Knowledge-Based Systems 195 (2020) 105626

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Community detection in complex networks with an ambiguous structure using central node based link prediction ☆

Hao Jiang, Zhenjie Liu, Chunlong Liu, Yansen Su, Xingyi Zhang *

Key Lab of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230601, China

ARTICLE INFO

Article history:
Received 12 July 2019
Received in revised form 3 February 2020
Accepted 6 February 2020
Available online 11 February 2020

Keywords:
Complex network
Community detection
Link prediction
Central node

ABSTRACT

Community detection in complex networks has aroused wide attention, since it can find some useful information hidden in the networks. Many different community detection algorithms have been proposed to detect the communities in a variety of networks. However, as the ratio of each node connecting with the nodes in other communities increases, namely, the community structure of networks becomes unclear, the performance of most existing community detection algorithms will considerately deteriorate. As a method of finding missing information, link prediction can predict undiscovered edges in the networks. However, the existing link prediction based community detection algorithms cannot deal with the networks with an ambiguous community structure, namely, the networks having a mixing parameter greater than 0.5. In this paper, we design a new strategy of link prediction and propose a community detection algorithm based on this strategy to detect the communities in complex networks, especially for the networks with an ambiguous community structure. Experimental results on synthetic benchmark networks and real-world networks indicate that the proposed community detection algorithm outperforms five state-of-the-art community detection algorithms, especially for those without a clear community structure.

1. Introduction

Community detection is to divide the nodes in complex networks into several groups, so that the nodes in the same group are tightly connected, whereas the nodes in different groups are sparsely connected [1]. The detected community structure can help researchers to find some useful information hidden in complex networks, e.g., functional modules in biological networks [2,3] and popular research topics in journals [4,5]. Moreover, the researchers can also build the recommendation systems according to the community structure [6,7]. Therefore, in the past few years, many community detection algorithms have been proposed, which achieved a promising performance on complex networks with a clear community structure. However, as the structure of communities becomes unclear, the performance of these existing algorithms will considerably deteriorate. In this paper, we refer the networks having the mixing parameter greater than 0.5 as the networks with an ambiguous community structure, whereas the networks having the mixing parameter smaller than 0.5 as the networks with a clear community structure. Here, the mixing parameter indicates the ratio of each node connecting with the nodes in other communities of the networks.

Link prediction, as a branch of data mining, is usually used to find missing information. It can also be employed to find the edges in the networks that are undiscovered or might emerge in the future. Although the link prediction has been applied to community detection, the existing methods only utilize the designed prediction index to score the probability of an edge emerging between each node pair, and then add the edges with a high score into the network. Usually the nodes in the node pair with a high score are more likely belonging to a community. However, when the structure of communities becomes unclear, the node pairs belonging to different communities will also get a high score. In this case, link prediction will add a large number of edges between communities into the network, which will weaken the community structure. That is to say, the existing link prediction based community detection algorithms cannot deal with the networks with an unclear community structure.

Generally speaking, a central node is located at the center of the community, which will connect with the majority of the nodes in the community. Therefore, link prediction related with the central nodes can increase the edges within the community,

☆ 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.2020.105626.

* Corresponding author.
E-mail addresses: jh9305@mail.ustc.edu.cn (H. Jiang), zhenjiel56@gmail.com (Z. Liu), 18255811669@163.com (C. Liu), suyansen1985@163.com (Y. Su), xyzhanghust@gmail.com (X. Zhang).

https://doi.org/10.1016/j.knosys.2020.105626

whereas reduce the edges between the communities. Then, the structure of communities will be enhanced. To this end, in this paper, we propose a central node based link prediction strategy to deal with the networks without a clear community structure, and apply it to community detection in complex networks. The main contribution of this paper can be summarized as follows.

(1) A new link prediction strategy is proposed to deal with the network with an ambiguous community structure. Different from traditional link prediction strategies, the proposed method only predicts the edges related with the central nodes, and it uses both the common neighbor information and the non-common neighbor information of the node pairs to calculate their similarity, instead of just common neighbor information. In this way, the structure of communities will be enhanced by the proposed link prediction strategy.
(2) Utilizing the proposed link prediction strategy, we propose a community detection algorithm for complex networks, namely CLPE, especially for those without a clear community structure. In CLPE, the proposed link prediction strategy is first used to enhance the community structure of complex networks. Then, a community expansion strategy is adopted to find the communities in complex networks. Finally, a community merging strategy is used to optimize the detected communities.
(3) Simulation experiments are conducted on both synthetic benchmark networks and real-world networks to verify the performance of the CLPE for community detection. Experimental results indicate that the proposed CLPE is superior over five state-of-the-art community detection algorithms, especially for the networks with an unclear community structure and those with large communities.

The rest of this paper is organized as follows. Existing community detection algorithms are reviewed in Section 2. Section 3 presents the details of the proposed CLPE. The experimental results and the detailed analysis are reported in Section 4. Section 5 discusses the CLPE and Section 6 gives the conclusion and future work.

2. Related work

Recently, a large number of algorithms have been proposed to detect communities in complex networks. The first group is the graph partitioning based algorithms, which iteratively divide the network into several subnetworks until a given number of communities are obtained. In the work presented by Kernighan et al. [8], a method called Kernighan–Lin algorithm was proposed by randomly partitioning the network into two subnetworks and exchanging the nodes among subnetworks to detect the communities. Besides, a community detection algorithm called BSSCD, was proposed by using the similarity of nodes to detect the communities [9].

The second group is the spectral clustering based algorithms, where the eigenvectors of the network are used to detect the communities. This kind of algorithms was firstly proposed in the work presented by Donath et al. [10]. Then, in the work presented by Shi et al. [11] and Ng et al. [12], two improved algorithms were proposed by utilizing the unnormalized spectral clustering and normalized spectral clustering techniques, respectively. Recently, a new spectral clustering algorithm using sparse linear coding was proposed in the work presented by Mahmood et al. [13], and a spectral clustering based community detection algorithm combining the multi-similarity method and K-means clustering algorithm was proposed in the work presented by Ni et al. [14].

The third group is the modularity optimization based algorithms. The modularity is the most widely used metric for measuring the quality of communities [15]. The larger the modularity is, the higher quality of the communities often is. Therefore, by optimizing the modularity, the communities in the networks can be detected. However, optimizing the modularity is an NP-complete problem [16], and many optimization algorithms have been employed to optimize the modularity, such as greedy technique [17,18], simulated annealing [19] and genetic algorithm [20,21].

The fourth group is the label propagation based algorithms. This group of algorithms first assigns a unique label to each node, and then iteratively updates the label according to the neighboring nodes. The nodes with the same label belong to a community. The first label propagation based community detection algorithm called LPA, was proposed in the work presented by Raghavan et al. [22], where the label of each node is updated according to the label shared by the majority of its neighbors. Hereafter, some improvements have been reported for LPA, such as LPAm [23], CK-LPA [24], LPA-CBD [25] and PSPLPA [26].

The fifth group is the hierarchical clustering based algorithms, where the edges connecting nodes with a low similarity are iteratively removed to detect the communities. As a representative of such algorithms, GN algorithm iteratively removes the edges with the largest betweenness to obtain the communities [1,2]. Since the calculation of betweenness is time-consuming, some improved algorithms have been proposed to efficiently detect the communities [27,28].

In addition to the above five groups of algorithms, there are also some other algorithms for community detection in complex networks, such as random walk [29,30], local expansion [31], multi-objective evolutionary algorithm [32–36], and nonnegative matrix factorization [37]. However, the performance of all these community detection algorithms considerably deteriorates as the community structure becomes ambiguous.

Moreover, as aforementioned, the link prediction has also been used for community detection. Link prediction is a branch of data mining, by which the missing information can be found [38]. Applying it to complex networks, link prediction can find the edges which are undiscovered or might emerge in the future. Therefore, some researchers have used link prediction for community detection. In the work presented by Burgess et al. [39], a community detection algorithm, termed EdgeBoost, was proposed, where the link prediction was first used to add some edges into networks, and then the traditional network clustering algorithm was employed to detect the communities. In addition, two novel prediction indices was proposed in the work presented by Cheng et al. [40] to improve the effectiveness of link prediction and community detection. Besides, in the work presented by Chen et al. [41], a link prediction based community detection algorithm was proposed, where a novel link prediction technique was used to replace some links in the networks to enhance the structure of communities.

Although these algorithms use the link prediction to detect the communities in complex networks, as mentioned in Section 1, when the structure of communities is ambiguous, using link prediction for all nodes will make the structure of communities more ambiguous. Besides, the existing link prediction indices only use the common neighbor information of nodes, which will also make the structure of communities more ambiguous when the community structure is unclear.

3. The proposed CLPE algorithm

In this section, we present the proposed community detection algorithm CLPE in detail. In CLPE, a central node based link prediction strategy is first suggested to enhance the structure of communities and then a local expansion strategy is employed to detect the communities from the enhanced networks. To be specific, the proposed algorithm consists of two main steps: (1) Central node based link prediction, (2) Community expansion.

3.1. Central node based link prediction

As mentioned above, the traditional link prediction strategies predict a large number of edges between different communities when the structure of communities is unclear, which will make the community structure more ambiguous. To this end, in this paper, we propose a link prediction strategy which only predicts the edges related to the central nodes of the communities. The central node of a community generally locates at the center of the community. That is to say, a central node will connect with the majority of the nodes in the same community. Therefore, carrying out the link prediction related with the central nodes can effectively reduce the probability of predicting the edges between communities.

