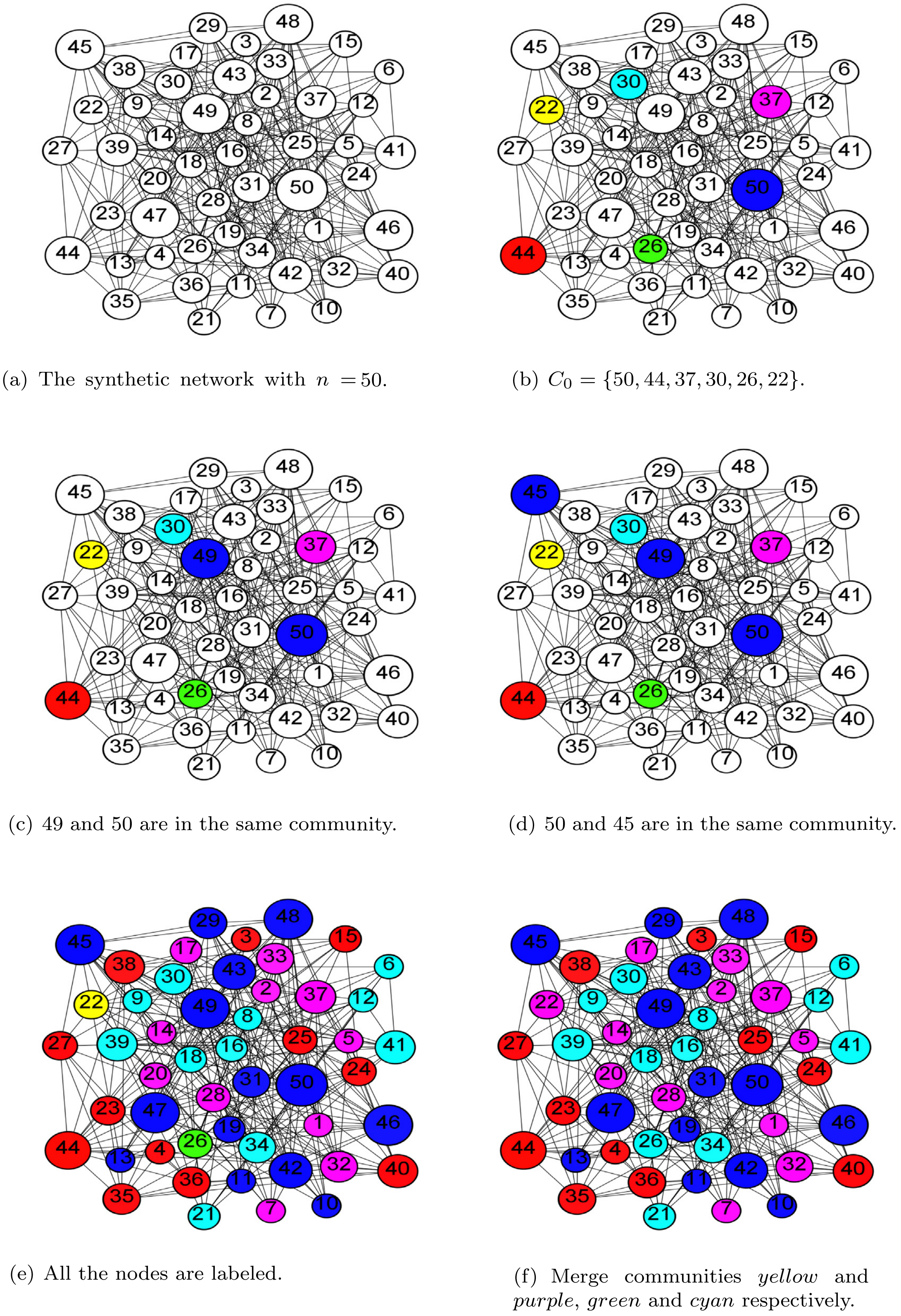
In this stage, generate the initial community set at first.

Let the initial community be $\{C'_1, C'_2, \dots, C'_k\}$, where $C'_j = \{v_{i_j}\}$ and $v_{i_j} \in C_0$ for $j = 1, \dots, k$. That is to say, a central node corresponds to a community.

At the second step, label the node $v \in V - C_0$ the same color as the node $u \in C_0$ if v is the neighbor of u and satisfy

$$sim(u, v) = \max_{v_i\in C_0} \max_{v_j\in N(v_i)} Sim(v_i, v_j). \tag{2}$$

Then label v the same color as u. If $u \in C'_{j'}$, then update $C'_{j'}$ by $C'_{j'} \cup v$. Going on this process until all nodes in $V - C_0$ are colored. Thus, set $PreC = \{C'_1, C'_2, \dots, C'_k\}$ be the pre-community. In this step, there are two cases need to set when the node v is to label. One case is that there are more than one neighbors of v, say nodes u_1 and u_2, having the maximal similarity with the



(a) The synthetic network with $n = 50$.

(b) $C_0 = \{50, 44, 37, 30, 26, 22\}$.

(c) 49 and 50 are in the same community.

(d) 50 and 45 are in the same community.

(e) All the nodes are labeled.

(f) Merge communities *yellow* and *purple*, *green* and *cyan* respectively.

Fig. 1. The 6 panels show the example of the synthetic networks and results of the algorithm Table 4 runs. The size of the node is proportional to the degree of it. That is, the larger degrees the bigger nodes. (a) is the synthetic network with $n = 50$. (b) is the visualization of the central nodes identification stage. There are six central nodes in C_0. (c) to (e) are the parts of visualization of the label propagation stage. There are six preliminary communities $PreC = \{yellow, green, cyan, red, blue, purple\}$. And (f) is the community combination stage. The final community set is $C = \{cyan, red, blue, purple\}$.

node v, and u_1 and u_2 belong to two different communities C_1 and C_2 respectively. Then, take v into C_1 or C_2 randomly, and update C_1 or C_2 by the choice of the algorithm. For example, the nodes 50 and 44 are the neighbors of 45, and they have the maximum similarity, so the node 45 is randomly marked as blue or red. Another case is on the contrary, there are more one nodes, say v_1

Table 1
The neighbors having the maximum similarity with nodes in C_0.

Nodes	Neighbors	$\max Sim(u, v)$
50	49	0.83
44	38	0.75
37	32	0.67
30	39	0.76
26	34	0.53
22	32	0.53

Table 2
The neighbors having the maximum similarity with nodes in $C_0 \cup \{49\}$.

Nodes	Neighbors	$\max Sim(u, v)$
50	45	0.83
49	45	0.79
44	38	0.75
37	32	0.67
30	39	0.76
22	32	0.53
26	34	0.53

Table 3
The increase of the modularity of each pair of communities in *PreC*.

$\triangle Q$	Blue	Red	Purple	Cyan	Yellow	Green
blue	0	−0.1252	−0.1069	−0.1111	−0.0127	−0.0127
red	−0.1252	0	−0.0747	−0.0903	−0.0091	−0.0091
purple	−0.1069	−0.0747	0	−0.0719	0.0265	−0.0031
cyan	−0.1111	−0.0903	−0.0719	0	−0.0033	0.0262
yellow	−0.0127	−0.0091	0.0265	−0.0033	0	−0.0007
green	−0.0127	−0.0091	−0.0031	0.0262	−0.0007	0

$\triangle Q$ is a symmetric matrix.