In this paper, a local density based central node identification strategy is adopted to find the central nodes in complex networks. This method was proposed by Rodriguez and Laio [42] in 2014, and was used by Wang et al. [43] for community detection. It considers two indices for each node i, the node density ρ_i and the relative distance δ_i. The node density ρ_i of node i is defined as follows

$$\rho_i = \sum_j \Psi(d_{i,j} - d_c). \tag{1}$$

In (1), $\Psi(x) = 1$ if $x \le 0$, otherwise $\Psi(x) = 0$, $d_{i,j}$ denotes the distance between nodes i and j, and d_c is the cutoff distance. Here, we set d_c to 1, and in this case the node density is the degree of the node. The sensitivity analysis of parameter d_c will be given in Section 4.5.

The relative distance δ_i is measured by the minimum distance between node i and the other nodes with the higher density than i, which can be calculated by (2).

$$\delta_i = \min_{\rho_j > \rho_i}(d_{i,j}). \tag{2}$$

Usually, the center of a community is a node with the higher density than its neighbors, meanwhile it has a relatively large distance to the nodes with a higher density. Therefore, the central index of node i is defined as follows

$$SC_i = \rho_i * \delta_i. \tag{3}$$

For the node with the largest node density, we conventionally take $\delta_i = \max_j(d_{i,j})$. Note that only for nodes having a local or global maximal density, their δ_i are much larger than the distance of the nearest neighbors. The SC value of the node with the largest node density is equal to $\max_i(\rho_i) * \max_j(d_{i,j})$.

From the definition, the central index of node i is proportional to ρ_i and δ_i. The larger the SC_i value of node i is, the more likely node i is to be a central node of a community. Algorithm 1 presents the procedure of identifying central nodes, which is performed as follows. First, the SC_i of each node i in the network is calculated. Second, the nodes are sorted in the descending order according to the SC values, and the first half of nodes are added into the candidate set CC. Third, the first node CN_k in CC is taken as a central node. If the distance between CN_k and the rest nodes in CC is smaller than the cutoff distance d_c, then these candidate nodes will be deleted from CC. This procedure is repeated until the candidate set CC is empty.

Algorithm 1 FindCentralNode

Input: $G(V, E)$: a complex network;
SC: the central index value of all nodes in the network.
Output: CN: central node set.

$CN \leftarrow \emptyset$
sort the nodes in V in descending order according to SC
$CC \leftarrow$ the first half of nodes in V
$k \leftarrow 0$
while $|CC| \neq 0$ **do**
  $k \leftarrow k + 1$
  $CN_k \leftarrow$ the first node in CC
  $CC \leftarrow CC - \{CN_k\}$
  for each $v \in CC$ **do**
    if $dist(v, CN_k) < d_c$ **then**
      delete v from CC
    end if
  end for
  $CN \leftarrow CN \cup \{CN_k\}$
end while
return CN

Once the central nodes in the network are identified, the CLPE algorithm uses the central node based link prediction strategy to enhance the structure of communities. As mentioned above, the link prediction related to the central nodes can reduce the prediction probability of the edges in the community, and strengthen the prediction probability of the edges between the communities. For this reason, the CLPE algorithm only predicts whether to add or remove edges related with the central nodes. The prediction index related to the central nodes is defined as follows

$$\begin{aligned} F(c,x) = & \sum_{z \in \Gamma(c) \cap \Gamma(x)} \left(\frac{S(c,z)}{deg(c)} + \frac{S(x,z)}{deg(x)}\right) \\ & - \sum_{k \in \Gamma(x) | (\Gamma(c) \cap \Gamma(x))} \frac{S(x,k)}{deg(x)}, \end{aligned} \tag{4}$$

$$S(c,x) = \frac{|\Gamma(c) \cap \Gamma(x)|}{|\Gamma(c) \cup \Gamma(x)|}. \tag{5}$$

In (4), c denotes a central node of the network, $\Gamma(c)$ is the set of all neighbors of c, $deg(c)$ is the degree of c, $S(c, x)$ is the *Jaccard* similarity between node c and node x, which can be calculated by (5), and $\Gamma(x)|(\Gamma(c) \cap \Gamma(x))$ is the set of nodes belonging to $\Gamma(x)$ while not belonging to $\Gamma(c) \cap \Gamma(x)$. The first item in (4) uses the common neighbor information to describe the attraction of the central node c to the node x, whereas the second item uses the non-common neighbor information to describe the repulsive force of node x to the node c. If $F(c, x) > 0$, it indicates that the node x has obvious dependent relation to the central node c, which means that the node x is more likely to belong to the same community with node c; otherwise, it indicates that nodes c and x are more likely belonging to different communities.

It can be seen from the above definition, the link prediction index proposed in this paper can better reflect the characteristics of communities. In general, the common neighbors of nodes in the same community are obviously more than those of nodes in different communities. When the structure of communities is unclear, the connection of nodes will become chaotic. This leads to a large number of common neighbors of nodes in different communities. Therefore, when the structure of communities is ambiguous, the link prediction only relying on common neighbor information of node pairs will become unreliable. To this end, besides the common neighbor information, the proposed link prediction strategy also uses the non-common neighbor information, namely, the second item in (4), to reinforce the difference

Algorithm 2 LinkPrediction

Input: $G(V, E)$: a complex network;
CN: the central node set of the network.
Output: G': changed network.

$G' \leftarrow G$
for each $c \in CN$ **do**
  $NI \leftarrow$neighbor set of node c
  for each $v \in NI$ **do**
    if $F(c, v) < 0$ **then**
      remove the edge between node c and node v from G'
    end if
  end for
  $SI \leftarrow$a set of nodes that have common neighbors but not connect with node c
  for each $v \in SI$ **do**
    if $F(c, v) > 0$ **then**
      add the edge between node c and node v into G'
    end if
  end for
end for
return G'

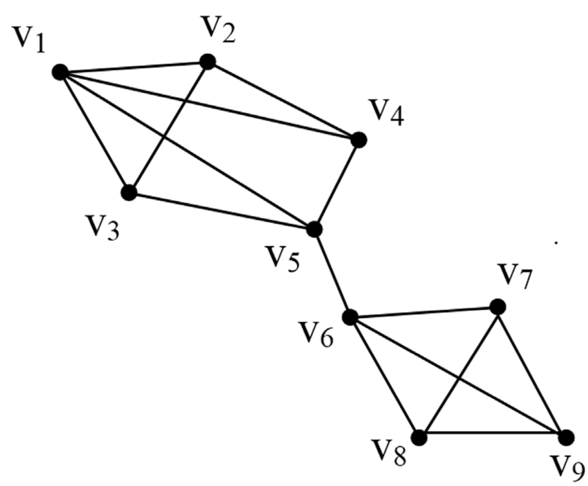


Fig. 1. An example of community expansion.

between nodes. Specifically, the steps of central node based link prediction are performed as follows.

Assume that the NI is the neighbor set of a central node c, and SI is a set of nodes that have common neighbors but not connect with the node c. For each central node c, the proposed link prediction strategy performs the following two operations. (1) Remove Edges: For each node v in NI, if $F(c, v) < 0$, the algorithm will remove the edge between nodes c and v; (2) Add Edges: For each node v in SI, if $F(c, v) > 0$, the algorithm will add an edge between nodes c and v. Algorithm 2 gives the procedure of central node based link prediction.

3.2. Community expansion strategy

According to the community structure, the central node of a community has the highest centrality index. As the distance from the central node increases, the centrality of nodes in the community will gradually decrease. After adding and removing edges related to the central nodes, the nodes at the community boundary will directly connect with the central node, and the connection between the central nodes and the nodes in the other communities will be removed. Therefore, the central node based

Algorithm 3 LocalExpand

Input: $G(V, E)$: a complex network;
SC: the center index value of all the nodes.
Output: Com: the communities of G.

$Com \leftarrow \emptyset$
set all nodes in the network to be unvisited and sort them in descending order according to the SC value
for each $v \in V$ **do**
  if v is not visited **then**
    v is marked as visited
    $expand \leftarrow v$; $C \leftarrow v$
    $NI \leftarrow$ nodes that are not visited in the neighbors of node v
    while $|expand| \neq 0$ **do**
      $expand \leftarrow \emptyset$
      $NI \leftarrow$ sort nodes in NI in descending order according to SC value
      for each $ni \in NI$ **do**
        $NN \leftarrow$ the neighboring nodes of ni whose SC values are larger than the SC value of ni
        if $|NN| > 0$ **then**
          $oval \leftarrow$ the average similarity between ni and its neighbors
          if $\arg\max_{nn \in NN} S(nn, ni) \in C$ && $S(nn, ni) > oval$ **then**
            $expand \leftarrow expand \cup \{ni\}$
            $C \leftarrow C \cup \{ni\}$
            ni is marked as visited
          end if
        end if
      end for
      $NI \leftarrow$ unvisited neighbor of all nodes in C
    end while
    $Com \leftarrow Com \cup C$
  end if
end for
return Com

link prediction strategy clears the community boundary to a certain extent, by which the convergence of the community expansion is accelerated, and the accuracy of community detection is improved.