Table 4
Three-stage algorithm.

```
Input: An undirected and unweighted network $G = (V, E)$.
Output: The Communities $C = \{C_1, C_2, \dots, C_t\}$.
Stage 1: Central nodes identification.
Stage 1.1: Ranking all nodes $\{v_1, v_2, \dots, v_n\}$ by their degree decreasing.
Stage 1.2: Let $C_0$ be the initial central nodes set and $v_1 \in C_0$.
    Updating $C_0$ by $C_0 \cup v_j$ if $v_j \in V - C_0$ and satisfies Eq. (1).
    Until no any node in $V - C_0$ such that the Eq. (1) holds, stop.
    Denote the final $C_0$ by $C_0 = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}$.
Stage 2: Label propagation.
    Let $\{C'_1, \dots, C'_k\}$ be the initial communities, $C'_j = \{v_{i_j}\}$ and $v_{i_j} \in C_0$.
    For $u \in C'_{j'}$ and $v \in V - C_0$, if $Sim(u, v)$ satisfies Eq. (2),
    Then update $C'_{j'}$ by $C'_{j'} \cup \{v\}$ and $C_0$ by $C_0 \cup \{v\}$.
    Go on this process until $V - C_0 = \emptyset$. Set $PreC = \{C'_1, C'_2, \dots, C'_k\}$.
Stage 3: Community combination.
    Merging $C'_{i_0}$ and $C'_{j_0}$ into one if they satisfy equation (3).
    Updating $C'_{i_0}$ and $C'_{j_0}$ by $C'_{i_0} \cup C'_{j_0}$ in $PreC$.
    Repeat this process until the modularity is no longer increased.
Output the final community set $C = \{C_1, C_2, \dots, C_t\}$.
```

and v_2, having the maximal similarity with the node in the same community. In this case, v_1 or v_2 will be randomly selected and colored with the same color with the nodes in community. For example, nodes 50 and 49 are in the same community, in the blue, and having the same maximum similarity with node 45 and 31. Then the node 45 is randomly chosen and colored blue. The node 31 would take into the next iteration. The remaining nodes will be taken into their own communities in the rest of the iteration.

In the example of Fig. 1(b), let the initial communities be $\{yellow, green, cyan, red, blue, purple\}$, where $yellow = \{22\}$, $green = \{26\}$, $cyan = \{30\}$, $red = \{44\}$, $blue = \{50\}$ and $purple = \{37\}$. The maximum similarity neighbors of center nodes in $C_0 = \{50, 44, 37, 30, 26, 11\}$ are shown in Table 1.

Clearly, nodes 50 and 49 have the largest similarity among the six maximal similarities, so the node 49 will be detected and labeled as the same color with the node 50 as shown in Fig. 1(c). After the node 49 is labeled, update the community $blue = \{50\}$ by $blue = \{50, 49\}$. Then go on checking the unlabeled neighbors of nodes in communities $\{yellow, green, cyan, red, blue, purple\}$. The node 45 has the largest similarity with the node 50, as shown in Table 2. So the node 45 is taken into the community *blue*, and again updated $blue = \{50, 49\}$ by $\{50, 49, 45\}$, see Fig. 1(d).

The remaining unlabeled nodes are labeled in the same way as above until all nodes are labeled. The result in Fig. 1(e) shows a partition of the network in which the same color nodes are in one community. There are six parts and forms the preliminary communities, named pre-community, $PreC = \{yellow = \{22\}$, $green = \{26\}$, $cyan = \{6, 8, 9, 12, 21, 18, 16, 30, 34, 39, 41\}$, $red = \{3, 4, 15, 23, 24, 25, 27, 35, 36, 38, 40, 44\}$, $blue = \{10, 11, 13, 19, 29, 31, 42, 43, 45, 46, 47, 48, 49, 50\}$, $purple = \{1, 2, 5, 7, 14, 17, 20, 28, 32, 33, 37\}\}$.

2.3. The third stage: Community combination

Obviously, it is weird that the sets *yellow* and *green* are one node communities in pre-community, moreover, nodes 22 and 26 have relative large degree. So in the third stage, we optimize the pre-community.

Modularity is introduced as evaluation index to measure the quality of the community structure in networks. It serves as the objective function during the process of calculating the communities [21]. Higher values for the modularity Q mean better community structures. Therefore, the object is to find the community assignment for each node in the network such that Q is maximized. Hence, in the third stage, the optimization function is to merge two communities C'_{i_0} and C'_{j_0} in pre-community into one $C'_{i_0} \cup C'_{j_0}$ such that the increase of modularity $\triangle Q$ satisfy

$$\triangle Q(C'_{i_0} \cup C'_{j_0}) \geq \max\{\triangle Q(C'_i \cup C'_j), 0\} \text{ for any } C'_i, C'_j \in PreC. \quad (3)$$

Where $Q = \frac{1}{2m} \sum_{v_i, v_j} (a_{ij} - \frac{d(v_i)d(v_j)}{2m}) \delta(C_i, C_j)$, a_{ij} is the element of adjacent matrix, $d(v_i)$ is the degree of v_i. C_i is the community in which vertex v_i belongs to, $\delta(C_i, C_j)$ is an indicator function, $\delta(C_i, C_j) = 1$ if $C_i = C_j$; Otherwise 0. $m = \frac{1}{2} \sum_{v_i \in V} d(v_i)$.

Any two communities are merged together to ensure the modularity Q is growing to be the largest. Repeat this process until the modularity is no longer increased. At this moment, the communities merging process stopped, and the optimal community is formed.

There are 6 communities $PreC = \{yellow, green, cyan, red, blue, purple\}$ in Fig. 1(e), then the modularity of each pairs of *PreC* are shown in Table 3:

It is found in Table 3, the maximum increase is the merging of communities *purple* and *yellow*. Hence, merge *purple* and *yellow* into *purple*, and update *PreC* by {green, cyan, red, blue, purple}. Therefore, we go on this process, until any two communities in *PreC* do not increase. Fig. 1(f) shows the 4 communities, $\{cyan, red, blue, purple\}$, are produced. During the merging, the $yellow = \{22\}$ and $green = \{26\}$ are merged by *purple* and *cyan* respectively. The final community is $\{cyan, red, blue, purple\}$.

2.4. Three-stage algorithm and its complexity

Combining the central nodes identification, the label propagation and the communities merger stages together, and the three-stage algorithm is presented in Table 4.

The community detection algorithm consists of three parts: centers identifying, label propagating and communities merging. In the first stage, it generally takes $O(n^2)$ time in sorting node's degree, and $O(n^3)$ time in calculating the average distance of the network. In the second stage, label propagation, the process of computing the similarity between the neighbor, the time cost is $O(n\langle k\rangle^2)$, where $\langle k\rangle$ is the average degree of the network. And in the last stage, the merge process begins with the k pre-communities, and merging the k pre-communities into t communities, $(t \le k \le n)$. It takes $O(k^2)$ when the modularity comparison on the combination of k pre-communities are computed.

In summary, the algorithm's complexity costs $O(n^3+nlog(n)+O(k^3)) \approx O(n^3)$.

3. Experimental results

We test the performances of three-stage algorithm on both synthetic and real-world networks by comparing the outcomes of TS algorithm with the ground-truth community structures and results of other community detection methods.

3.1. Synthetic networks

In the experiment, synthetic networks are LFR [22] benchmark networks. LFR networks have power-law distributions for both node's degree and communities' size, and symptom the features of real-world networks. Therefore, LFR networks are appropriate to be used to evaluate the performance of community detection algorithms.