When expanding the communities, the CLPE algorithm traverses the nodes in the network according to the SC value in the descending order. For a node v in the network which has not been assigned to a community, the CLPE first finds all the neighboring nodes NI of v, and then checks whether the nodes in NI need to be added in the community of node v. Next, the CLPE algorithm traverses nodes in NI according to the descending order of SC values. For a node v' in NI, v' will be added in the community of node v if v' meets the following conditions: (1) The neighboring nodes of v' have nodes whose SC values are greater than the SC value of v'; (2) Among these nodes, the node with the largest similarity (*Jaccard* index) with the node v' has been added in the community of v; (3) The largest similarity is not smaller than the average similarity between node v' and its neighbors. By taking the same operations on the neighbors of each newly added node, the community associated with node v is further expanded until no node in the network can be added. Once the community associated with v is found, the CLPE starts to detect the communities associated with other nodes, which have not been assigned to a community. Algorithm 3 gives the details of the community expansion operation.

For better understanding the operation of community expansion, we take the network in Fig. 1 as an example to illustrate

Algorithm 4 Framework of the proposed CLPE

Input: $G(V, E)$: a complex network;
Output: *Community*: the communities of G.

1: $Community \leftarrow \emptyset$
2: Calculate the central index value SC of all nodes
3: $CenterNode \leftarrow FindCentralNode(G, SC)$
4: $G \leftarrow LinkPrediction(G, CenterNode)$
5: $Community \leftarrow LocalExpand(G, SC)$
6: Merge the communities according to modularity Q
7: **return** G'

the community expansion in detail. In Fig. 1, the SC values of nodes V_1 to V_9 are {12, 3, 3, 3, 8, 12, 3, 3, 3}. First, the node V_1 will be added in the community C_1 and remarked as visited, because it has the largest SC value. The neighboring nodes of V_1 are $\{V_5, V_2, V_3, V_4\}$ (sorted by the SC value). Second, the proposed method will check whether the neighboring nodes can be added in community C_1. For the node V_5, its neighboring nodes are $\{V_1, V_3, V_4, V_6\}$. The average similarity between V_5 and its neighboring nodes is $oval = (\frac{2}{6} + \frac{1}{6} + \frac{1}{6} + 0)/4 = \frac{1}{6}$. Because (1) there is a neighboring node that has a larger SC value than that of node V_5, namely, the node V_1; (2) the node V_1 has been added in community C_1 and (3) the similarity between V_5 and V_1 ($\frac{2}{6}$) is greater than $oval$, the node V_5 will be added in the community C_1 and remarked as visited. Similarly, the nodes $\{V_2, V_3, V_4\}$ are also added in the community C_1. Third, the neighbors of the newly added nodes that are not visited will be checked whether they can be added in the community C_1. In the illustrative example, only the node V_6 is not visited. For the node V_6, its neighboring nodes are $\{V_5, V_7, V_8, V_9\}$. The average similarity between the node V_6 and its neighboring nodes is $oval = (0 + \frac{2}{5} + \frac{2}{5} + \frac{2}{5})/4 = \frac{3}{10}$. Only the node V_5 has a larger SC value than V_6. Because the similarity between nodes V_6 and V_5 (0) is smaller than $oval$, the node V_6 cannot be added in the community C_1. At this time, no node can be added in community C_1. Therefore, the community associated with V_8 has been found. Next, using the unvisited nodes and repeating above steps, the other communities can also be found.

3.3. The proposed CLPE

The general framework of CLPE is presented in Algorithm 4, which mainly consists of the following three steps. At the first step, Algorithm 1 is used to identify the central nodes in complex networks. At the second step, Algorithm 2 is employed to predict edges related with the central nodes. At the third step, Algorithm 3 is applied to community expansion. Here, it is worth mentioning that in Algorithm 3, the central index SC_i of each node needs to be recalculated after adding and removing edges.

In order to further optimize the results of community expansion, the CLPE algorithm merges the communities in a way that can increase the modularity Q [15,44]. Here, the modularity Q is calculated in the original network. In the CLPE, only when the community modularity Q is greater than the threshold α, the merging operation will be performed. We believe that when the modularity of a community is smaller than the threshold α, the community structure detected by the algorithm is not ideal, and the merging operation will make the detected results worse. When the modularity is greater than the threshold α, the merging operation will bring certain optimization of the detected results. The sensitivity analysis of parameter α will be given in Section 4.4.

Table 1
The meaning of the parameters in LFR networks.

Parameters	Meaning
N	The number of nodes in the network
d_{ave}	The average degree of the nodes
d	The maximum degree of the nodes
γ	The exponent of the degree distribution
β	The exponent of the community size distribution
min_c	The number of nodes in the smallest community
max_c	The number of nodes in the biggest community
μ	The ratio of each node connecting with the nodes in other communities.

3.4. Complexity analysis

In Algorithm 4, we can find that the proposed CLPE contains the following steps, (1) Central index value calculation, (2) Central node identification, (3) Central node based link prediction, (4) Community expansion, and (5) Community merging. Assume that there are E edges and N nodes in a network, and the average degree of each node is K. The time complexity of the first step is $O(N * E)$. Because for calculating the central index value of each node, the density of each node is first calculated which takes a time complexity of $O(N+E)$. Then the minimum distance between each pair of nodes is calculated needing a time complexity of $O(N*E)$, and finally the central index value of each node is calculated which consumes a time complexity of $O(N)$. The second step has a time complexity of $O(N*K)$, since the neighboring nodes of each central node need to be visited. The time complexity of link prediction is $O(N * K^2) = O(E * K)$ because the prediction index between each central node and its neighboring nodes and nodes that have common neighbors but not connect with it should be calculated. The step of community expansion holds a time complexity of $O(N * K)$. Because all the nodes are visited only once, and when a node is visited, the similarity between the node and its neighbors needs to be calculated. The time complexity of community merging is $O(N^2)$ because the maximum number of communities is N. Overall, the time complexity of the proposed CLPE is $O(N * E)$.

4. Experimental results and analysis

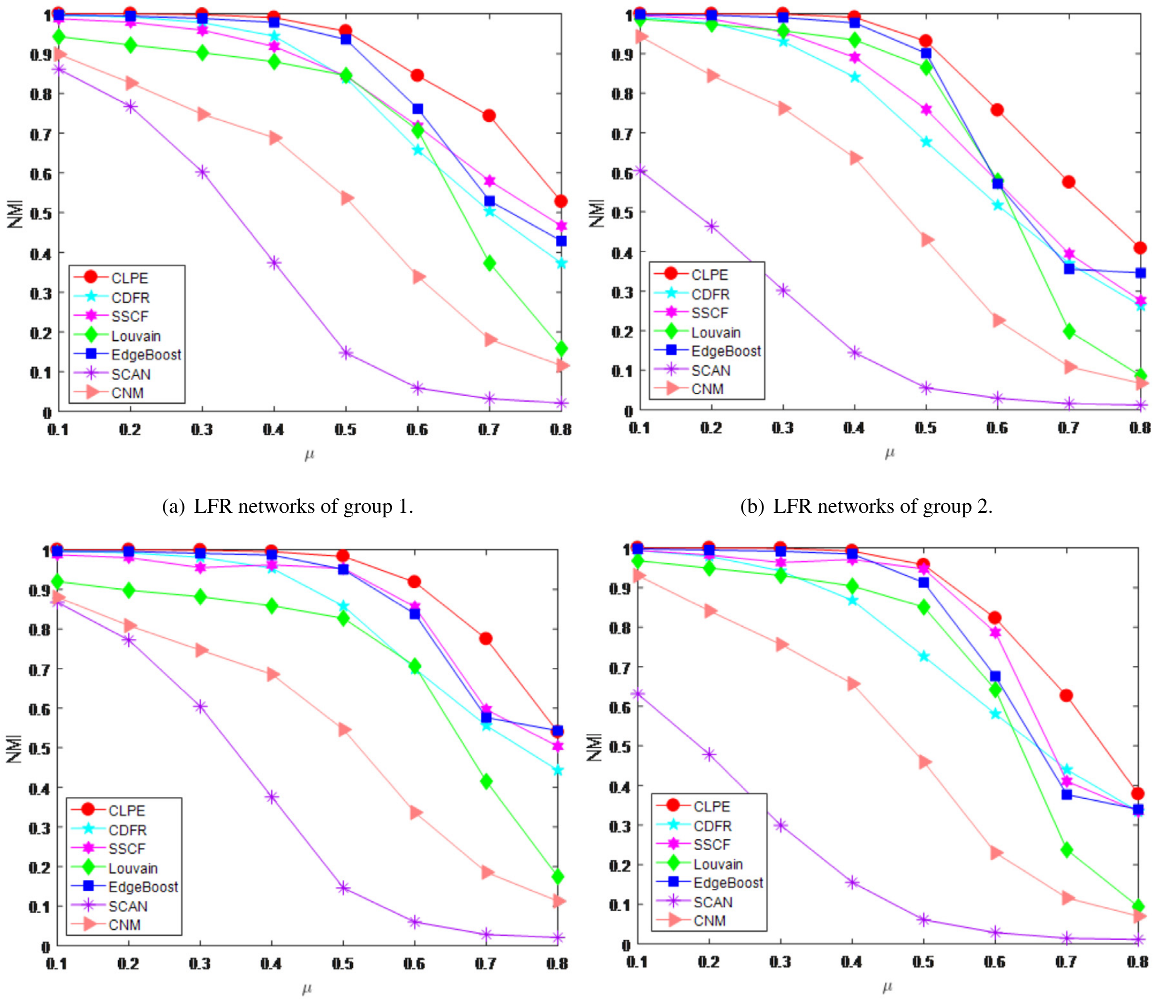
In this section, we verify the performance of the proposed CLPE by comparing it with six state-of-the-art community detection algorithms, including SSCF [13], Louvain [45], EdgeBoost [39], SCAN [46], CNM [47], and CDFR [48]. The experiments are conducted on both synthetic benchmark networks and real-world networks, and all experimental results reported for the CLPE are obtained in the case that the parameter α is set to 0.3.

4.1. Experimental setting and evaluation measures

4.1.1. Tested networks

The synthetic networks we adopted in the simulation experiments are the LFR networks [49]. Since LFR networks can reflect some important features of the real-world complex networks, they have been widely used for testing the performance of community detection algorithms. The LFR networks are defined as $LFR(N, d_{ave}, d, \gamma, \beta, min_c, max_c, \mu)$, and the meaning of these parameters is shown in Table 1. Note that the larger the value of μ is, the more ambiguous the community structure of LFR network is.

In simulation experiments, we consider four groups of LFR networks with different network sizes and community sizes. The details are listed in Table 2. For each group, eight LFR networks with the value of μ ranging from 0.1 to 0.8 with the interval 0.1



(a) LFR networks of group 1.

(b) LFR networks of group 2.

(c) LFR networks of group 3.

(d) LFR networks of group 4.

Fig. 2. The *NMI* values of the proposed CLPE and six compared algorithms on four groups of LFR benchmark networks with different values of μ.

Table 2
The network and community size of LFR networks.

	Network size	Community size
1	5000	$min_c = 10$, $max_c = 50$
2	5000	$min_c = 20$, $max_c = 100$
3	10 000	$min_c = 10$, $max_c = 50$
4	10 000	$min_c = 20$, $max_c = 100$

Table 3
The information of nine real-world networks.

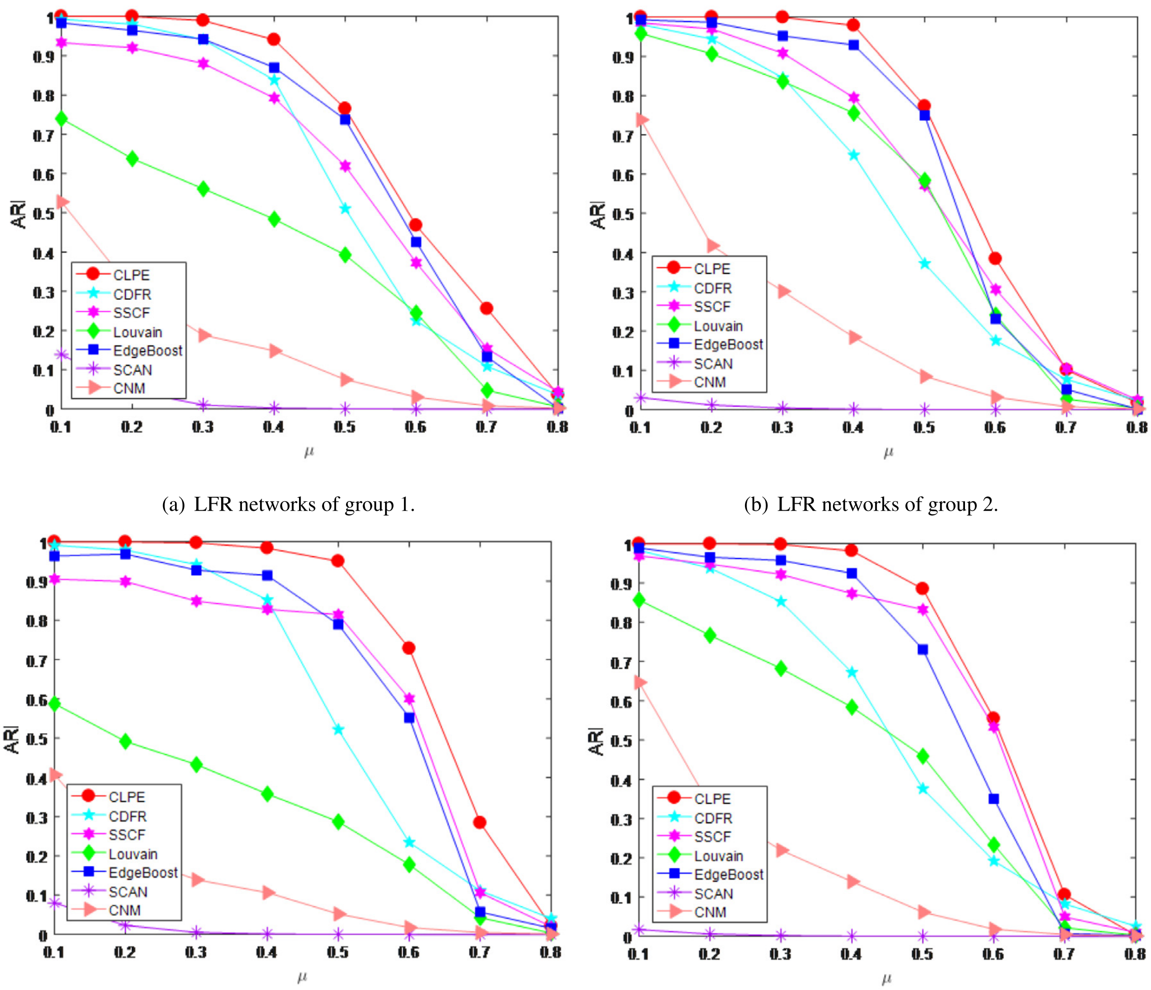
Network	N_{node}	N_{link}	N_{com}	Description
Karate	34	78	2	Zachary's karate club
Dolphin	62	160	2	Dolphin social network
Football	115	613	12	American college football
Polbooks	105	441	12	Books about US politics
Netscience	1589	2742	–	Netscience network
PPI	2445	6265	–	Protein-protein network
Blogs	3984	6803	–	Blogs network
ca-HepTh1	9877	25 998	–	Collaboration network of Arxiv High Energy Physics Theory
Erdos	6927	11 850	–	Erdos co-author network

are generated to verify the performance of CLPE for community detection under different ambiguous levels of the community structure.

For real-world networks, nine networks are tested in the experiments, namely, Zachary's karate club network (karate, for short) [50], Bottlenose dolphins network (dolphin, for short) [51], American college football network (football, for short) [1], Books about US politics network (polbooks, for short) [52], netscience network (netscience for short) [34], protein–protein network (PPI for short) [53], blogs network (blogs for short) [35], collaboration network of Arxiv high energy physics theory (ca-HepTh1 for short) [54] and Erdos co-author network (Erdos for short) [55]. The details of the nine real-world networks are listed in Table 3, where '-' means that the ground truth of the network is unknown.

4.1.2. Evaluation metrics

In this paper, we use three evaluation metrics to evaluate the quality of detected communities including the normalized mutual information (*NMI*) [56,57], adjusted rand index (*ARI*) [48] and average conductance of detected communities (*COND*) [58–60]. For the LFR networks, we adopt the indexes of *NMI* and *ARI* to measure the similarity between the true clustering results and



(a) LFR networks of group 1. (b) LFR networks of group 2.

(c) LFR networks of group 3. (d) LFR networks of group 4.

Fig. 3. The *ARI* values of the proposed CLPE and six compared algorithms on four groups of LFR benchmark networks with different values of μ.

detected results because the ground truth of LFR networks is known. For the real-word networks, the ground truth of some of them is unknown. So the *COND* is used as the evaluation metric for real-world networks.

Given two partitions A and B of a network, let C be the confusion matrix whose element $C_{i,j}$ is the number of nodes in the community i of the partition A that are also in the community j of the partition B.

The $NMI(A,B)$ is defined as follows

$$NMI(A,B)=\frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B}C_{i,j}\log(\frac{C_{i,j}N}{C_{i.}C_{.j}})}{\sum_{i=1}^{C_A}C_{i.}\log(\frac{C_{i.}}{N})+\sum_{j=1}^{C_B}C_{.j}\log(\frac{C_{.j}}{N})}. \tag{6}$$

In (6), $C_A(C_B)$ is the number of communities 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 in the network. If $A=B$, $NMI(A,B)=1$. If A and B are completely different, $NMI(A,B)=0$.

The $ARI(A,B)$ is defined as follows

$$ARI(A,B)=\frac{\sum_{i=1}^{C_A}\sum_{j=1}^{C_B}\binom{C_{i,j}}{2}-\binom{N}{2}^{-1}\sum_{i=1}^{C_A}\binom{C_{i.}}{2}\sum_{j=1}^{C_B}\binom{C_{.j}}{2}}{\frac{1}{2}\left[\sum_{i=1}^{C_A}\binom{C_{i.}}{2}+\sum_{j=1}^{C_B}\binom{C_{.j}}{2}\right]-\binom{N}{2}^{-1}\sum_{i=1}^{C_A}\binom{C_{i.}}{2}\sum_{j=1}^{C_B}\binom{C_{.j}}{2}}. \tag{7}$$