Generally, there are 8 structural parameters for LFR networks: the size of network n, the average degree $\langle k\rangle$ and the upper bound of degree k_{max}, the power-law exponents for the size of communities and the nodes' degree β and α, the maximum and the minimum size of communities $maxc$ and $minc$, the mixing parameter μ.

Among all the parameters, the mixing parameter μ is the fraction of links connecting each node in a community to nodes in the other communities to the total degree of nodes. $\mu = \frac{\sum_C |E(C,V-C)|}{\sum_{v\in V} d(v)}$, where $E(C, V-C)$ is the edges between C and the other nodes except in C, and $|\cdot|$ is the size function. By the definition of μ, it is easy to find μ displays the ratio of edges intra communities to the total. The higher of μ means the more ambiguous community structures. For each set of parameters, 20 networks are generated. The parameters of LFRs are set as follows.

- The number of nodes n: $n = 50, 500, 1000$ and 5000 respectively.
- The average degree $\langle k\rangle = 5, 15, 20, 20$ corresponding to $n = 50, 500, 1000, 5000$ and the upper bound of degree $k_{max} = 0.1n$.
- The power-law exponents for the size of communities is $\beta = 1$ or 2 respectively.
- The power-law exponents for the nodes' degree is $\alpha = 2$.
- The maximum size for communities is $maxc = 0.1n$ and the minimum is $minc = 5, 10, 10, 10, 20$ corresponding to $n = 50, 500, 1000, 5000$.
- The mixing parameter μ is set $\mu = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7$.

To test the performance of the three-stage algorithm, we compare the results of the partitioning with the other seven popular algorithms: Louvain, Fastgreedy, Infomap, Eigenvector, Label propagation (LPA), Walktrap and node2vec. NMI index [23] and modularity function [23] are used to evaluate the efficient and accuracy.

Before the experiment on LFR synthetic networks, we introduce the NMI index, normalized mutual information, to evaluate the efficiency of the three-stage algorithm, and also compare the accuracy with the other seven algorithms.

MI measures how much knowing one of these variables reduces uncertainly about the other. The normalized mutual information (*NMI*) is usually measures the similarity between the true community structures and the detected ones in networks:

$$NMI(X, Y) = \frac{2I(X, Y)}{H(X) + H(Y)}, \tag{4}$$

where $I(X, Y)$, the mutual information, measures the information shared by variables X and Y, $H(X)$ is the entropy of community of X. For example, if X and Y are independent, then knowing X does not given any information about Y and vice versa, so $NMI(X, Y) = 0$. At the other extreme, if X and Y are deterministic for each other, then all information covered by X is shared with Y and vice versa, so $NMI(X, Y) = 1$.

Fig. 2 shows the results of TS algorithm comparing with seven other algorithms on LFR synthetic networks comparison with *NMI*. Panels in Figs. 2(a) to 2(f) show that values of NMI decrease with increasing networks' size n and the mixing parameter μ respectively. It is because the greater μ means the more ambiguous community structures. Therefore, it gets more difficult to detect accurate communities as μ increases.

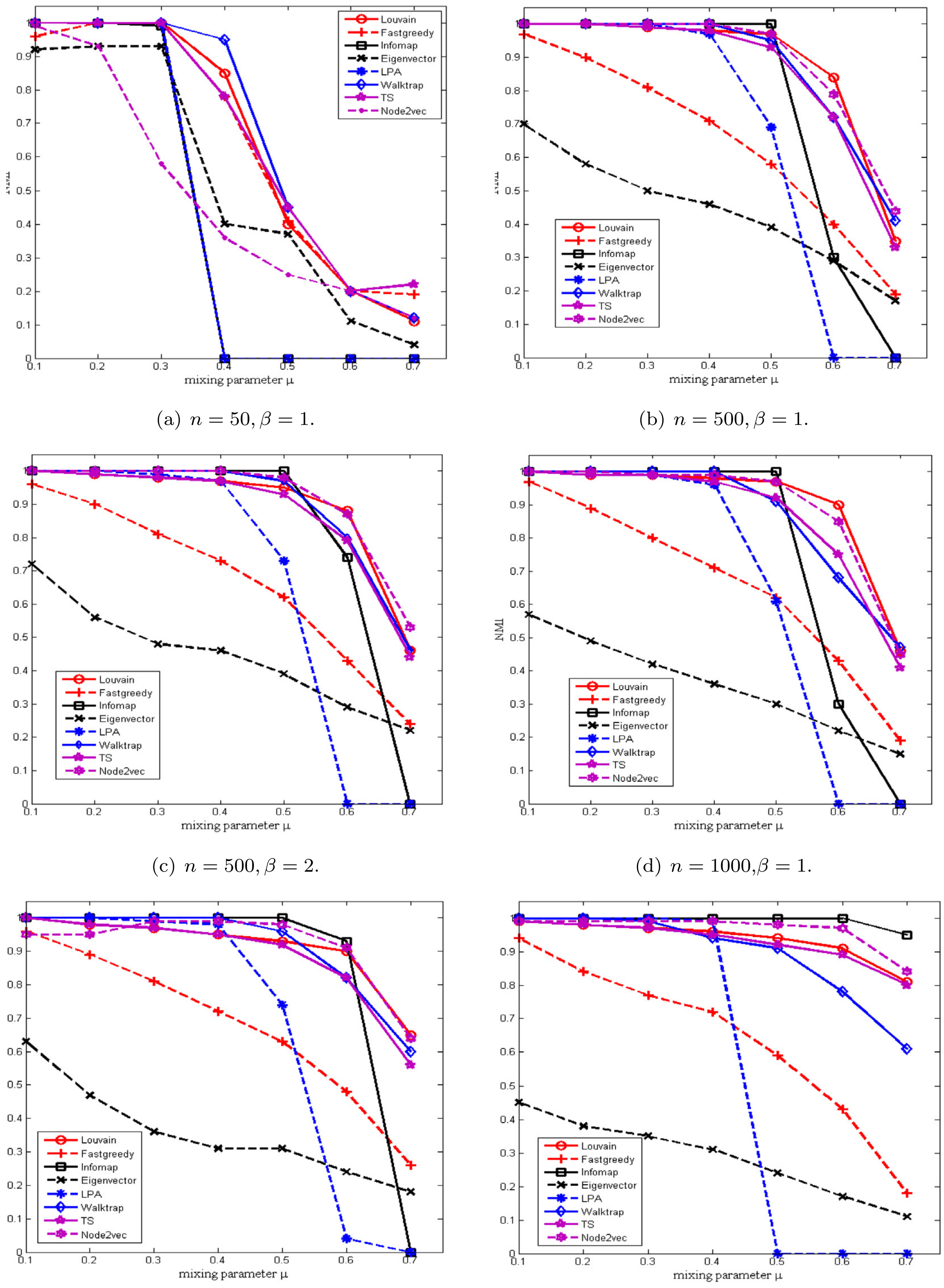
Panel 2(a) shows the results for LFR networks with nodes number $n = 50$. By the generation of LFR networks, the average degree $\langle k\rangle = 5 = k_{max} = 0.1n = 5$, and $maxc = 0.1n = minc = 5$. It is very uniform networks. It is not hard to understand the behaviors of all algorithms. When $\mu \le 0.3$, except for the fastgreedy, Eigenvector and node2vec methods, all other algorithms' results have little difference, and almost reach to 1. Node2vec's performance drops rapidly at $\mu \ge 0.2$. When $\mu \ge 0.3$, LPA begin to have a sharp decrease and then fall to 0 since $\mu = 0.4$. Infomap's performance is steady when $\mu \le 0.6$, but it radically declined to 0 when $\mu \ge 0.7$. As to Louvain, Walktrap and three-stage have little difference, TS algorithm has the best result when $\mu = 0.7$.