The *COND* is defined as follows

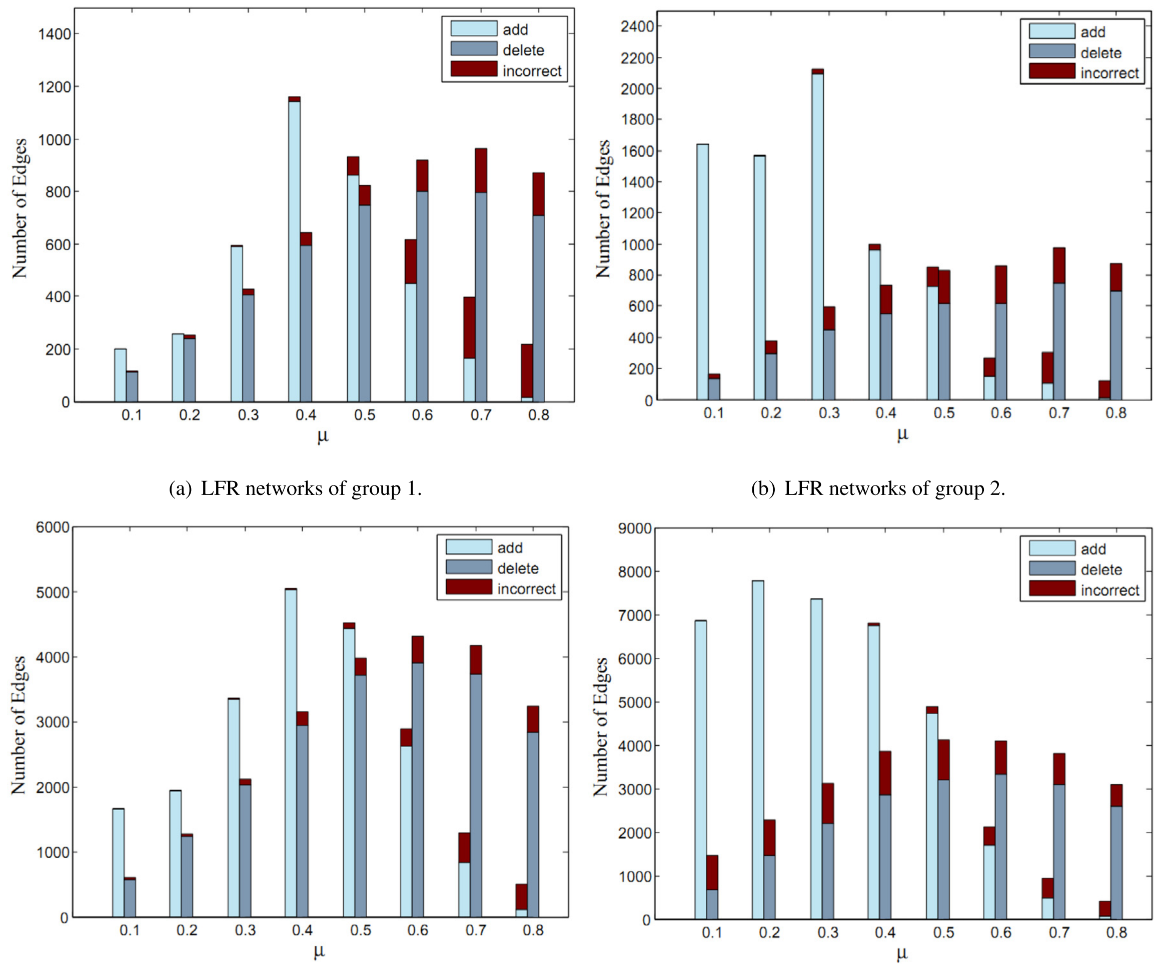
$$COND=\frac{1}{k}\sum_{s=1}^{k}\frac{cut(s)}{\min\{vol(s),vol(\bar{s})\}}, \tag{8}$$

where k is the number of communities found in the network, $cut(s)$ is the number of edges with only one node in the community s, while the other node in the complement node set of s, and $vol(s)$ is the sum of degree of all nodes in s. Here, the complement node set of s, denoted by $\bar{s}$, contains all nodes not in s. A smaller value of *COND* indicates a better detected result.

4.2. Experiments on synthetic benchmark networks

Fig. 2 plots the *NMI* values of the proposed CLPE and the six compared community detection algorithms on four groups of LFR networks, where the *NMI* values reported here are the average values of eight LFR networks with the same parameter setting. From Fig. 2, the following observations can be obtained.

First, compared with the six considered community detection algorithms, the proposed CLPE algorithm achieves an overall better performance on LFR networks. Although when $\mu<0.5$, the EdgeBoost also holds a comparable performance. However, when $\mu>0.5$, the performance of EdgeBoost considerably deteriorates, whereas the proposed CLPE performs the best. It is because that the EdgeBoost uses the traditional link prediction strategy to



(a) LFR networks of group 1.

(b) LFR networks of group 2.

(c) LFR networks of group 3.

(d) LFR networks of group 4.

Fig. 4. The ratio of correctly and incorrectly added edges and deleted edges on LFR networks by the proposed CLPE. Each histogram shows the mean value of the numbers of edges averaged over eight LFR networks with the same parameter setting.

add the edges. When the community structure is clear, it will predict a large number of edges in the communities, by which the community structure of networks can be enhanced. Therefore, the EdgeBoost holds a comparable performance when $\mu < 0.5$. However, as μ increases, namely, the community structure of the networks becomes unclear, the traditional link prediction strategy will predict a large number of edges between different communities. It will weaken the community structure of the network. Therefore, the performance of EdgeBoost considerably deteriorates when $\mu > 0.5$.

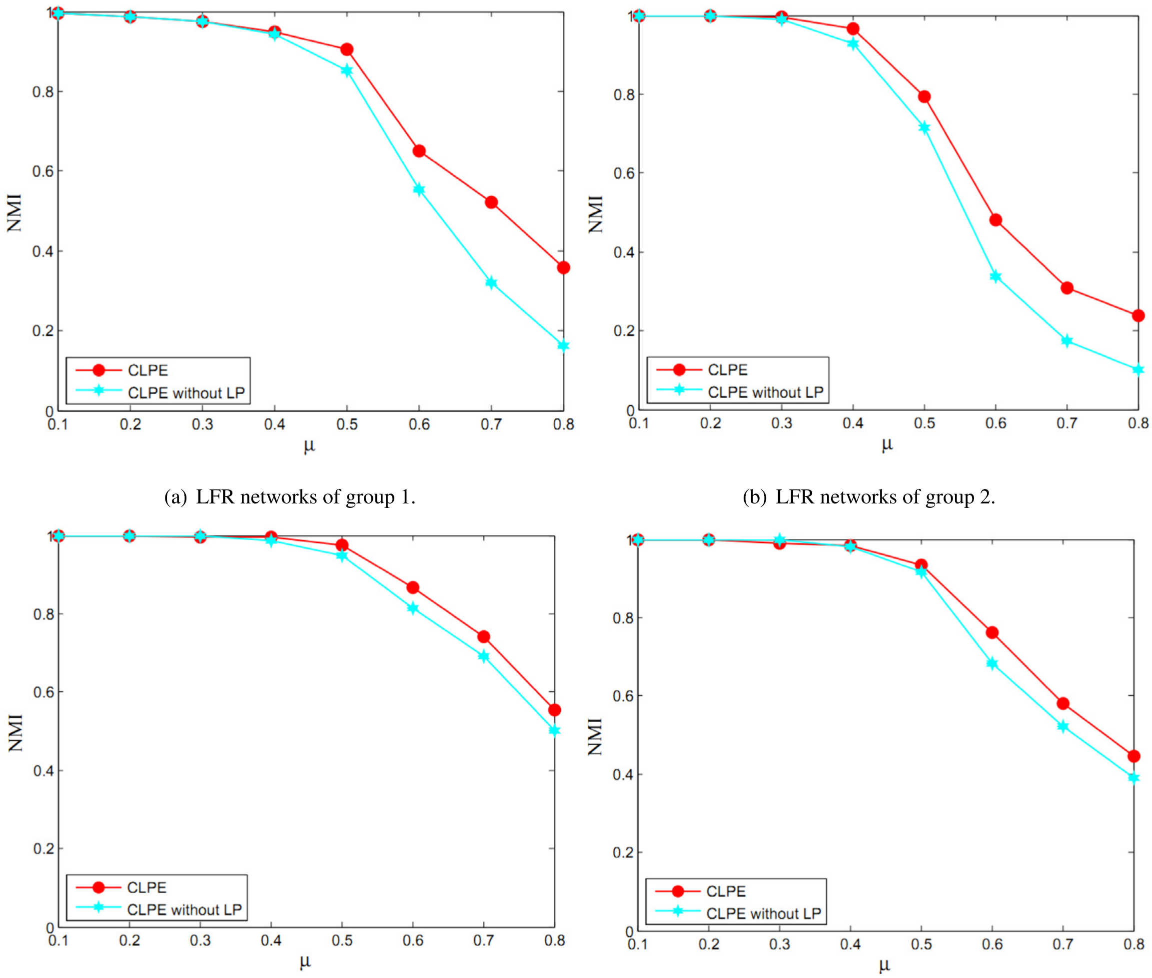
Second, the proposed CLPE algorithm is more suitable for LFR networks with an ambiguous community structure. In Fig. 2, the performance of all compared community detection algorithms considerably deteriorates with the increase of μ. It is because that as μ increases, the community structure of networks becomes ambiguous, and the community detection will become challenging. However, when $\mu > 0.5$, the proposed CLPE achieves significantly better performance than the other six algorithms. Contrarily, the *NMI* of SCAN and CNM decrease significantly.

Third, the proposed CLPE algorithm performs better than the compared algorithms on the network with a large size. It can be seen from Fig. 2 that the proposed CLPE performs better on the LFR networks of groups 3 and 4 than on the LFR networks of groups 1 and 2. It means that the proposed CLPE is more applicable to networks with a large size.