Panels 2(b) and 2(c) show the results for LFR networks with nodes number $n = 500$ and the exponent of the size of communities distribution $\beta = 1, 2$ respectively. The average degree $\langle k\rangle = 15$, $k_{max} = 0.1n = 50$, and $maxc = 0.1n = 50$, $minc = 10$. The difference between Panels 2(b) and 2(c) are the distribution of community's size, the former exponent is $\beta = 1$ and the later is 2. Even though the communities structure in this two cases might have a big diversity, the values of three-stage algorithm together with the other seven ones in this two panels display much similarities. When $\mu \le 0.5$ all results except the fastgreedy and Eigenvector have little difference, and almost reach to 1. When $\mu \ge 0.5$, LPA has a sharp decrease and falls to 0. Infomap's performance keeps steady at $\mu \le 0.6$ and then radically declines to 0. As to Louvain, Walktrap, node2vec and three-stage have little difference.

Panels 2(d) and 2(e) show the results for LFR networks with $n = 1000$ and the topological parameters are $\langle k\rangle = 20$, $k_{max} = 0.1n = 100$, and $maxc = 0.1n = 100$, $minc = 10$. Even though the communities structure in this two cases might have a big diversity since the exponent of the size of communities distribution $\beta = 1$ and 2 respectively. But the algorithms display similar behaviors to Panels 2(b) and 2(c).

Panel 2(f) is the results for $n = 5000$. The topological parameters have much more diversity than panel 2(d). Louvain, Infomap, node2vec and three-stage algorithms still perform good. Others have similar trends as above panels.

Fig. 2 is the results of algorithms on LFR synthetic networks with given topological parameters. The two pairs of panels, panels 2(b) and 2(c), panels 2(d) and 2(e), make some sense: The first one is the size of the networks does not decide the accuracy and effectiveness of algorithms; The second is the distribution of the size of communities does not distinguish the performance the



(a) $n = 50, \beta = 1.$

(b) $n = 500, \beta = 1.$

(c) $n = 500, \beta = 2.$

(d) $n = 1000,\beta = 1.$

(e) $n = 1000, \beta = 2.$

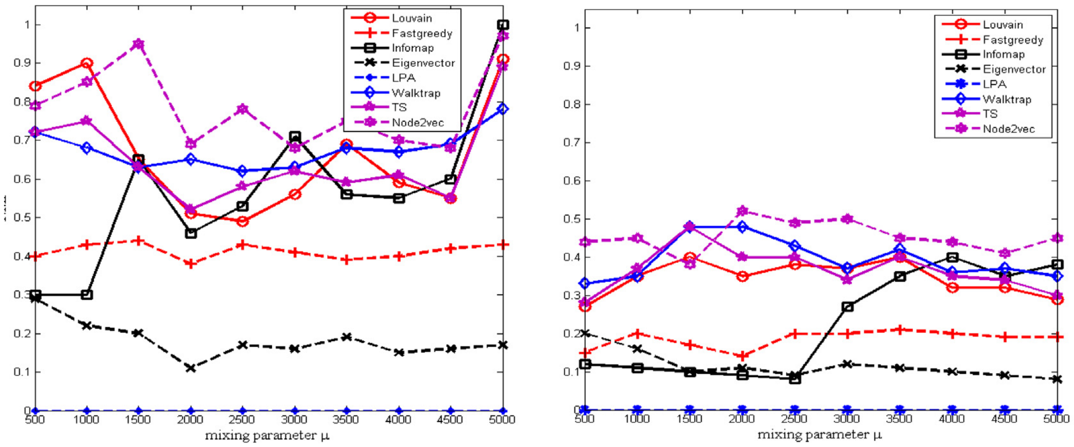
(f) $n = 5000, \beta = 1.$

Fig. 2. Comparisons of seven algorithms, Fastgreedy, Informap,Eigenvector, LPA, Walktrap and Louvain and Node2vec with the TS algorithm on LFR networks with different network size. The values of *NMI* are averages over 20 realization of each networks, and the similarity index is Sϕrenson.

algorithms; And the third is most algorithms do not work well with the increase of μ.

Panels in Figs. 2(a) to 2(f) show that values of NMI decreasing with increasing networks' size *n* and the mixing parameter μ respectively. It is because the greater μ means the more ambiguous community structures. Therefore, it get more difficult to detect accurate communities as μ increases.

In Fig. 2, algorithms begin to lose efficacy with μ growing. It seemed that the parameter μ affects algorithms, so we compare the feasibility and efficiency of algorithms when networks get more ambiguous. Fix $\mu = 0.6$ and 0.7, increase the size of networks from 500 to 5000 with interval 500. The results are shown in Fig. 3. Panel 3(b) shows the results when $\mu = 0.6$. Three-stage algorithm, Infomap, Louvain and node2vec achieve better results than others in different network scales. For $n =$



(a) Networks with $\mu = 0.6$ and $n = 500$ to 5000.

(b) Networks with $\mu = 0.7$ and $n = 500$ to 5000.

Fig. 3. Comparing the effectiveness of algorithms when the size of LFR networks dynamic for fixed $\mu = 0.6$ and $\mu = 0.7$. The values of *NMI* are averages over 20 realization of each networks.

1000, 2000, 2500 and 4000, three-stage algorithm is better than Infomap, Louvain. The performance of the node2vec algorithm is quite good. Walktrap's results are relatively good and steady. Fastgreedy and LeadingEigen's results are not quite well, and LPA cannot detect any communities in this test. Panel 3(b) is the result when $\mu = 0.7$. Three-stage algorithm, Louvain and walktrap achieve better results than others.

The pair of panels in Fig. 3 reveal that the robustness of the algorithm: The robust of three-stage algorithm is close to Louvain, Walktrap and node2vec, and they achieve similar performances whatever the networks topology are. In other words, μ does not affect algorithms directly.

3.2. Real-world networks with ground truth

Four real-world networks with undirect and unweighed links shown in Table 5 with ground truth are used to test the efficiency and accuracy of our algorithm, also compared with the above seven algorithms.

Zachary karate club network [24] is a famous social network, in which 34 members a karate club and 78 links between pairs of members who interacted outside the club. Some members formed part of a small group around the coach, other members chose a new coach, and the last part of the members gave up karate. So the different choices led to three divisions of the community, and finally two communities in Zachary Karate are formed. The second social network is dolphin network [25] which is formed by the frequency of the bottlenose dolphins played together. There are 62 dolphins with 159 associations in the dolphin network. The groups of dolphins are mainly divided into two communities: the male and female. The third is Polbooks network which is the political books' data on US politics recorded in 2005 by Adamic and Glance [26]. The network is an undirected graph object with 105 books and 882 links between them, and there is three groups. The fourth network is the American college football [27], it is known as the validating community detection algorithms.

We run three-stage algorithm on the four real networks, the detected community structures for each network with ground truth are visualized in Fig. 5, each color is a community in each panel. The gray lines in panels 5(a) and 5(b) divide the network into communities by three-stage algorithm. The values of modularity and *NMI* of three-stage algorithm together with the other seven algorithms are shown in Table 7 respectively. The proposed three-stage algorithm displays almost perfect performances both of modularity and *NMI* on the four networks.

Table 5
Real-world Networks with ground truth for experiments.

Networks	n	$\langle k \rangle$	Description
Karate	34	4.59	Zachary's social network of a karate club [24]
Dolphins	62	5.13	Dolphin social network [25]
Polbooks	105	8.40	Books about US politics [26]
Football	115	10.66	Network of American football games [27]

There are four visualized communities on Zachary Karate network in Fig. 5(a). If the red and the green communities, the blue and the purple communities are combined to two new communities respectively, it is the ground truth of Zachary Karate. Fig. 5(b) shows detected communities on Dolphin network. There are two communities and it is similar to the original partition of the network. The only difference is the assignment of the node 40 because three-stage algorithm gets a little modularity value compared with Louvain, Fastgreedy, Infomap, node2vec, Eigenvector and Label propagation, as shown in Table 7. Fig. 5(c) visualizes American college football network. The original divisions are 12 communities. While three-stage algorithm detects 10 communities. NMI value is 0.6 better than any of other methods as shown in Table 7. There are 3 communities in US political book network by ground truth. While 4 communities are detected using three-stage algorithm. For LPA and Eigenvector algorithms, their NMI values are all bigger than ours. However, our method can outperform them in modularity value.

From Table 7, we can see that for networks with ground truth communities, three-stage algorithm achieves good performance compared with other algorithms using *NMI* and *Q* criterions.

Fig. 4 summarizes the values of *NMI* and modularity for algorithms in previous four real networks. In terms of *NMI*, the three-stage algorithm produces good results compared to other methods, especially for Karate and Dolphin network. In terms of modularity, three-stage algorithm performs better than the others methods in karate and football networks. As well as for the other networks, three-stage algorithm remains competitive.

As mentioned above, the proposed approach deals with undirected and unweighted networks. The proposed algorithms are scalable and deterministic. By the comparison with the other algorithms, the three-stage algorithm improves the modularity and NMI values greatly.

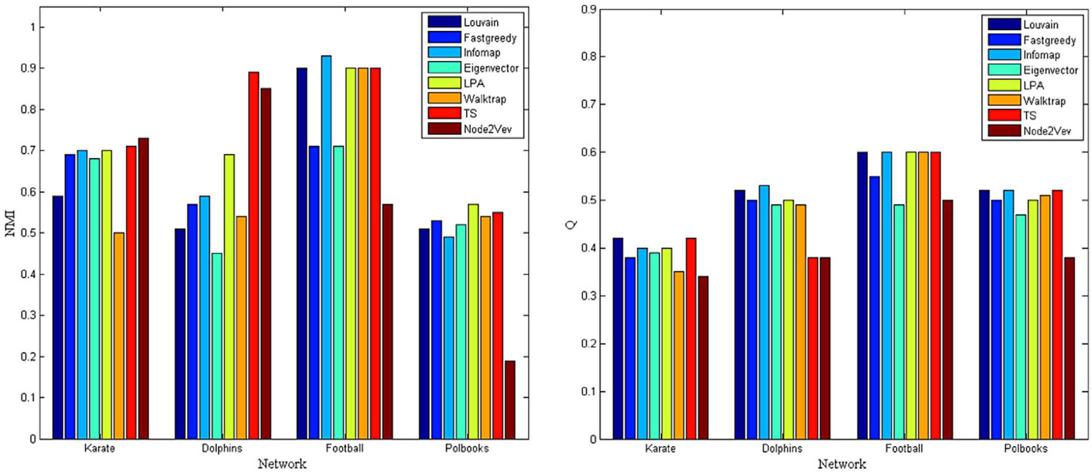
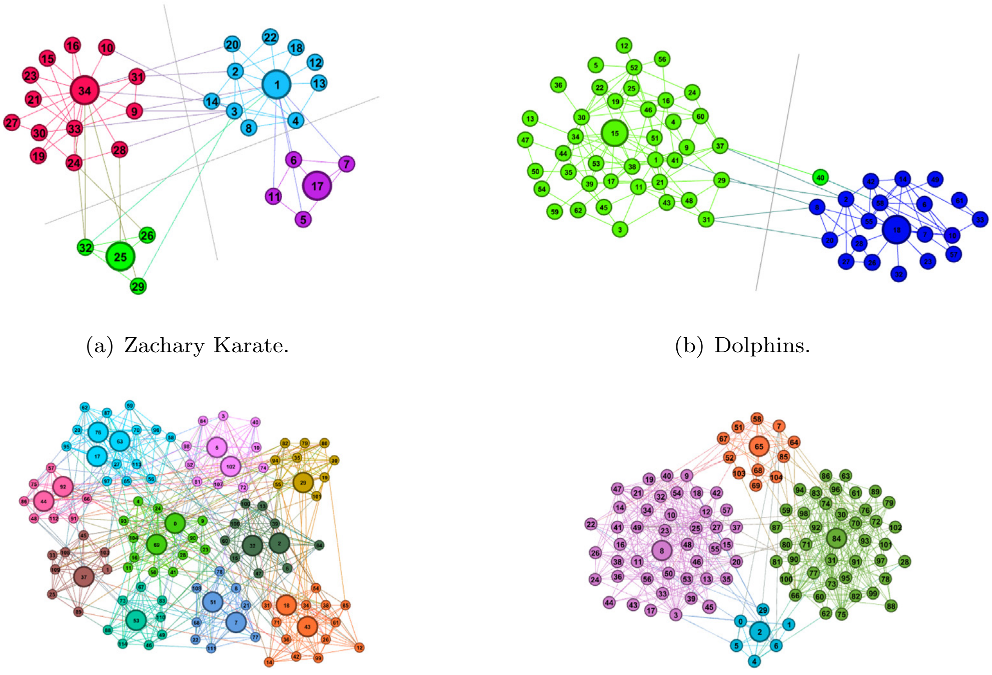


Fig. 4. NMI and modularity results for the algorithm TS in the networks with ground truth.



(a) Zachary Karate.

(b) Dolphins.

(c) Football.

(d) US Political Books.

Fig. 5. The visualizations of communities detected by three-stage algorithm.

3.3. Real-world networks without ground truth

Six different real-world networks without ground truth with different scales, shown in Table 6, are all analyzed by three-stage algorithm as well as seven other algorithms as mentioned above. For those networks, modularity is used to measure the quality of community detection results.

The Lesmis network is a co-appearance network of characters in the novel Les Miserables, as complied by Donald Ervin Knuth. The Adjnoun network is compiled by Newman, which is the adjacency network of common adjectives and nouns in the novel David Copperfield by Charles Dickens. The Jazz network is the collaboration network between Jazz musicians. Each node is a Jazz musician and an edge denotes that two musicians have played together in a band. The email network is an email communication network at the University Rovira i Virgili in Tarragona in the south of Catalonia in Spain. Nodes are users and edges indicate that at least one email was sent. The Polblogs is a directed network of hyperlinks between blogs on US politics, recorded in 2005 by Adamic and Glance, and the directions on edges are omitted in this section. The PowerGrid is an undirected, unweighted network representing the topology of the Western States Power Grid of the United States, compiled by D. Watts and S. Strogatz.

The values of modularity and the number of communities of three-stage algorithm together with the other seven algorithms are shown in Table 8 respectively. The three-stage algorithm displays good performances both of modularity and the number of communities on the six networks.