Fig. 3 plots the *ARI* values of the CLPE and the six compared community detection algorithms on four groups of LFR networks. From Fig. 3, we can find that the proposed CLPE algorithm still achieves a better performance than the other six compared community detection algorithms. From Fig. 3, we can find that the proposed CLPE holds the largest *ARI* on the four groups of the LFR networks. When $\mu > 0.5$, the performance of all algorithms considerably deteriorates, whereas the performance of the proposed CLPE is more robust than the other six compared algorithms. Besides, the *ARI* of CLPE on LFR networks of groups 3 and 4 is larger than that on LFR networks of groups 1 and 2. Therefore, the results obtained from Fig. 2, can still be obtained from Fig. 3.

To show the effectiveness of the proposed link prediction strategy, we calculate the ratio of incorrectly added (deleted) edges on the LFR networks with different values of μ. Note that we calculate this ratio on the enhanced networks, namely, the output of Algorithm 2. If an added edge connecting two nodes in the same community, it is a correctly added edge. If a deleted edge connecting two nodes in different communities on the original network, it is a correctly deleted edge. The experimental results are shown in Fig. 4.

From Fig. 4, we can see that when the community structure of the networks is clear, namely, $\mu \le 0.5$, the central node based link prediction has a better accuracy of adding and deleting edges. When the community structure of the networks becomes



(a) LFR networks of group 1.

(b) LFR networks of group 2.

(c) LFR networks of group 3.

(d) LFR networks of group 4.

Fig. 5. The *NMI* values of the proposed CLPE with link prediction strategy and CLPE without link prediction strategy on four groups of LFR benchmark networks with different values of μ.

ambiguous, namely, $\mu > 0.5$, the accuracy of the proposed link prediction strategy decreases. But it still remains at a relatively satisfactory accuracy. The experimental results in Fig. 2 indicate that the CLPE algorithm performs better than the existing community detection algorithms when the community structure of the networks is ambiguous, namely, $\mu > 0.5$. Therefore, the proposed central node based link prediction strategy will improve the performance of community detection for the networks without a clear community structure.

To further illustrate the effectiveness of the proposed link prediction strategy, Fig. 5 shows the performance of CLPE algorithm with the proposed link prediction strategy and the CLPE algorithm without the proposed link prediction strategy on four groups of LFR networks. In Fig. 5, the *NMI* of CLPE with the link prediction strategy is greater than that of the CLPE algorithm without the link prediction strategy when $\mu \geq 0.4$. When $\mu < 0.4$, both of them can obtain an accurate detected result. It demonstrates that the proposed link prediction strategy used in CLPE improves the performance of community detection when the community structure of networks is ambiguous.

4.3. Experiments on real-world networks

Table 4 shows the *COND* values of the proposed CLPE and six compared community detection algorithms on nine real-world networks. In Table 4, the best *COND* value on each network is highlighted. As can be seen from Table 4, the CLPE algorithm obtains the best *COND* value on five of nine real-world networks. Furthermore, on the PPI network, the value of *COND* obtained by CLPE algorithm is only slightly greater than the smallest *COND* value obtained by the compared community detection algorithms. Overall, the CLPE algorithm outperforms the other community detection algorithms on real-world networks.

Table 4
The conductance of nine real-world networks.

Network	EB	Louvain	SSCF	SCAN	CNM	CDFR	CLPE
karate	0.267	0.288	0.207	0.487	0.281	**0.132**	0.292
dolphin	0.243	0.292	**0.064**	0.533	0.316	0.153	0.219
football	0.318	0.285	0.340	0.482	0.288	0.309	**0.285**
Polbooks	0.278	0.276	0.189	0.502	0.255	0.185	**0.182**
netscience	0.005	0.002	0.201	0.039	0.003	0.858	**0.002**
PPI	0.401	0.199	0.294	0.496	**0.194**	0.727	0.198
Blogs	0.168	0.107	0.141	0.430	0.099	0.626	**0.097**
ca-HepTh1	0.284	**0.017**	0.254	0.326	0.029	0.705	0.257
Erdos	0.374	0.233	0.576	0.568	0.256	0.769	**0.233**

* 'EB' represents the EdgeBoost algorithm.

Moreover, the ratio of incorrectly added (deleted) edges to all added (deleted) edges on the real-world networks is also calculated and shown in Fig. 6. Note that Fig. 6 only shows the ratio

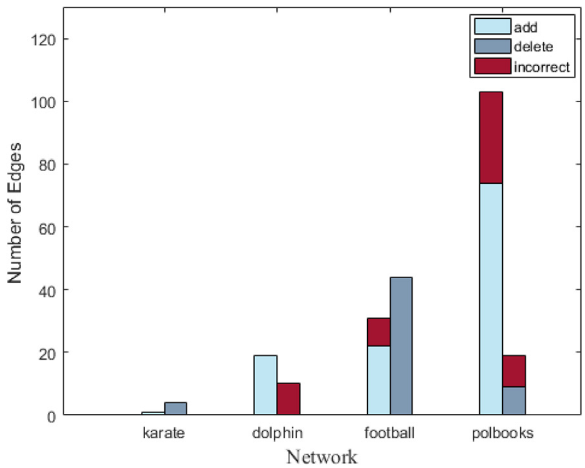


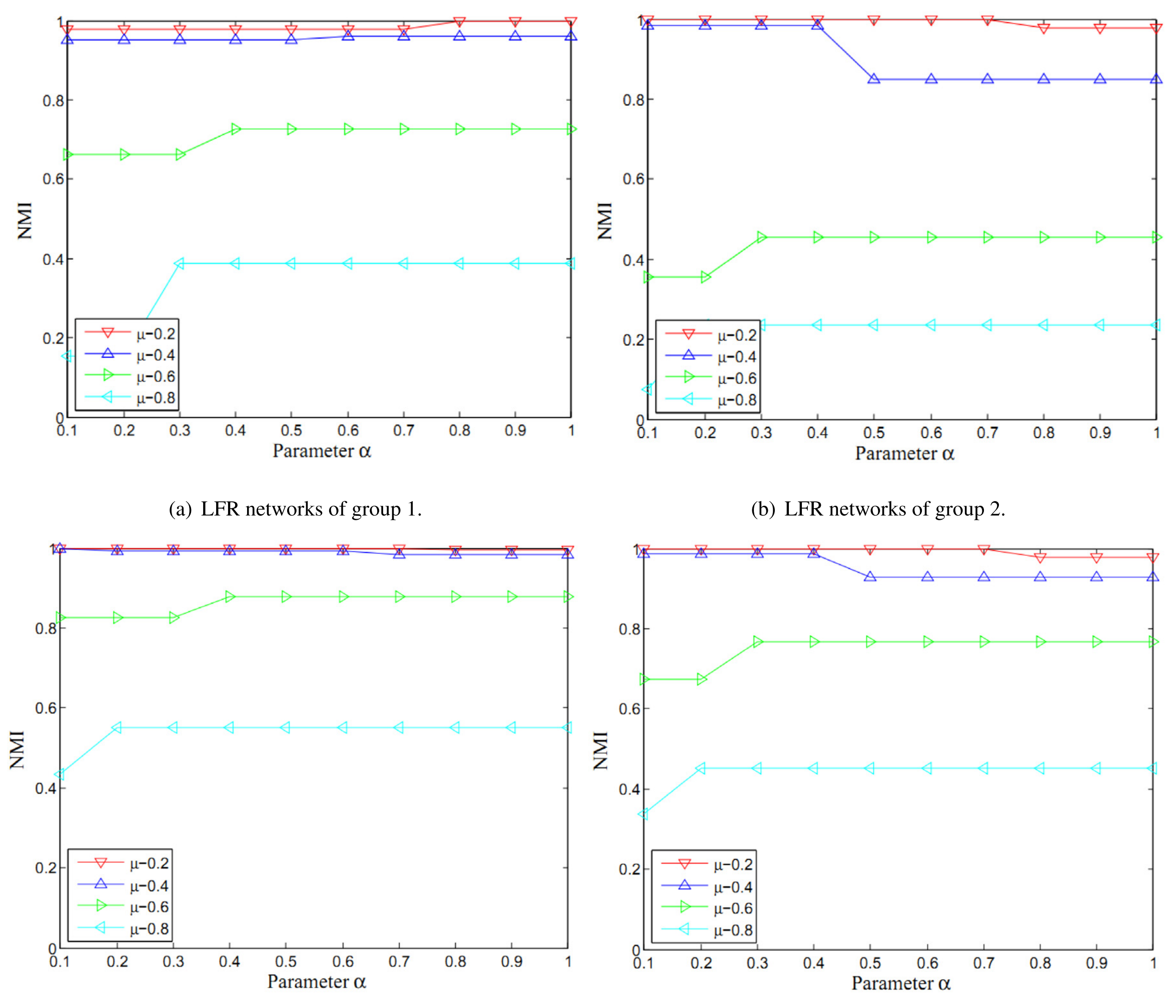
Fig. 6. The ratio of correctly and incorrectly added edges and deleted edges on real-world networks by the proposed CLPE.

of incorrectly added (deleted) edges of polbooks, karate, dolphin and football, because among the nine real-world networks, only their ground truth is known.

In Fig. 6, even though some edges are incorrectly added into and deleted from the original real-world networks, the central node based link prediction strategy used in CLPE still enhances the structure of communities. It is because that in Fig. 6 the number of correctly added and deleted edges is greater than the number of incorrectly added and deleted edges. Additionally, in Table 4, the competitive performance on the nine real-world networks also demonstrates the effectiveness of the central node based link prediction strategy used in CLPE. Therefore, on real-world networks the CLPE algorithm is still effective.