Three-stage algorithm detects Lesmis network into six communities $C_1 = \{$ 49,56,58,59,60,61,62,63,64,65,66,67,68,74,75,77$\}$, $C_2 = \{$13,17,18,19,20,21, 22, 23, 24, 31, 32 $\}$, $C_3 = \{$ 25,26,41, 42,43, 69,70,71, 72,76 $\}$, $C_4 = \{$ 27,40,44,50, 51,52, 53,54, 55,57, 73$\}$, $C_5 = \{47, 48\}$, and C_6 is the remaining 27 nodes. The maximum size of community is 27, the minimum is 2, and the

Table 6
Six real world networks without ground truth.

Networks	n	$\langle k \rangle$	Description
Lesmis	77	6.60	Network of characters in Victor Hugo's novel "Les Miserables" [28]
Adjnoun	112	7.59	Network of common adjective and noun adjacent in novel "David Copperfield" [19]
Jazz	198	27.70	Network of Jazz musicians [29]
Email	1133	9.62	Network of e-mail interchanges [30]
Polblogs	1222	27.36	Blogs about politics [26]
PowerGrid	4941	2.67	The Western States Power Grid of the United States [31]

Table 7
Performances comparison in the networks with ground truth.

Networks	Ground truth		TS			Louvain		Fastgreedy		Infomap		LPA		Eigenvector		Walktrap		Node2Vec	
	C	Q	C	NMI	Q	NMI	Q	NMI	Q	NMI	Q	NMI	Q	NMI	Q	NMI	Q	NMI	Q
Karate	2	0.37	4	0.71	0.42	0.59	0.42	0.69	0.38	0.70	0.40	0.70	0.40	0.68	0.39	0.50	0.35	0.73	0.34
Dolphins	2	0.38	2	0.89	0.38	0.48	0.52	0.61	0.50	0.50	0.52	0.69	0.50	0.45	0.49	0.54	0.49	0.85	0.38
Polbooks	3	0.41	4	0.55	0.52	0.51	0.52	0.53	0.50	0.49	0.52	0.57	0.50	0.71	0.49	0.54	0.51	0.57	0.50
Football	12	0.55	10	0.90	0.60	0.88	0.60	0.70	0.55	0.92	0.60	0.92	0.60	0.52	0.47	0.90	0.60	0.19	0.38

Table 8
Performance comparison in the networks without ground truth.

Networks	TS		Louvain		Fastgreedy		Infomap		Eigenvector		LPA		Walktrap		Node2Vec	
	C	Q	C	Q	C	Q	C	Q	C	Q	C	Q	C	Q	C	Q
Jazz	3	0.44	4	0.44	4	0.44	7	0.28	3	0.39	2	0.28	11	0.44	2	0.29
Email	9	0.55	12	0.54	16	0.51	68	0.52	7	0.49	8	0.28	49	0.53	141	0.33
Adjnoun	7	0.29	7	0.29	7	0.29	2	0.01	10	0.24	1	0.00	25	0.22	2	0.01
Power Grid	36	0.93	40	0.93	39	0.93	483	0.82	35	0.83	479	0.81	364	0.83	49	0.40
Polblogs	10	0.42	2	0.42	2	0.43	2	0.42	2	0.42	2	0.43	2	0.43	2	0.42
Lesmis	6	0.54	6	0.56	5	0.50	9	0.55	8	0.53	6	0.55	8	0.52	7	0.31

C is the number of communities and Q is the modularity.

others are about tens. And the number of communities matches to Louvain's and LPA's results. For the modularity of Lesmis network, Infomap and LPA achieve the same modularity value, and there is not much difference for all methods. For Adjnoun and Jazz networks, there are 7 and 3 communities by three-stage algorithm respectively, see Tables 9 and 10.

Table 8 shows Louvain, Fastgreedy and three-stage algorithm achieve good results with relative small number of communities in Adjnoun network. And in Jazz network, three-stage algorithm get 3 communities which close to Louvain and Fastgreedy algorithms both on the number of communities and the modularity. Fig. 6 show communities and their nodes. In Email network, the three-stage algorithm performs the best, 9 communities and modularity is 0.55. LPA hardly detects its community structures. There is not much difference for all methods. All algorithms except three-stage algorithm get 2 communities in Polblogs network. All algorithms have little difference in modularity values. For PowerGrid network, the three-stage algorithm achieves good results with relative small number of communities. The other four algorithms get community numbers that are almost 10 times than that of Louvain, Fastgreedy, node2vec and our three-stage algorithm. Together, Tables 8, 9, 10 and Fig. 6, it is observed that for some networks, such as Jazz, Email, Adjnounil, Power Grid, three-stage algorithm obtain the best results than other algorithms. But for some other networks, such as Lesmis, the modularity obtained by our approach could be inferior to several other methods. We speculate that in these networks, there exist dense connections among detected community centers, therefore it becomes more challenging to identify exact communities. On the contrary, for networks whose community centers demonstrate sparse interconnections among each other, our approach is expected to produce significant performance.

4. Discussion and conclusion

The local and global information are all considered in three-stage algorithm including central nodes identification, label propagation and communities combination. The distance of each pair of nodes is a global information in the stage of identifying the community center nodes. It completely depends on the structure of networks even though it is also a time consuming at the first stage. And in the third stage, there is no any doubt to take modularity to measure community. The three-stage algorithm is also compared with the other seven algorithms, in which Fastgreedy, Informap,Eigenvector, LPA, Walktrap and Louvain are popular methods to detect communities while node2vec is unusual. In this section, the further discussion on the selection of similarities and the node2vect are given in follows.

4.1. Similarity comparison

In the second stage of TS algorithm, the noncentral nodes will be labeled the same color if they are the most similar, where the similarity of nodes is a local information to diffuse the labels until all nodes are labeled. Similarity measures play an important role in label propagation stage. And a critical effect on the accuracy of the stage 2 is to choose proper index to express similarity. Hence, we employ different similarity measures and compare the different results on three-stage algorithm.

In Table 11, we list ten similarity measures used to measure the similarity of nodes or links in networks summarized in reference [32]. The measures of Common Neighbors(CN), Salton, Jaccard, Hub Promoted(HP), Hub Depressed(HD), Leicht–Holm–Newman(LHN), Preferential Attachment(PA) and Adamic–Adar (AA) are based on the local structural information (i.e. neighborhood information). In addition, the first seven measures, from CN to LHN, only differ in the denominator. If the investigated network simultaneously has large clustering coefficient and large degree heterogeneity, there are significant differences among those seven measures. PA is a proximity measure and often used to quantify the functional significance of edges subject to various network-based dynamics, which does not require information on the neighborhood of each node. AA refines the simple counting of common neighbors by assigning the less connected neighbors

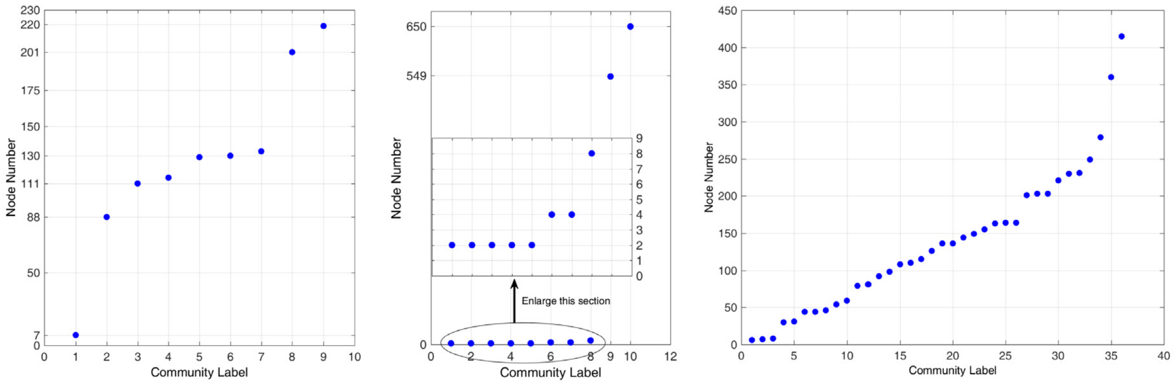
Table 9
Communities in Adjnoun network detected by three-stage algorithm.