4.4. Sensitivity of parameter α

After completing the community expansion, the CLPE algorithm merges the communities according to the modularity Q. In order to show the sensitive of parameter α of CLPE, we use the CLPE to detect the communities on four groups of LFR networks under the parameter α varying from 0.1 to 1 with interval 0.1. The experimental results are shown in Fig. 7. As shown in Fig. 7, it can be found that the proposed CLPE has a better overall performance when the value of parameter α is set to 0.3 to 0.4. The main reason is that when the structure of communities is clear, the communities obtained after the expansion operation



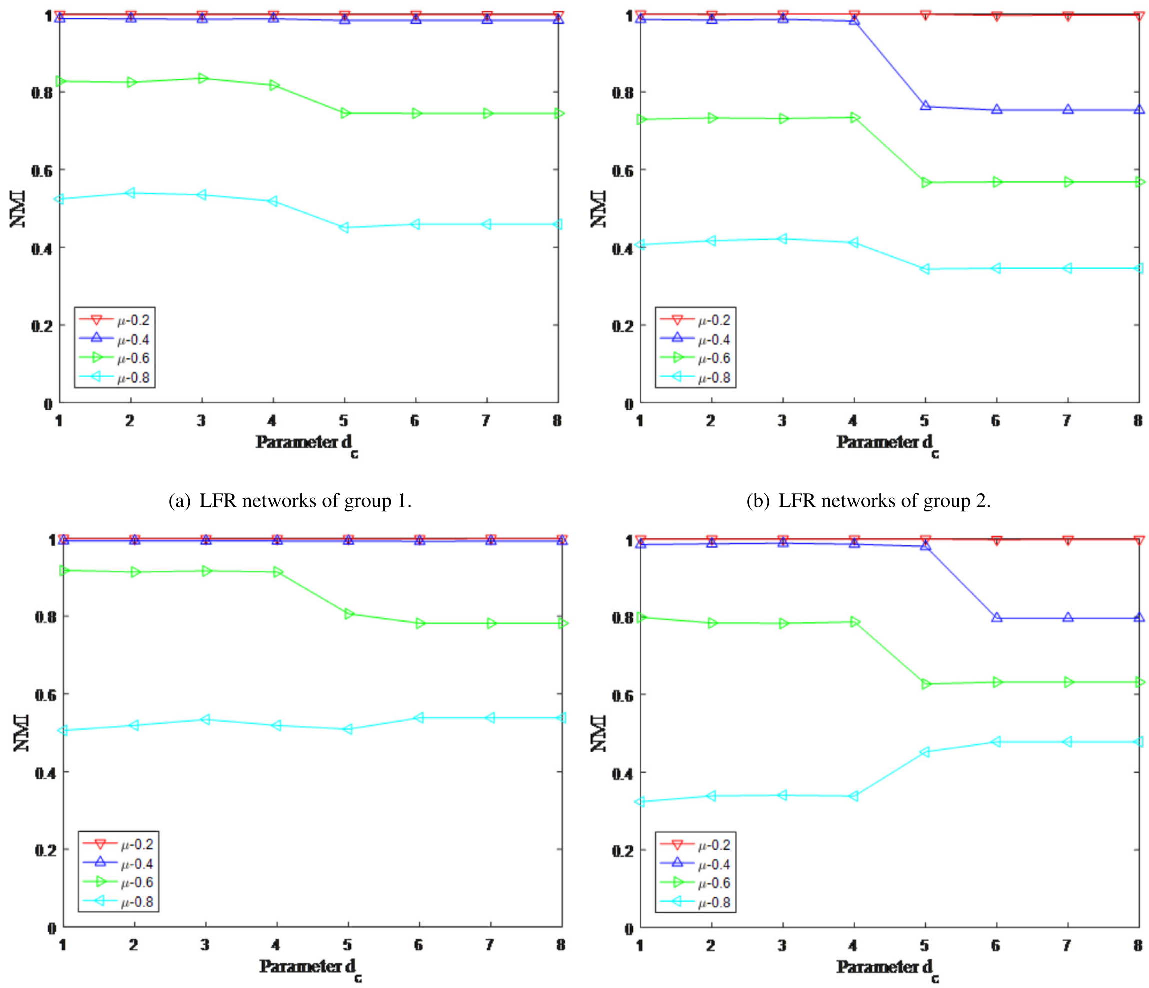
(a) LFR networks of group 1.

(b) LFR networks of group 2.

(c) LFR networks of group 3.

(d) LFR networks of group 4.

Fig. 7. The *NMI* values of the proposed CLPE on four groups of LFR networks under different values of parameter α.



(a) LFR networks of group 1.

(b) LFR networks of group 2.

(c) LFR networks of group 3.

(d) LFR networks of group 4.

Fig. 8. The *NMI* values of the proposed CLPE on four groups of LFR networks under different values of parameter d_c.

generally have the high modularity. At this time, the merged communities will often have a better community structure. In this case, the smaller parameter α is, the better merged communities will be obtained. However, when the community structure of the networks is ambiguous, the structure of communities obtained after the expansion operation is also ambiguous. At this time, the merged communities will suffer from over merging, which results in a decreasing accuracy. In this case, the parameter α cannot be set too large.

4.5. Sensitivity of parameter d_c

We also use four groups of LFR networks to verify the sensitivity of the parameter d_c of CLPE on four groups of LFR networks under parameter d_c varying from 1 to 8 with interval 1. The experimental results are shown in Fig. 8. In Fig. 8, we can find that the *NMI* of CLPE decreases as d_c increases in most cases. It is because that as d_c increases, the density of each node will increase, which will increase the difficulty of finding central nodes. However, when the community structure of the networks is ambiguous, the ρ_i with a small d_c will be seriously affected by noisy links. A large d_c will reduce the effect of noisy links. Therefore, in Fig. 8(d), when $\mu = 0.8$ the *NMI* of CLPE increases as d_c increases. Overall, the proposed CLPE has a better performance when the parameter d_c is set to 1 to 4. However, with the increase of d_c, the time of calculating ρ_i will increase. Therefore, in this paper, we set d_c to 1.

5. Discussion

Because the community detection is an ill-defined problem, and the community is also a structure that is not well defined [61], in some cases a community might not contain a central node, such as cliquish structure. However, the proposed CLPE can still detect the communities without central nodes.

Take the network in Fig. 1 as an example. In Fig. 1, the nodes $\{V_6, V_7, V_8, V_9\}$ constitute a cliquish structure. After the community $\{V_1, V_2, V_3, V_4, V_5\}$ is detected, the node with the largest *SC* value that is not visited, namely, the node V_6, will be added in a new community. Next, the neighboring nodes of V_6 that are not visited, namely, $\{V_7, V_8, V_9\}$, will be checked whether they can be added in this community. For the node V_7, its neighboring nodes are $\{V_6, V_8, V_9\}$. The average similarity between the node V_7 and its neighboring nodes is $oval = (\frac{2}{5} + \frac{1}{2} + \frac{1}{2})/3 = \frac{7}{15}$. Because only one neighboring node of the node V_6 has been added in this community, and the similarity between node V_6

and V_7 ($\frac{2}{5}$) is smaller than $oval$, the node V_7 cannot be added in this community. Similarly, the nodes V_8 and V_9 also cannot be added in this community. Next, the proposed method will divide nodes V_7, V_8 and V_9 into a new community. After community expansion, the CLPE will merge the communities according to the modularity. In the above example, after community expansion, we obtain three communities including $\{V_1, V_2, V_3, V_4, V_5\}$, $\{V_6\}$ and $\{V_7, V_8, V_9\}$. Because the modularity (0.3044) is greater than 0.3, and the modularity is 0.4244 after merging the communities $\{V_6\}$ and $\{V_7, V_8, V_9\}$, the proposed CLPE will merge the communities $\{V_6\}$ and $\{V_7, V_8, V_9\}$. Finally, the cliquish structure $\{V_6, V_7, V_8, V_9\}$ is found. Therefore, the proposed method can detect the communities without central nodes.

6. Conclusion

As the community structure of complex networks becomes ambiguous, the performance of most existing community detection algorithms considerably deteriorates. In this paper, we propose an algorithm called CLPE to detect the community in networks, especially for the networks with an ambiguous community structure. Specifically, CLPE first uses a designed central node based link prediction strategy to add/remove the edges so that the community structure of networks is enhanced. Then, a community expansion strategy is employed to detect all communities in the networks. Experimental results on synthetic benchmark networks and real-world networks demonstrate the effectiveness of CLPE.

There still remains some work that needs to be further studied. In this paper, in order to enhance the community structure of networks by link prediction, we propose a central node based link prediction strategy. However, because the proposed link prediction strategy needs to calculate the similarity of a large number of pairs of nodes, it is time-consuming. Therefore, in the future a more efficient link prediction strategy is expected. Additionally, in the real world, a node in the network usually belongs to various communities. Consequently, extending CLPE into overlapping community detection also needs to be investigated. Moreover, the mathematical proof of CLPE also deserves further investigation.

CRediT authorship contribution statement

Hao Jiang: Methodology, Writing - original draft. **Zhenjie Liu:** Methodology, Validation. **Chunlong Liu:** Methodology, Validation. **Yansen Su:** Writing - review & editing. **Xingyi Zhang:** Methodology, Writing - review & editing.

Acknowledgments

This work is supported in part by the Key Project of Science and Technology Innovation 2030 supported by the Ministry of Science and Technology of China (Grant No. 2018AAA0101300) and the National Natural Science Foundation of China (Grant No. 61672033, 61822301, 61876123, and U1804262).

References

[1] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.
[2] C. Conaco, D.S. Bassett, H. Zhou, M.L. Arcila, S.M. Degnan, B.M. Degnan, K.S. Kosik, Functionalization of a protosynaptic gene expression network, Proc. Natl. Acad. Sci. 109 (Supplement 1) (2012) 10612–10618.
[3] R. Sharan, I. Ulitsky, R. Shamir, Network-based prediction of protein function, Mol. Syst. Biol. 3 (1) (2007) 88.
[4] M.J. Cobo, M. Martínez, M. Gutiérrez-Salcedo, H. Fujita, E. Herrera-Viedma, 25 years at knowledge-based systems: A bibliometric analysis, Knowl.-Based Syst. 80 (2015) 3–13.
[5] Y. Zhang, H. Chen, J. Lu, G. Zhang, Detecting and predicting the topic change of knowledge-based systems: a topic-based bibliometric analysis from 1991 to 2016, Knowl.-Based Syst. 133 (2017) 255–268.
[6] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3) (2010) 75–174.
[7] W. Yuan, D. Guan, Y. Lee, S. Lee, S.J. Hur, Improved trust-aware recommender system using small-worldness of trust networks, Knowl.-Based Syst. 23 (3) (2010) 232–238.
[8] B.W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, Bell Syst. Tech. J. 49 (2) (1970) 291–307.
[9] D. Chen, Y. Yan, D. Wang, X. Huang, Community detection algorithm based on structural similarity for bipartite networks, in: Proceedings of the 7th IEEE International Conference on Software Engineering and Service Science, 2016, pp. 98–102.
[10] W.E. Donath, A.J. Hoffman, Lower bounds for the partitioning of graphs, in: Selected Papers of Alan J Hoffman: With Commentary, World Scientific, 2003, pp. 437–442.
[11] J. S.H.I., Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[12] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: A nalysis and an algorithm, in: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, MIT Press, 2001, pp. 849–856.
[13] A. Mahmood, M. Small, Subspace based network community detection using sparse linear coding, IEEE Trans. Knowl. Data Eng. 28 (3) (2016) 801–812.
[14] L. Ni, P. Manman, W. Jiang, K. Li, A community detection algorithm based on multi-similarity method, Cluster Comput. 22 (2018) 2865–2874.
[15] M.E. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133.
[16] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, D. Wagner, On Modularity: np-Completeness and Beyond, Tech. Rep., Faculty of Informatics, Univ. Karlsruhe, Wagner, 2006.
[17] Z. Bu, C. Zhang, Z. Xia, J. Wang, A fast parallel modularity optimization algorithm (FPMQA) for community detection in online social network, Knowl.-Based Syst. 50 (2013) 246–259.
[18] Z. Zhang, P. Peng, D. Han, T. Ming, Self-adaptive louvain algorithm: Fast and stable community detection algorithm based on the principle of small probability event, Physica A 506 (15) (2018) 975–986.
[19] C. Mu, J. Xie, Y. Liu, F. Chen, Y. Liu, L. Jiao, Memetic algorithm with simulated annealing strategy and tightness greedy optimization for community detection in networks, Appl. Soft Comput. 34 (2015) 485–501.
[20] C. Pizzuti, GA-Net: A genetic algorithm for community detection in social networks, in: Proceedings of the International Conference on Parallel Problem Solving from Nature, Springer, 2008, pp. 1081–1090.
[21] R. Shang, J. Bai, L. Jiao, C. Jin, Community detection based on modularity and an improved genetic algorithm, Physica A 392 (5) (2013) 1215–1231.
[22] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (3) (2007) 036106.
[23] M.J. Barber, J.W. Clark, Detecting network communities by propagating labels under constraints, Phys. Rev. E 80 (2) (2009) 026129.
[24] Z. Lin, X. Zheng, N. Xin, D. Chen, CK-LPA: Efficient community detection algorithm based on label propagation with community kernel, Physica A 416 (2014) 386–399.
[25] G. Chun, Z. Ruisheng, Z. Zhili, W. Jiaxuan, H. Rongjing, LPA-CBD an improved label propagation algorithm based on community belonging degree for community detection, Internat. J. Modern Phys. C 29 (2) (2018) 1850011.
[26] T. Ma, M. Yue, J. Qu, T. Yuan, M. Al-Rodhaan, PSPLPA: Probability and similarity based parallel label propagation algorithm on spark, Physica A 503 (1) (2018) 366–378.
[27] U. Brandes, M. Gaertler, D. Wagner, Experiments on graph clustering algorithms, in: Proceedings of the European Symposium on Algorithms, Springer, 2003, pp. 568–579.
[28] P. Holme, M. Huss, H. Jeong, Subnetwork hierarchies of biochemical pathways, Bioinformatics 19 (4) (2003) 532–538.
[29] R. Lambiotte, J. Delvenne, M. Barahona, Random walks, Markov processes and the multiscale modular organization of complex networks, IEEE Trans. Netw. Sci. Eng. 1 (2) (2014) 76–90.
[30] P. Pons, M. Latapy, Computing communities in large networks using random walks, in: Proceedings of the International Symposium on Computer and Information Sciences, Springer, 2005, pp. 284–293.
[31] J. Eustace, X. Wang, J. Li, Approximating web communities using subspace decomposition, Knowl.-Based Syst. 70 (2014) 118–127.
[32] C. Pizzuti, Evolutionary computation for community detection in networks: A review, IEEE Trans. Evol. Comput. 22 (3) (2018) 464–483.
[33] C. Liu, J. Liu, Z. Jiang, A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks, IEEE Trans. Cybern. 44 (12) (2014) 2274–2287.

[34] M. Gong, Q. Cai, X. Chen, L. Ma, Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition, IEEE Trans. Evol. Comput. 18 (1) (2014) 82–97.
[35] L. Zhang, H. Pan, Y. Su, X. Zhang, Y. Niu, A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection, IEEE Trans. Cybern. 47 (9) (2017) 2703–2716.
[36] B. Amiri, L. Hossain, J.W. Crawford, R.T. Wigand, Community detection in complex networks: Multi-objective enhanced firefly algorithm, Knowl.-Based Syst. 46 (2013) 1–11.
[37] Z. Zhang, Y. Wang, Y. Ahn, Overlapping community detection in complex networks using symmetric binary matrix factorization, Phys. Rev. E 87 (6) (2013) 062803.
[38] L. Lü, T. Zhou, Link prediction in complex networks: A survey, Physica A 390 (6) (2011) 1150–1170.
[39] M. Burgess, E. Adar, M. Cafarella, Link-prediction enhanced consensus clustering for complex networks, PLoS One 11 (5) (2016) e0153384.
[40] H. Cheng, Y. Ning, Z. Yin, C. Yan, X. Liu, Z. Zhang, Community detection in complex networks using link prediction, Mod. Phys. Lett. B 32 (01) (2016) 1850004.
[41] M. Chen, A. Bahulkar, K. Kuzmin, B.K. Szymanski, Improving network community structure with link prediction ranking, in: Proceedings of the 7th Workshop on Complex Networks CompleNet, Springer, 2016, pp. 145–158.
[42] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[43] X. Wang, G. Liu, J. Li, J.P. Nees, Locating structural centers: A density-based clustering method for community detection, PLoS One 12 (1) (2017) e0169355.
[44] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[45] V.D. Blondel, J.L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (10) (2008) P10008.
[46] X. Xu, N. Yuruk, Z. Feng, T.A. Schweiger, SCAN: A structural clustering algorithm for networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2007, pp. 824–833.
[47] A. Clauset, M.E. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (6) (2004) 066111.
[48] W. Luo, Z. Yan, C. Bu, D. Zhang, Community detection by fuzzy relations, IEEE Trans. Emerg. Top. Comput. PP (99) (2017) 1.
[49] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (4) (2008) 046110.
[50] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (4) (1977) 452–473.
[51] D. Lusseau, The emergent properties of a dolphin social network, Proc. R. Soc. Lond. Ser. B 270 (Suppl 2) (2003) S186–S188.
[52] M.E. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (23) (2006) 8577–8582.
[53] S. Gregory, Finding overlapping communities in networks by label propagation, New J. Phys. 12 (10) (2010) 103018.
[54] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, Acm Trans. Knowl. Discov. Data 1 (1) (2007) 2.
[55] S. He, Z. Zhu, G. Jia, D. Tennant, Q. Huang, K. Tang, J. Heath, M. Musolesi, X. Yao, Cooperative co-evolutionary module identification with application to Cancer disease module discovery, IEEE Trans. Evol. Comput. 20 (6) (2016) 874–891.
[56] J.P. Bagrow, Evaluating local community methods in networks, J. Stat. Mech. Theory Exp. 2008 (05) (2008) P05001.
[57] Z. Ding, X. Zhang, D. Sun, B. Luo, Overlapping community detection based on network decomposition, Sci. Rep. 6 (2016) 24115.
[58] S.E. Schaeffer, Graph clustering, Comput. Sci. Rev. 1 (1) (2007) 27–64.
[59] H. Yin, A.R. Benson, J. Leskovec, D.F. Gleich, Local higher-order graph clustering, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 555–564.
[60] U. Gargi, W. Lu, V. Mirrokni, S. Yoon, Large-scale community detection on youtube for topic discovery and exploration, in: Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, 2011.
[61] S. Fortunato, D. Hric, Community detection in networks: A user guide, Phys. Rep. 659 (2016) 1–44.