C_i	Nodes
C_1	18,29,31,32,33,34,35,36,39,40,53,66,83,85,87,94,104,105,108
C_2	1,2,3,4,9,10,20,23,26,27,41,42,43,46,47,57,62,74,78,86,91,92,98,112
C_3	5,6,21,28,44,64,65,80,81,82,84,95,99,101,110,111
C_4	7,8,14,15,17,24,25,37,38,48,50,52,55,56,58,60,67,68,69,71,73,76,88,106,109
C_5	11,12,13,19,22,45,49,51,54,70,79,90,100,103
C_6	16,61,63,72,77,89,93,96,97,102,107
C_7	30,59,75

C_i is the community index.

Table 10
Communities in Jazz network detected by three-stage algorithm.

C_i	Nodes
C_1	2,7,10,11,12,14,17,19,22,30,31,34,36,43,49,52,53,54,55,56,57,59,61,67,69,70,71, 72,81,82,83,84,87,89,93,94,112,113,114,118,121,125,127,129,130, 136,141, 142,143,146,150,151,158,161,164,165,170,174,175,177,178,182, 183,185,186,190,192,193,194,195,196,197
C_2	1,8,9,15,16,20,23,24,32,33,35,38,40,42,44,46,48,50,58,60,62,63,64,65,66,68,74,78, 80,91,95,98,99,100,101,103,104,105,106,107,108,109,110,111,116 117,119, 120, 122,123,131,132,134,135,137,139,154,159,162,166,168,171 179,187,188
C_3	3,4,5,6,13,18,21,25,26,27,28,29,37,39,41,45,47,51,73,75,76,77,79,85,86,88,90,92, 96,97,102,115,124,126,128,133,138,140,144,145,147,148,149,152,153,155, 156,157,160,163,167,169,172,173,176,180,181,184,189,191,198



(a) Email

(b) Polblogs

(c) Power

Fig. 6. The three network's scatter of community label's node number.

Table 11
Definition of ten similarity measures.

Measures	Definition	Measures	Definition
CN	$\vert\mathit{\Gamma}(v_i)\cap\mathit{\Gamma}(v_j)\vert$	HD	$\frac{\vert\mathit{\Gamma}(v_i)\cap\mathit{\Gamma}(v_j)\vert}{max\{k_{(v_i)},k_{(v_j)}\}}$
Salton	$\frac{\vert\mathit{\Gamma}(v_i)\cap\mathit{\Gamma}(v_j)\vert}{\sqrt{k_{(v_i)}\times k_{(v_j)}}}$	LHN	$\frac{\vert\mathit{\Gamma}(v_i)\cap\mathit{\Gamma}(v_j)\vert}{k_{(v_i)}\times k_{(v_j)}}$
Jaccard	$\frac{\vert\mathit{\Gamma}(v_i)\cap\mathit{\Gamma}(v_j)\vert}{\vert\mathit{\Gamma}(v_i)\cup\mathit{\Gamma}(v_j)\vert}$	PA	$k_{(v_i)}\times k_{(v_j)}$
Sϕrenson	$\frac{2\vert\mathit{\Gamma}(v_i)\cap\mathit{\Gamma}(v_j)\vert}{k_{(v_i)}+k_{(v_j)}}$	AA	$\sum_{z\epsilon\mathit{\Gamma}(v_i)\cap\mathit{\Gamma}(v_j)}\frac{1}{\log k(z)}$
HP	$\frac{\vert\mathit{\Gamma}(v_i)\cap\mathit{\Gamma}(v_j)\vert}{min\{k_{(v_i)},k_{(v_j)}\}}$	RA	$\sum_{z\epsilon\mathit{\Gamma}(v_i)\cap\mathit{\Gamma}(v_j)}\frac{1}{k(z)}$

more weight [33]. Assuming that each transmitter has a unit of resource, and equally distribute it between all its neighbors, then resource allocation (RA) index can be defined as the amount of resource v_j received from v_i, which works well on the networks with large clustering coefficient, high degree heterogeneity and absence of a strongly assortative linking pattern.

In order to choose well-performing algorithms, the results of the tenth cited similarities that have been tested in the algorithms will be compared: Table 12 presents the Q and NMI results for both algorithms using different similarities, which are tested in four data sets, from which we can see that except PA, AA and RA, other seven similarity measures can obtain better results of community detection, but there is little difference among them on different networks, e.g., Jaccard, Sϕrenson HD and LHN obtain the best detection results on Karate network, LAN and HD arrive at the best detection results on Polbooks network. The similarities except PA, AA and RA can obtain excellent results on both dolphins and football networks.

PA, AA and RA perform the worst on four networks, because it is often used to quantify the functional significance of edges subject to various network-based dynamics. Maybe it is suitable to dynamical networks. The Sϕrenson similarity gives the best results compared to other similarities. Based on these results, the proposed algorithm uses Sϕrenson similarity to detect similar nodes (see Fig. 7).

4.2. Why node2vec is introduced?

Node2vec [5] is one of the outstanding semi-supervised machine-learning methods by which each node in networks is

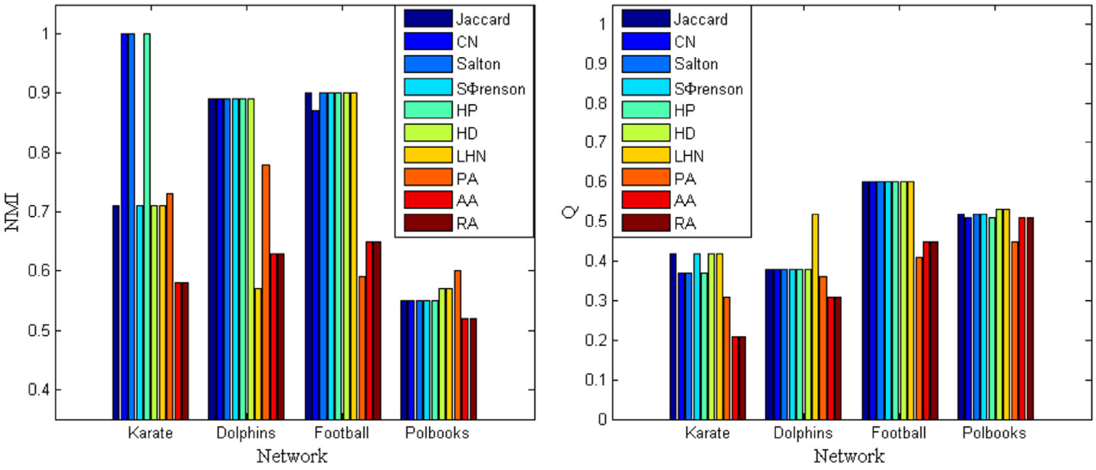


Fig. 7. NMI and modularity results for the ten similarity algorithms.

Table 12
Results of NMI and Q of the above similarities on three-stage algorithm.

Sim	Karate			Dolphins			Football			Polbooks		
	C	NMI	Q	C	NMI	Q	C	NMI	Q	C	NMI	Q
Jaccard	4	0.71	0.42	2	0.89	0.38	10	0.90	0.60	4	0.55	0.52
CN	2	1.00	0.37	2	0.89	0.38	9	0.87	0.60	3	0.55	0.51
Salton	2	1.00	0.37	2	0.89	0.38	10	0.90	0.60	4	0.55	0.52
Sϕrenson	4	0.71	0.42	2	0.89	0.38	10	0.90	0.60	4	0.55	0.52
HP	2	1.00	0.37	2	0.89	0.38	10	0.90	0.60	3	0.55	0.51
HD	4	0.71	0.42	2	0.89	0.38	10	0.90	0.60	4	0.57	0.53
LHN	4	0.71	0.42	4	0.57	0.52	10	0.90	0.60	4	0.57	0.53
PA	2	0.73	0.31	2	0.78	0.36	6	0.59	0.41	2	0.60	0.45
AA	2	0.58	0.21	2	0.63	0.31	7	0.65	0.45	4	0.52	0.51
RA	2	0.58	0.21	2	0.63	0.31	7	0.65	0.45	4	0.52	0.51

translated into a vector, and then provides the rich features' representation of nodes. Therefore, node2vec is a technical tool to deal with the expression of data. In this issue, node2vec is combined with k-means clustering to detect communities and simply denoted by node2vec. The results of the clustering are compared with the results of TS algorithm together with the other seven algorithms.

The parameters of node2vec in this issue are set as follows: the dimension is 128, the walks per node is set 10, the walk length is set 80, and the neighbors size is set 10. because both of the local and the global information are considered in community detection. Then nodes in the neighborhood and nodes far away from central nodes are both important. Therefore, during the feature learning in node2vec, the speeds of the random walk visiting or leaving a node are set to 1.

Comparing TS algorithm with node2vec and the other six community detection algorithms, experiments' results show that Node2vec performs well on the most networks. That might because node2vec is the node's vector representation while the other seven methods and our TS are the nodes' matrix's. The dimension of vector representation is lower than matrix's. Then, the efficiency of node2vec is better than TS algorithm and the other six methods. Hence, node2vec lights up the way to detect communities by machine learning methods.

4.3. Conclusion

The three-stage algorithm is easy to understand, because the small-world phenomena displays the centralities, and the social convergence makes the community. Modularity is just a artificial criteria having nothing with the generating mechanism of communities. Although TS algorithm dose not always show the best performance in all experimental networks comparing with the seven other popular algorithms both on synthetic networks and real-world networks. The results reveal that TS algorithm is close to Louvain and Walktrap and they achieve the same performance whatever the networks topology are. In other words, the three-stage algorithm is also robust, and the performance of efficient and accuracy are also strong. That is, TS algorithm provides an alternative method to detect communities step by step in social networks.

The results in this issue show that TS algorithm can often achieve better performance except Louvain. It must be sincerely admitted that the excellent performance of Louvain algorithm in community detection. Louvain is a classical community detection algorithm with high efficiency, which has already been incorporated in some network analysis tools such as Gephi, graph and networkX, which shows that Louvain is approved by researches. We think achieving similar results with Louvain also show the effectiveness and efficiency of the proposed TS algorithm.

There are some works to improve for three-stage algorithm in the future: The complexity is too high for big networks. Therefore, optimizing the first stage is one of the further works.

Acknowledgments

We would like to thank the anonymous reviewers for the constructive comments and suggestions, which undoubtedly improved the presentation of this paper. We also show our great appreciation to all the authors who collected and shared the data, such as Karate, Dolphins, Football, Polbooks, Email, Jazz, Adjnom, Lesmis and LFRs to be benchmark networks.

References

[1] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (2002) 7821–7826.

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

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

[4] A. Clauset, M.E. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (2004) 066111.

[5] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 13–17.

[6] C. Shang, S. Feng, Z. Zhao, J. Fan, Efficiently detecting overlapping communities using seeding and semi-supervised learning, Int. J. Mach. Learn. Cybern. 6 (2015) 1–14.

[7] Z. Ding, X. Zhang, D. Sun, B. Luo, Low-rank subspace learning based network community detection, Knowl.-Based Syst. 155 (2018) 71–82.

[8] J.C. Lv, K.K. Tan, Z. Yi, S. Huang, A family of fuzzy learning algorithms for robust principal component analysis neural networks, fuzzy systems, IEEE Trans. 18 (2010) 217–226.

[9] M. M(pi)up, M. Schmidt, Bayesian community detection, Neural Comput. 24 (2012) 2434–2456.

[10] W. Liu, M. Pellegrini, X. Wang, Detecting communities based on network topology, Sci. Rep. 4 (2014) 5739.

[11] K.R. Žalik, B. Žalik, Memetic algorithm using node entropy and partition entropy for community detection in networks, Inform. Sci. 445–446 (2018) 38–49.

[12] G. Bello-Orgaz, S. Salcedo-Sanz, D. Camacho, A multi-objective genetic algorithm for overlapping community detection based on edge encoding, Inform. Sci. 462 (2018) 290–314.

[13] A. Lancichinetti, S. Fortunato, Consensus clustering in complex networks, Sci. Rep. 2 (2012) 336.

[14] S. Zhang, R.-S. Wang, X.-S. Zhang, Identification of overlapping community structure in complex networks using fuzzy c-means clustering, Physica A 374 (2007) 483–490.

[15] Y. Pan, D.-H. Li, J.-G. Liu, J.-Z. Liang, Detecting community structure in complex networks via node similarity, Physica A 389(14) (2012) 2849–2857.

[16] K.R. Žalik, Maximal neighbor similarity reveals real communities in networks, Sci. Rep. 5 (2015) 18374.

[17] S. Ahajjam, M.E. Haddad, H. Badir, A new scalable leader-community detection approach for community detection in social networks, Social Networks 54 (2018) 41–49.

[18] M. Rosvall, C.T. Bergstrom, Maps of information flow reveal community structure in complex networks, Proc. Natl. Acad. Sci. USA 105 (2008) 1118–1123.

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

[20] 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.

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

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

[23] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech. Theory Exp. (2005) P09008.

[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, The emergent properties of a dolphin social network, Proc. R. Soc. B 270 (2003) S186–S188.

[26] L.A. Adamic, N. Glance, The political blogosphere and the 2004 us election: divided they blog, in: Proceedings of the 3rd International Workshop on Link Discovery, 2005, pp. 36–43.

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

[28] D.E. Knuth, The stanford graphbase: a platform for combinatorial algorithms, in: Acm-Siam Symposium on Discrete Algorithms Society for Industrial and Applied Mathematics, 1993, pp. 41–43.

[29] P.M. Gleiser, L. Danon, Community structure in jazz, Adv. Complex Syst. 6 (2003) 565–573.

[30] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, A. Arenas, Self-similar community structure in a network of human interactions, Phys. Rev. E 68 (2003) 065103.

[31] D.J. Watts, S.H. Strogatz, Collective dynamics of ’small-world’ networks, Nature 393 (1998) 440–442.

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

[33] L.A. Adamic, E. Adar, Friends and neighbors on the web, Soc. Netw. 25 (2003) 211–230.